ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΠΟΠΤΥΧΙΑΚΟ
ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

# Σχεδιασμός και Ανάπτυξη Διαδικτυακής Πλατφόρμας για τη Διαχείριση Πληροφοριών Μοτοσυκλέτας

Ιωάννης Δημήτριος Αρμαμέντος
321/201203, icsd12013@icsd.aegean.gr

Επιβλέπων: Χαράλαμπος Αλεξόπουλος

ΙΟΥΛΙΟΣ 2026
www.icsd.aegean.gr

Η Διπλωματική Εργασία παρουσιάστηκε ενώπιον

του Διδακτικού Προσωπικού του Πανεπιστημίου Αιγαίου

για την απόκτηση του προπτυχιακού διπλώματος Ηλεκτρονικού Μηχανικού

Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ
ΕΠΙΚΥΡΩΝΕΙ ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ
Ιωάννη Δημητρίου Αρμαμέντου

Χαράλαμπος Αλεξόπουλος, Επιβλέπων

Επίκουρος Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

---

Κωνσταντίνος Μαλιάτσος, Μέλος

Μόνιμος Επίκουρος Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

---

Γιάννης Χαραλαμπίδης, Μέλος

Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

© 2026

Με επιφύλαξη παντός δικαιώματος.

Ιωάννης Δημήτριος Αρμαμέντος

Πανεπιστήμιο Αιγαίου

# Περίληψη

**Σχεδιασμός και Ανάπτυξη Διαδικτυακής Πλατφόρμας για τη Διαχείριση Πληροφοριών Μοτοσυκλέτας**

Το Moto-Vault είναι μια διαδικτυακή πλατφόρμα για τη διαχείριση των πληροφοριών μοτοσυκλέτας. Η ανάγκη για την ανάπτυξη της εφαρμογής προέκυψε από τον σκορπισμό δεδομένων όπως το ιστορικό συντήρησης, οι ανεφοδιασμοί, τα έξοδα, τα ανταλλακτικά, τα αξεσουάρ, τα έγγραφα, οι φωτογραφίες, οι υπενθυμίσεις και οι διαδρομές GPX σε διαφορετικά μέσα και εφαρμογές. Στόχος ήταν η συγκέντρωση αυτών των πληροφοριών σε ένα εύκολο περιβάλλον, οργανωμένο γύρω από κάθε μοτοσυκλέτα.

Η ανάπτυξη ακολούθησε τη μεθοδολογία Design Science Research Methodology. Για το frontend χρησιμοποιήθηκε Angular, ενώ το backend αναπτύχθηκε με Spring Boot και τα δεδομένα αποθηκεύονται σε PostgreSQL. Υποστηρίζονται η διαχείριση πολλαπλών μοτοσυκλετών, η καταγραφή συντήρησης και καυσίμων, η αποθήκευση ιδιωτικών αρχείων και φωτογραφιών, οι υπενθυμίσεις, η εισαγωγή και προβολή αρχείων GPX, τα γενικά στατιστικά, καθώς και λειτουργίες εισαγωγής, εξαγωγής και δημιουργίας αναφορών. Στο κομμάτι της ασφάλειας, υλοποιήθηκε αυθεντικοποίηση(επαλήθευση ταυτότητας), έλεγχος πρόσβασης, στην ακεραιότητα των δεδομένων, στη διαχείριση σφαλμάτων και στην αξιοπιστία των λειτουργιών αποθήκευσης.

Η χρηστικότητα του συστήματος αξιολογήθηκε με το ερωτηματολόγιο System Usability Scale από δώδεκα μοτοσικλετιστές. Η μέση βαθμολογία ήταν 82, παρέχοντας θετική αρχική γνώμη για την χρηστικότητα της εφαρμογής. Εντοπίστηκαν περιορισμοί που σχετίζονται με το μικρό δείγμα, τη χειροκίνητη καταχώριση δεδομένων, την απουσία εφαρμογής για κινητές συσκευές και την έλλειψη διασύνδεσης με εξωτερικές υπηρεσίες. Το Moto-Vault ανταποκρίνεται στους βασικούς στόχους της εργασίας και αποτελεί μια υποδομή για μελλοντική ανάπτυξη και επέκταση.

**Λέξεις κλειδιά**: διαχείριση πληροφοριών μοτοσυκλέτας, διαδικτυακή εφαρμογή, ιστορικό συντήρησης, καταγραφή καυσίμων, διαδρομές GPX, αξιολόγηση χρηστικότητας, κλίμακα χρηστικότητας συστήματος

# Abstract

**Design and Development of a Web Platform for Motorcycle Information Management**

This thesis is about the design, development, and evaluation of Moto-Vault, a web platform for the organized management of motorcycle related information. The need for the application rose from the fragmentation of data such as maintenance history, fuel entries, expenses, spare parts, accessories, documents, photographs, reminders, and GPX routes across different media and applications. The main objective was to bring this information together in a unified, secure, and easy environment organized around each motorcycle.

The development process followed the Design Science Research Methodology. The application was implemented using Angular for the frontend, Spring Boot for the backend, and PostgreSQL for structured data storage. Its main features include the management of multiple motorcycles, maintenance and fuel tracking, private file and photo storage, reminders, GPX file import and visualization, summary statistics, and data import, export, and reporting functions. Extra effort was given to authentication, access control, data integrity, error handling, and the reliability of storage-related operations.

The usability of the system was evaluated with the System Usability Scale by twelve motorcyclists. The mean score was 82, providing a positive initial indication of the perceived usability of the application. The evaluation also highlighted limitations related to the small sample, the manual entry of data, the absence of a native mobile application, and the lack of integration with external services. Overall, Moto-Vault meets the main objectives of the thesis and provides a functional foundation for future extensions.

**Keywords:** motorcycle information management, web application, maintenance history, fuel tracking, GPX routes, usability evaluation, System Usability Scale

# Ενδεικτικός Πίνακας Περιεχομένων

1. Εισαγωγή ..... 18
1.1 Ο σκοπός της Διπλωματικής Εργασίας ..... 18
1.2 Σκοπός της Διπλωματικής Εργασίας ..... 19
1.3 Αντικείμενο και εύρος της Διπλωματικής Εργασίας ..... 20
1.4 Δομή ..... 20
2. Ανάλυση Τρέχοντος Επιπέδου ..... 22
2.1 Βασικές Έννοιες και Ορισμοί ..... 22
2.1.1 Διαχείριση πληροφοριών του οχήματος ..... 22
2.1.2 Το Ιστορικό συντήρησης ..... 22
2.1.3 Προληπτική και διορθωτική συντήρηση ..... 22
2.1.4 Καταγραφή κόστους και κατανάλωσης καυσίμου ..... 23
2.1.5 Φωτογραφικό υλικό και έγγραφα ..... 23
2.1.6 Αρχεία διαδρομών GPX ..... 23
2.1.7 Ψηφιακό ιστορικό μοτοσυκλέτας ..... 23
2.2 Ψηφιακή καταγραφή πληροφοριών οχήματος ..... 24
2.2.1 Από την έντυπη στην ψηφιακή καταγραφή ..... 24
2.2.2 Οργάνωση και σύνδεση των δεδομένων ..... 24
2.2.3 Οφέλη για τον ιδιοκτήτη ..... 25
2.2.4 Ποιότητα και αξιοπιστία των πληροφοριών ..... 25
2.2.5 Προστασία δεδομένων και ασφάλεια ..... 25
2.2.6 Μεταφορά και εξαγωγή δεδομένων ..... 26
2.2.7 Περιορισμοί της ψηφιακής καταγραφής ..... 26
2.3 Υφιστάμενες εφαρμογές και υπηρεσίες ..... 27
2.3.1 Motorfolio ..... 27
2.3.2 Στρίφτεροι ..... 27
2.3.3 MotoVault Motorcycle Garage ..... 27
2.3.4 Rydful ..... 27
2.3.5 RideLog ..... 28
2.3.6 Fuel Log ..... 28
2.3.7 Motohub ..... 28
2.3.8 Drivvo ..... 28
2.3.9 MotorManage ..... 28
2.3.10 MotoLogger ..... 29
2.3.11 Ομοιότητες στη σχεδίαση και η επίδραση των εργαλείων AI ..... 29

2.4 Συγκριτική ανάλυση υφιστάμενων λύσεων ..... 30
2.4.1 Κριτήρια σύγκρισης ..... 30
2.4.2 Πίνακας σύγκρισης ..... 31
2.4.3 Κοινό, σκοπός και τρόπος πρόσβασης ..... 31
2.4.4 Συντήρηση, καύσιμα και οικονομική παρακολούθηση ..... 32
2.4.5 Διαδρομές, κοινωνικές λειτουργίες και αρχεία GPX ..... 32
2.4.6 Φωτογραφίες και έγγραφα ..... 33
2.4.7 Θέση και διαφοροποίηση του Moto-Vault ..... 33
2.5 Συμπεράσματα ..... 33
3. Μεθοδολογία ..... 35
3.1 Design Science Research ..... 35
3.2 Design Science Research Methodology ..... 35
3.3 Ερευνητικά ερωτήματα ..... 36
3.3.1 Κύριο ερευνητικό ερώτημα ..... 36
3.3.2 Πρώτο επιμέρους ερευνητικό ερώτημα ..... 37
3.3.3 Δεύτερο επιμέρους ερευνητικό ερώτημα ..... 37
3.3.4 Τρίτο επιμέρους ερευνητικό ερώτημα ..... 37
3.3.5 Τέταρτο επιμέρους ερευνητικό ερώτημα ..... 37
3.3.6 Πέμπτο επιμέρους ερευνητικό ερώτημα ..... 37
3.3.7 Σύνδεση με τη DSRM ..... 38
3.4 Μεθοδολογικοί στόχοι ..... 38
3.4.1 Αναγνώριση και τεκμηρίωση του προβλήματος ..... 38
3.4.2 Μετατροπή των αναγκών σε απαιτήσεις ..... 38
3.4.3 Τεκμηριωμένος και επαναληπτικός σχεδιασμός ..... 38
3.4.4 Τεχνική επαλήθευση και αξιολόγηση χρηστικότητας ..... 38
3.4.5 Καταγραφή αποτελεσμάτων, περιορισμών και βελτιώσεων ..... 39
3.5 Εφαρμογή της DSRM στην παρούσα εργασία ..... 39
3.5.1 Αναγνώριση του προβλήματος ..... 39
3.5.2 Καθορισμός των στόχων ..... 39
3.5.3 Σχεδιασμός και ανάπτυξη ..... 39
3.5.4 Επίδειξη του συστήματος ..... 40
3.5.5 Αξιολόγηση ..... 40
3.5.6 Αντιστοίχιση με τη δομή της εργασίας ..... 40
3.6 Μεθοδολογία αξιολόγησης ..... 41
3.6.1 Τεχνική αξιολόγηση ..... 41

3.6.2 Αξιολόγηση χρηστικότητας ..... 41
3.6.3 Πρόσθετες ερωτήσεις ..... 42
3.6.4 Υπολογισμός της βαθμολογίας SUS ..... 42
3.6.5 Σύνδεση με τα ερευνητικά ερωτήματα ..... 42
4. Ανάλυση και Σχεδιασμός ..... 43
4.1 Χρήστες και σενάρια χρήσης ..... 43
4.1.1 Κύριοι χρήστες ..... 43
4.1.2 Εξωτερικά συστήματα ..... 43
4.1.3 Δευτερεύοντες ενδιαφερόμενοι ..... 43
4.1.4 Χαρακτηριστικά του κοινού ..... 43
4.1.5 Βασικά σενάρια χρήσης ..... 43
4.2 Λειτουργικές απαιτήσεις ..... 44
4.2.1 Λογαριασμός και αυθεντικοποίηση ..... 44
4.2.2 Διαχείριση οχημάτων ..... 44
4.2.3 Ιστορικό συντήρησης ..... 45
4.2.4 Καύσιμα και στατιστικά ..... 45
4.2.5 Αξεσουάρ και ανταλλακτικά ..... 45
4.2.6 Έγγραφα ..... 45
4.2.7 Υπενθυμίσεις ..... 45
4.2.8 Φωτογραφίες οχήματος ..... 45
4.2.9 Διαδρομές GPX ..... 45
4.2.10 Dashboard και συγκεντρωτικά στοιχεία ..... 46
4.2.11 Εισαγωγή, εξαγωγή και αναφορές ..... 46
4.2.12 Προτιμήσεις και διεθνοποίηση ..... 46
4.2.13 Έλεγχος πρόσβασης και αρχεία ..... 46
4.3 Μη λειτουργικές απαιτήσεις ..... 46
4.3.1 Ασφάλεια και ιδιωτικότητα ..... 46
4.3.2 Ακεραιότητα και προστασία δεδομένων ..... 47
4.3.3 Αξιοπιστία και διαχείριση σφαλμάτων ..... 47
4.3.4 Απόδοση ..... 47
4.3.5 Χρηστικότητα ..... 47
4.3.6 Προσβασιμότητα ..... 48
4.3.7 Συντηρησιμότητα και επεκτασιμότητα ..... 48
4.3.8 Φορητότητα και συμβατότητα ..... 48
4.3.9 Δοκιμασιμότητα ..... 48

4.3.10 Λειτουργία και παρατηρησιμότητα ..... 48
4.4 Ιεράρχηση απαιτήσεων ..... 49
4.4.1 Ιεράρχηση απαιτήσεων ..... 49
4.4.2 Μη λειτουργικές προτεραιότητες ..... 50
4.5 Περιπτώσεις χρήσης ..... 50
4.5.1 Ρόλοι χρηστών ..... 50
4.5.2 Κύριες περιπτώσεις χρήσης ..... 51
4.5.3 Διάγραμμα περιπτώσεων χρήσης ..... 52
4.6 Αρχιτεκτονικός σχεδιασμός ..... 54
4.6.1 Συνολική αρχιτεκτονική ..... 54
4.6.2 Monorepo και διαχωρισμός εφαρμογών ..... 54
4.6.3 Frontend ..... 54
4.6.4 Backend ως αρθρωτός μονόλιθος ..... 54
4.6.5 Βάση δεδομένων ..... 55
4.6.6 Αποθήκευση αρχείων ..... 55
4.6.7 Εξωτερικές υπηρεσίες ..... 55
4.6.8 Τυπική ροή αιτήματος ..... 55
4.6.9 Επιλογές και περιορισμοί ..... 55
4.7 Σχεδιασμός δεδομένων ..... 57
4.7.1 Βασικές αρχές ..... 57
4.7.2 Χρήστης και όχημα ..... 57
4.7.3 Συντήρηση και συνημμένα ..... 57
4.7.4 Ανεφοδιασμοί ..... 57
4.7.5 Αξεσουάρ και ανταλλακτικά ..... 57
4.7.6 Έγγραφα και υπενθυμίσεις ..... 57
4.7.7 Φωτογραφίες και διαδρομές ..... 58
4.7.8 Λογαριασμός, tokens και background δεδομένα ..... 58
4.7.9 Ακεραιότητα και διαγραφές ..... 58
4.8 Σχεδιασμός REST API ..... 59
4.8.1 Βασικές αρχές ..... 59
4.8.2 Κύριες ομάδες endpoints ..... 60
4.8.3 HTTP μέθοδοι ..... 60
4.8.4 Αντιπροσωπευτικά endpoints ..... 60
4.8.5 DTOs και validation ..... 61
4.8.6 Αυθεντικοποίηση και ιδιοκτησία ..... 61

4.8.7 Αρχεία ..... 61
4.8.8 Status codes και σφάλματα ..... 61
4.8.9 Δημόσια endpoints ..... 62
4.9 Σχεδιασμός διεπαφής χρήστη ..... 63
4.9.1 Βασικοί στόχοι ..... 63
4.9.2 Πλοήγηση με βάση το όχημα ..... 63
4.9.3 Δημόσιες οθόνες ..... 63
4.9.4 Γκαράζ και λεπτομέρειες οχήματος ..... 63
4.9.5 Dashboard ..... 63
4.9.6 Λίστες και κενές καταστάσεις ..... 64
4.9.7 Φόρμες ..... 64
4.9.8 Ανάδραση και σφάλματα ..... 64
4.9.9 Αρχεία και πολυμέσα ..... 64
4.9.10 Χάρτες και GPX ..... 64
4.9.11 Responsive σχεδιασμός ..... 64
4.9.12 Διεθνοποίηση και προτιμήσεις ..... 64
4.9.13 Προσβασιμότητα ..... 64
4.9.14 Διάγραμμα πλοήγησης ..... 65
4.10 Σχεδιασμός εγκατάστασης και διάγραμμα ανάπτυξης ..... 65
4.10.1 Βασικοί κόμβοι ..... 65
4.10.2 Πρόσβαση χρήστη και Cloudflare ..... 65
4.10.3 VPS και containers ..... 66
4.10.4 PostgreSQL και migrations ..... 66
4.10.5 Object storage ..... 66
4.10.6 Email και εξωτερικές υπηρεσίες ..... 66
4.10.7 Ασφάλεια δικτύου ..... 66
4.10.8 Ροή εγκατάστασης ..... 66
4.10.9 Αντίγραφα ασφαλείας και ανάκτηση ..... 67
5. Τεχνολογίες και εργαλεία ανάπτυξης ..... 68
5.1 Java και Spring Boot ..... 68
5.1.1 Java 21 ..... 68
5.1.2 Spring Boot ..... 68
5.1.3 Persistence και ασφάλεια ..... 69
5.1.4 Βοηθητικές βιβλιοθήκες ..... 69
5.1.5 Αιτιολόγηση επιλογής ..... 69

5.2 Angular και TypeScript ..... 69
5.2.1 Angular ..... 69
5.2.2 TypeScript ..... 70
5.2.3 Βασικοί μηχανισμοί του frontend ..... 70
5.2.4 Διεθνοποίηση και εμφάνιση ..... 70
5.2.5 Δοκιμές και διαχείριση dependencies ..... 70
5.2.6 Αιτιολόγηση επιλογής ..... 70
5.3 PostgreSQL και Flyway ..... 71
5.3.1 PostgreSQL ..... 71
5.3.2 Σχεσιακή ακεραιότητα ..... 71
5.3.3 Flyway ..... 71
5.3.4 Αιτιολόγηση επιλογής ..... 71
5.4 Τεχνολογίες αποθήκευσης αρχείων ..... 72
5.4.1 Αφαίρεση αποθήκευσης ..... 72
5.4.2 Τοπικό filesystem ..... 72
5.4.3 S3 compatible storage ..... 72
5.4.4 Διαχωρισμός δεδομένων και αρχείων ..... 72
5.5 Εργαλεία ανάπτυξης και δοκιμών ..... 73
5.5.1 Περιβάλλον ανάπτυξης και version control ..... 73
5.5.2 Build και dependency management ..... 73
5.5.3 Development υπηρεσίες ..... 73
5.5.4 Εργαλεία δοκιμών ..... 73
5.5.5 Debugging και logging ..... 73
5.6 Συνεχής ολοκλήρωση και έλεγχοι ποιότητας ..... 74
5.6.1 Backend CI ..... 74
5.6.2 Frontend CI ..... 74
5.6.3 Security checks ..... 74
5.6.4 Pull request flow ..... 74
5.7 Υποδομή και εργαλεία εγκατάστασης ..... 75
5.7.1 OVH VPS ..... 75
5.7.2 Docker ..... 75
5.7.3 Dokploy και Traefik ..... 76
5.7.4 Cloudflare ..... 76
5.7.5 R2 και SMTP ..... 76
5.7.6 Trade-offs ..... 76

5.8 Leaflet και OpenStreetMap ..... 76
5.8.1 Leaflet ..... 76
5.8.2 OpenStreetMap ..... 77
5.8.3 Κατανομή επεξεργασίας ..... 77
5.8.4 Περιορισμοί ..... 77
5.9 Συνολική αποτίμηση τεχνολογικών επιλογών ..... 77
6. Υλοποίηση ..... 79
6.1 Οργάνωση και δομή του πηγαίου κώδικα ..... 79
6.1.1 Δομή backend ..... 79
6.1.2 Εσωτερική δομή backend feature ..... 80
6.1.3 Δομή frontend ..... 80
6.1.4 Tests και migrations ..... 80
6.1.5 Αρχεία υποδομής ..... 80
6.2 Υλοποίηση του backend ..... 81
6.2.1 Ροή επεξεργασίας αιτήματος ..... 81
6.2.2 Controllers και DTOs ..... 81
6.2.3 Service layer ..... 81
6.2.4 Ownership-aware πρόσβαση ..... 82
6.2.5 Συναλλαγές ..... 82
6.2.6 Κεντρική διαχείριση σφαλμάτων ..... 82
6.2.7 Αντιπροσωπευτική λειτουργία οχήματος ..... 82
6.3 Υλοποίηση του frontend ..... 82
6.3.1 Ροή αλληλεπίδρασης ..... 83
6.3.2 Standalone components και features ..... 83
6.3.3 Routing και vehicle shell ..... 83
6.3.4 Frontend services και interceptors ..... 83
6.3.5 Reactive state ..... 84
6.3.6 Reactive forms και validation ..... 84
6.3.7 Διεθνοποίηση και προτιμήσεις ..... 84
6.3.8 Αρχεία και downloads ..... 84
6.3.9 Responsive διεπαφή ..... 84
6.4 Αυθεντικοποίηση, εξουσιοδότηση και ασφάλεια ..... 84
6.4.1 Τοπική εγγραφή και επαλήθευση email ..... 85
6.4.2 Σύνδεση και JWT ..... 85
6.4.3 JwtAuthFilter και SecurityContext ..... 85

6.4.4 Session metadata και ανάκληση ... 86
6.4.5 Google Sign-In ... 86
6.4.6 Ανάκτηση και αλλαγή κωδικού ... 86
6.4.7 Ownership authorization ... 86
6.4.8 Upload και parser security ... 86
6.4.9 Rate limiting και production ασφάλεια ... 86
6.5 Υλοποίηση της διαχείρισης δεδομένων ... 87
6.5.1 Persistence model ... 87
6.5.2 Σχέσεις ιδιοκτησίας ... 88
6.5.3 Repositories και queries ... 88
6.5.4 DTO mapping ... 88
6.5.5 Συναλλαγές και dirty checking ... 88
6.5.6 Constraints και indexes ... 89
6.5.7 Ενημέρωση χιλιομέτρων ... 89
6.5.8 Υπολογισμός κατανάλωσης ... 89
6.5.9 Καταστάσεις εγγράφων και υπενθυμίσεων ... 89
6.5.10 Migrations ... 89
6.6 Διαχείριση αρχείων και φωτογραφιών ... 89
6.6.1 Κατηγορίες αρχείων ... 89
6.6.2 StorageService ... 90
6.6.3 Storage keys και filenames ... 90
6.6.4 Upload validation ... 90
6.6.5 Metadata ... 90
6.6.6 Ιδιωτική λήψη ... 90
6.6.7 Φωτογραφίες οχήματος ... 91
6.6.8 Αποτυχία κατά το upload ... 91
6.6.9 Durable διαγραφή ... 91
6.6.10 Reconciliation ... 91
6.7 Επεξεργασία και παρουσίαση αρχείων GPX ... 91
6.7.1 Ροή επεξεργασίας ... 91
6.7.2 GpxParser ... 92
6.7.3 Όρια ... 92
6.7.4 Tracks και segments ... 92
6.7.5 Απόσταση και διάρκεια ... 92
6.7.6 Υψομετρικά στοιχεία και waypoints ... 92

6.7.7 Route entity και endpoints ... 92
6.7.8 Παρουσίαση με Leaflet ... 93
6.8 Εισαγωγή, εξαγωγή και δημιουργία αναφορών ... 95
6.8.1 Εξαγωγή οχήματος ... 95
6.8.2 Streaming δημιουργία archive ... 95
6.8.3 Προεπισκόπηση εισαγωγής ... 95
6.8.4 Ασφάλεια ZIP ... 95
6.8.5 Εισαγωγή ... 96
6.8.6 Backup και restore λογαριασμού ... 96
6.8.7 PDF ιστορικού συντήρησης ... 96
6.9 Background εργασίες και μηχανισμοί αξιοπιστίας ... 96
6.9.1 Βασική αρχή ... 96
6.9.2 Email outbox ... 96
6.9.3 Storage deletion queue ... 97
6.9.4 Rollback tracking ... 97
6.9.5 Reconciliation ... 97
6.9.6 Concurrency και locking ... 97
6.9.7 Παρατηρησιμότητα ... 97
6.10 Αυτοματοποιημένες δοκιμές ... 98
6.10.1 Backend tests ... 98
6.10.2 Security tests ... 98
6.10.3 Storage και file tests ... 98
6.10.4 GPX tests ... 99
6.10.5 Import, export και PDF tests ... 99
6.10.6 Background worker tests ... 99
6.10.7 Frontend tests ... 99
6.10.8 Κάλυψη και CI ... 100
6.11 Ρυθμίσεις περιβάλλοντος και λειτουργία της εφαρμογής ... 100
6.11.1 Spring profiles ... 100
6.11.2 Development περιβάλλον ... 100
6.11.3 Production περιβάλλον ... 100
6.11.4 Typed configuration ... 100
6.11.5 ProductionConfigurationValidator ... 101
6.11.6 Βάση και migrations ... 101
6.11.7 Storage και email ... 101

6.11.8 Frontend configuration ..... 101
6.11.9 Container λειτουργία ..... 101
6.11.10 Μυστικά και logs ..... 101
6.12 Συνολική αποτίμηση της υλοποίησης ..... 102
6.12.1 Κάλυψη βασικών λειτουργιών ..... 102
6.12.2 Αρχιτεκτονική ..... 102
6.12.3 Ασφάλεια και αξιοπιστία ..... 102
6.12.4 Συντηρησιμότητα ..... 102
6.12.5 Περιορισμοί ..... 103
6.12.6 Συμπέρασμα ..... 103
7. Παρουσίαση του Moto-Vault ..... 104
7.1 Εισαγωγή ..... 104
7.2 Εγγραφή, σύνδεση και διαχείριση λογαριασμού ..... 104
7.2.1 Δημιουργία λογαριασμού ..... 104
7.2.2 Σύνδεση στην εφαρμογή ..... 105
7.2.3 Προφίλ, ταυτότητες και συνεδρίες ..... 106
7.2.4 Προτιμήσεις και χρήση αποθηκευτικού χώρου ..... 107
7.2.5 Διαγραφή λογαριασμού ..... 108
7.3 Αρχική σελίδα και πίνακας ελέγχου ..... 108
7.3.1 Ψηφιακό γκαράζ ..... 108
7.3.2 Άμεση πρόσβαση στις λειτουργίες ..... 109
7.3.3 Οικονομικά στοιχεία γκαράζ ..... 109
7.4 Διαχείριση μοτοσυκλετών ..... 110
7.4.1 Δημιουργία και επεξεργασία οχήματος ..... 110
7.4.2 Κοινό πλαίσιο και πλοήγηση ..... 111
7.4.3 Επισκόπηση οχήματος ..... 112
7.5 Ιστορικό συντήρησης ..... 112
7.5.1 Λίστα, αναζήτηση και κατηγορίες ..... 113
7.5.2 Δημιουργία εγγραφής ..... 113
7.5.3 Συνημμένα και συνδεδεμένα στοιχεία ..... 114
7.5.4 Χιλιόμετρα και αναφορά PDF ..... 114
7.6 Καταγραφή καυσίμων και στατιστικά κατανάλωσης ..... 115
7.6.1 Στατιστικά κατανάλωσης και κόστους ..... 115
7.6.2 Ιστορικό ανεφοδιασμών ..... 115
7.6.3 Δημιουργία ανεφοδιασμού ..... 116

7.6.4 Πλήρη και μερικά γεμίσματα ..... 116
7.7 Αξεσουάρ, ανταλλακτικά και έγγραφα ..... 117
7.7.1 Αξεσουάρ ..... 117
7.7.2 Ανταλλακτικά ..... 119
7.7.3 Έγγραφα ..... 119
7.7.4 Κοινά χαρακτηριστικά ..... 120
7.8 Υπενθυμίσεις ..... 121
7.8.1 Λίστα και καταστάσεις ..... 121
7.8.2 Τύποι υπενθυμίσεων ..... 121
7.8.3 Δημιουργία υπενθύμισης ..... 122
7.8.4 Σύνδεση με επισκόπηση και έγγραφα ..... 122
7.9 Διαδρομές GPX ..... 123
7.9.1 Λίστα, αναζήτηση και μεταφόρτωση ..... 123
7.9.2 Ενημερωτικό πλαίσιο ..... 123
7.9.3 Επιλογή διαδρομής, χάρτης και στατιστικά ..... 124
7.9.4 Υψομετρικό προφίλ ..... 125
7.9.5 Σημεία ενδιαφέροντος ..... 126
7.9.6 Συνδεδεμένες φωτογραφίες ..... 126
7.10 Φωτογραφίες και media υλικό ..... 126
7.10.1 Προβολή και διαχείριση ..... 126
7.10.2 Μεταφόρτωση και metadata ..... 127
7.10.3 Κύρια φωτογραφία ..... 127
7.10.4 Σύνδεση με GPX διαδρομές ..... 127
7.10.5 Ιδιωτικότητα ..... 127
7.11 Εισαγωγή, εξαγωγή και αναφορές ..... 127
7.11.1 Εξαγωγή οχήματος ..... 127
7.11.2 Προεπισκόπηση και εισαγωγή ..... 128
7.11.3 Backup και restore λογαριασμού ..... 129
7.11.4 Αναφορά ιστορικού συντήρησης ..... 129
7.11.5 Διαφορά backup και PDF ..... 130
7.12 Προσαρμοστικότητα και πολυγλωσσική υποστήριξη ..... 130
7.12.1 Προβολή γκαράζ σε κινητή συσκευή ..... 130
7.12.2 Προβολή οχήματος σε μικρή οθόνη ..... 131
7.12.3 Διατήρηση λειτουργικότητας ..... 132
7.12.4 Γλώσσες ..... 132

7.12.5 Μονάδες και τοπική μορφοποίηση ..... 132
7.13 Συνολική αποτίμηση της διεπαφής ..... 132
7.13.1 Συνοχή και οργάνωση ..... 132
7.13.2 Διασύνδεση λειτουργιών ..... 133
7.13.3 Προσαρμοστικότητα και σημεία προς αξιολόγηση ..... 133
7.13.4 Μετάβαση στην αξιολόγηση ..... 133
8. Αξιολόγηση και Συζήτηση ..... 134
8.1 Εισαγωγή ..... 134
8.2 Σχεδιασμός της αξιολόγησης και συμμετέχοντες ..... 134
8.3 Εργαλείο και διαδικασία αξιολόγησης ..... 134
8.4 Αποτελέσματα αξιολόγησης ..... 135
8.4.1 Αποτελέσματα SUS ..... 135
8.4.2 Πρόσθετες ερωτήσεις ..... 136
8.5 Συζήτηση και προτάσεις βελτίωσης ..... 137
8.5.1 Αυτοματοποίηση καταχωρίσεων ..... 138
8.5.2 Εφαρμογή για κινητές συσκευές ..... 138
8.5.3 Διασύνδεση με συνεργεία ..... 138
8.6 Περιορισμοί και συνολική αποτίμηση ..... 138
9. Συμπεράσματα και Προοπτικές ..... 139
9.1 Εισαγωγή ..... 139
9.2 Επίτευξη στόχων και απάντηση στα ερευνητικά ερωτήματα ..... 139
9.2.1 Κύριο ερευνητικό ερώτημα ..... 139
9.2.2 Λειτουργικές και μη λειτουργικές απαιτήσεις ..... 139
9.2.3 Αρχιτεκτονική και τεχνικές επιλογές ..... 140
9.2.4 Ενοποίηση λειτουργιών ..... 140
9.2.5 Αντιλαμβανόμενη χρηστικότητα ..... 140
9.2.6 Περιορισμοί και δυνατότητες βελτίωσης ..... 140
9.2.7 Συνολική επίτευξη στόχων ..... 141
9.3 Συνεισφορά της εργασίας ..... 141
9.3.1 Σχεδιαστική συνεισφορά ..... 141
9.3.2 Τεχνική συνεισφορά ..... 141
9.3.3 Συνεισφορά ως προς την ασφάλεια και αξιοπιστία ..... 141
9.3.4 Η πρακτική συνεισφορά ..... 141
9.3.5 Μεθοδολογική συνεισφορά ..... 142
9.3.6 Η εμπειρική συνεισφορά ..... 142

9.4 Οι Περιορισμοί της εργασίας ..... 142
9.4.1 Περιορισμένο λειτουργικό εύρος ..... 142
9.4.2 Περιορισμένη αυτοματοποίηση ..... 143
9.4.3 Απουσία mobile εφαρμογής ..... 143
9.4.4 Περιορισμένες εξωτερικές διασυνδέσεις ..... 143
9.4.5 Περιορισμοί μοντέλου δεδομένων και δικαιωμάτων ..... 143
9.4.6 Κλιμάκωση και λειτουργία ..... 143
9.4.7 Περιορισμοί αξιολόγησης ..... 144
9.4.8 Απουσία συγκριτικής αξιολόγησης ..... 144
9.4.9 Συντήρηση και εξέλιξη ..... 144
9.5 Μελλοντικές επεκτάσεις ..... 144
9.5.1 Αυτοματοποίηση καταχωρίσεων ..... 144
9.5.2 Εφαρμογή για κινητές συσκευές ..... 145
9.5.3 Δεδομένα οχήματος και εξωτερικές διασυνδέσεις ..... 145
9.5.4 Συνεργεία, κοινή χρήση και πολλαπλοί ρόλοι ..... 146
9.5.5 Προηγμένες ειδοποιήσεις ..... 146
9.5.6 Προηγμένη ανάλυση και αναφορές ..... 147
9.5.7 Προσβασιμότητα, γλώσσες και απόδοση πολυμέσων ..... 147
9.5.8 Ασφάλεια, ποιότητα και τεχνική κλιμάκωση ..... 148
9.5.9 Ενίσχυση της αξιολόγησης ..... 148
9.5.10 Προτεραιότητες επόμενων βημάτων ..... 149
9.6 Τελικά συμπεράσματα ..... 149
10. Βιβλιογραφία ..... 150
Επιστημονικές πηγές ..... 150
Πρότυπα και τεχνικές προδιαγραφές ..... 150
Θεσμικές πηγές ..... 150
Υφιστάμενες εφαρμογές και υπηρεσίες ..... 151

# 1. Εισαγωγή

## 1.1 Ο σκοπός της Διπλωματικής Εργασίας

Η κατοχή και χρήση μιας μοτοσυκλέτας δημιουργεί με την πάροδο του χρόνου πολλές διαφορετικές πληροφορίες και ανάγκες. Σε αυτές περιλαμβάνονται οι εργασίες συντήρησης, οι επισκευές, οι ανεφοδιασμοί καυσίμου, τα έξοδα, τα ανταλλακτικά, ο πρόσθετος εξοπλισμός, οι φωτογραφίες και οι διαδρομές που έχει πραγματοποιήσει ο οδηγός. Οι πληροφορίες αυτές είναι χρήσιμες για την παρακολούθηση της κατάστασης του οχήματος και για τον προγραμματισμό μελλοντικών εργασιών. Παρόλα αυτά, συχνά δεν καταγράφονται σε ένα ενιαίο σύστημα και η παρακολούθηση τους ανά τα έτη καθίσταται πολύ δύσκολη.

Στην πράξη, ένας ιδιοκτήτης μπορεί να κρατά τις αποδείξεις σε έντυπη μορφή, να σημειώνει κάποιες εργασίες στο κινητό του και να αποθηκεύει τις φωτογραφίες στον υπολογιστή του ή σε cloud υπηρεσίες. Οι ανεφοδιασμοί βενζίνης ρεαλιστικά σχεδόν ποτέ δεν καταγράφονται, ενώ οι υπενθυμίσεις για την επόμενη συντήρηση μπορεί να καταγράφονται σε μια εφαρμογή ημερολογίου. Έτσι, τα δεδομένα μένουν σκόρπια και χάνεται χρόνος στην αναζήτησή τους. Το πρόβλημα μεγαλώνει όσο περνούν τα χρόνια και μαζεύονται περισσότερες εγγραφές.

Το σκόρπισμα αυτής της πληροφορίας, γίνεται πιο έντονη όσο αυξάνεται η ηλικία του οχήματος και ο αριθμός των καταχωρίσεων. Η εύρεση της ημερομηνίας μιας προηγούμενης εργασίας ή της χιλιομετρικής ένδειξης κατά την οποία αντικαταστάθηκε ένα εξάρτημα δεν είναι πάντα εύκολη. Το ίδιο ισχύει και για τον υπολογισμό του συνολικού κόστους συντήρησης ή της μέσης κατανάλωσης καυσίμου. Όταν τα δεδομένα βρίσκονται σε διαφορετικά σημεία, ο χρήστης δεν έχει άμεσα μια συνολική εικόνα του οχήματός του.

Η παρακολούθηση της συντήρησης είναι ιδιαίτερα σημαντική, επειδή αρκετές εργασίες βασίζονται είτε στον χρόνο είτε στα χιλιόμετρα που έχει διανύσει η μοτοσυκλέτα. Για παράδειγμα, μια αλλαγή λαδιών μπορεί να πρέπει να πραγματοποιηθεί μετά από συγκεκριμένο αριθμό χιλιομέτρων, ενώ άλλες εργασίες μπορεί να επαναλαμβάνονται ανά έτος. Χωρίς οργανωμένη καταγραφή, είναι πιθανό κάποια εργασία να καθυστερήσει ή να ξεχαστεί.

Ένα ολοκληρωμένο ιστορικό έχει αξία και πέρα από την καθημερινή χρήση, καθώς επιτρέπει στον ιδιοκτήτη να γνωρίζει τι έχει γίνει στη μοτοσυκλέτα, ποια εξαρτήματα έχουν τοποθετηθεί και πόσο έχει κοστίσει συνολικά η χρήση της. Μπορεί επίσης να προσφέρει καλύτερη ενημέρωση σε ένα συνεργείο ή σε έναν μελλοντικό αγοραστή. Το ψηφιακό ιστορικό δεν αντικαθιστά τα επίσημα παραστατικά και τα στοιχεία των συνεργείων, αλλά μπορεί να τα οργανώνει και να τα συγκεντρώνει σε ένα κοινό σημείο.

Υπάρχουν εφαρμογές που καλύπτουν μέρος αυτών των αναγκών. Ορισμένες επικεντρώνονται στην κατανάλωση καυσίμου, άλλες στο social media κομμάτι και άλλες στην καταγραφή διαδρομών. Η χρήση πολλών διαφορετικών εφαρμογών, όμως, δεν λύνει πλήρως το πρόβλημα. Τα δεδομένα παραμένουν χωρισμένα και συνήθως δεν συνδέονται μεταξύ τους με βάση το ίδιο όχημα, τη χιλιομετρική ένδειξη και τη χρονική στιγμή της καταχώρισης.

Το ζήτημα που εξετάζει η παρούσα εργασία είναι η έλλειψη ενός ενιαίου και εύχρηστου ψηφιακού εργαλείου, στο οποίο θα μπορεί ο κάτοχος μιας μοτοσυκλέτας να οργανώνει τις βασικές πληροφορίες του οχήματός του. Η ανάγκη αυτή περιλαμβάνει τόσο την

αποθήκευση των δεδομένων όσο και την παρουσίασή τους με τρόπο που να είναι κατανοητός και πρακτικός για τον χρήστη.

Για την αντιμετώπιση του προβλήματος σχεδιάστηκε και αναπτύχθηκε το Moto-Vault. Πρόκειται για μια διαδικτυακή πλατφόρμα που συγκεντρώνει πληροφορίες συντήρησης, ανεφοδιασμών, κόστους, εξαρτημάτων, φωτογραφιών, υπενθυμίσεων και διαδρομών. Κάθε καταχώριση συνδέεται με συγκεκριμένη μοτοσυκλέτα, ώστε να δημιουργείται σταδιακά ένα οργανωμένο ψηφιακό ιστορικό.

Για την ανάπτυξη και την αξιολόγησή, εφαρμόστηκαν οι αρχές της μεθοδολογίας Design Science Research (Hevner κ.ά., 2004; Peffers κ.ά., 2007).

## 1.2 Σκοπός της Διπλωματικής Εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός, η ανάπτυξη και η αξιολόγηση μιας διαδικτυακής πλατφόρμας για τη διαχείριση πληροφοριών μοτοσυκλέτας.

Η πλατφόρμα, με την ονομασία Moto-Vault, έχει ως βασικό στόχο να συγκεντρώνει σε ένα σημείο πληροφορίες που συνήθως βρίσκονται σκόρπιες. Σε αυτές περιλαμβάνονται το ιστορικό συντήρησης, οι ανεφοδιασμοί, τα έξοδα, τα ανταλλακτικά, τα αξεσουάρ, τα έγγραφα, οι υπενθυμίσεις, οι φωτογραφίες και οι διαδρομές.

Η εργασία δεν περιορίζεται στην αποθήκευση δεδομένων, καθώς ο στόχος είναι οι πληροφορίες να οργανώνονται ανά μοτοσυκλέτα και να παρουσιάζονται με τρόπο απλό και κατανοητό στον μέσο χρήστη, ώστε να μπορεί να βρίσκει εύκολα ότι χρειάζεται και να έχει συνολική εικόνα του οχήματός του.

Ταυτόχρονα, εξετάζεται αν μια εφαρμογή με πολλές διαφορετικές λειτουργίες μπορεί να παραμένει εύχρηστη και συνεπής. Για τον λόγο αυτό, το τελικό σύστημα αξιολογείται με το ερωτηματολόγιο System Usability Scale (SUS) (Brooke, 1996).

Οι βασικοί στόχοι της εργασίας είναι:

1. Η μελέτη του προβλήματος της διάσπαρτης καταγραφής πληροφοριών μοτοσυκλέτας.
2. Η αναγνώριση των βασικών αναγκών των χρηστών.
3. Η καταγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων.
4. Ο σχεδιασμός και η ανάπτυξη μιας ολοκληρωμένης διαδικτυακής εφαρμογής.
5. Η ασφαλής οργάνωση των δεδομένων και των αρχείων κάθε χρήστη.
6. Η υποστήριξη των βασικών λειτουργιών διαχείρισης μοτοσυκλέτας.
7. Η επαλήθευση της ορθής λειτουργίας του συστήματος μέσω δοκιμών.
8. Η αξιολόγηση της χρηστικότητας του Moto-Vault από πραγματικούς χρήστες.

## 1.3 Αντικείμενο και εύρος της Διπλωματικής Εργασίας

Το αντικείμενο της εργασίας είναι η ανάπτυξη του Moto-Vault ως ολοκληρωμένης διαδικτυακής εφαρμογής για ιδιώτες κατόχους μοτοσυκλετών.

Η εργασία περιλαμβάνει:

- τη μελέτη του προβλήματος και των διαθέσιμων λύσεων,
- την ανάλυση απαιτήσεων,
- τον σχεδιασμό της αρχιτεκτονικής και της βάσης δεδομένων,
- την ανάπτυξη του backend και του frontend,
- τη διαχείριση χρηστών, οχημάτων και σχετικών καταχωρίσεων,
- την αποθήκευση φωτογραφιών και εγγράφων,
- την επεξεργασία και προβολή διαδρομών GPX,
- την εισαγωγή και εξαγωγή δεδομένων,
- τη δημιουργία αυτοματοποιημένων δοκιμών,
- τη ρύθμιση του συστήματος για λειτουργία σε διαδικτυακό περιβάλλον,
- την αξιολόγηση της χρηστικότητας.

Το Moto-Vault απευθύνεται σε ιδιώτες χρήστες που θέλουν να οργανώνουν τις πληροφορίες μιας ή περισσότερων μοτοσυκλετών. Δεν αποτελεί εφαρμογή συνεργείου, σύστημα διαχείρισης στόλου ή επίσημο μητρώο συντήρησης. Επίσης, δεν αντικαθιστά τα παραστατικά, το βιβλίο service ή τα επίσημα στοιχεία κατασκευαστών και δημόσιων φορέων (πχ. Gov Wallet).

Στη περίπτωση της παρούσας εργασίας δεν περιλαμβάνονται:

- επαγγελματικοί λογαριασμοί συνεργείων,
- σύνδεση με επίσημες βάσεις δεδομένων κατασκευαστών,
- αυτόματη ανάκτηση στοιχείων μέσω VIN,
- εφαρμογή για Android ή iOS,
- πληρωμές και συνδρομητικά πακέτα,
- αυτόματη καταχώριση δεδομένων μέσω OCR ή τεχνητής νοημοσύνης,
- σύνδεση με OBD ή telematics,
- δοκιμές μεγάλης κλίμακας με μεγάλο αριθμό ταυτόχρονων χρηστών.

Οι λειτουργίες αυτές αποτελούν καλές και προγραμματισμένες μελλοντικές επεκτάσεις και παρουσιάζονται στο Κεφάλαιο 9.

Το εύρος της εργασίας περιορίζεται επομένως στον σχεδιασμό, την υλοποίηση και την αρχική αξιολόγηση μιας πλήρους αλλά προσωπικής πλατφόρμας διαχείρισης πληροφοριών μοτοσυκλέτας.

## 1.4 Δομή

Η διπλωματική εργασία οργανώνεται σε εννέα κεφάλαια, τα οποία παρουσιάζουν το πρόβλημα, τη μεθοδολογία, τον σχεδιασμό, την υλοποίηση και την αξιολόγηση του Moto-Vault.

**Κεφάλαιο 1**: Εισαγωγή. Ορίζει το πλαίσιο της εργασίας και περιγράφεται το πρόβλημα της χύμα καταγραφής πληροφοριών μοτοσυκλέτας. Παρουσιάζονται επίσης ο σκοπός, οι επιμέρους στόχοι και το εύρος της διπλωματικής εργασίας.

**Κεφάλαιο 2**: Ανάλυση Τρέχοντος Επιπέδου. Δίνονται οι βασικές έννοιες που σχετίζονται με τη διαχείριση πληροφοριών οχήματος και εξετάζονται υπάρχουσες παρόμοιες εφαρμογές. Η σύγκριση χρησιμοποιείται για τον εντοπισμό των αναγκών και των λειτουργικών κενών που επηρέασαν τον σχεδιασμό του Moto-Vault.

**Κεφάλαιο 3**: Μεθοδολογία. Ερευνητική προσέγγιση μέσω σχεδιασμού, μεθοδολογίας Design Science Research Methodology, τα ερευνητικά ερωτήματα και ο τρόπος εφαρμογής των σταδίων της κατά την ανάπτυξη και την αξιολόγηση της εφαρμογής.

**Κεφάλαιο 4**: Ανάλυση και Σχεδιασμός. Τύπος χρηστών και τα βασικά σενάρια χρήσης, οι λειτουργικές και μη λειτουργικές απαιτήσεις, η αρχιτεκτονική του συστήματος, το μοντέλο δεδομένων, το REST API, το UI/UX και η εγκατάσταση της εφαρμογής.

**Κεφάλαιο 5**: Τεχνολογίες και Εργαλεία Ανάπτυξης. Για πιο λόγο επιλέχθηκαν οι τεχνολογίες για το backend, το frontend, τη βάση δεδομένων, την αποθήκευση αρχείων, τους χάρτες, τις δοκιμές, τη συνεχή ολοκλήρωση και το περιβάλλον παραγωγικής λειτουργίας.

**Κεφάλαιο 6**: Υλοποίηση. Εστίαση στο προγραμματιστικό κομμάτι και παρουσίαση υλοποίησης του backend και του frontend. Εξετάζονται η αυθεντικοποίηση και η εξουσιοδότηση, η διαχείριση δεδομένων και αρχείων, η επεξεργασία διαδρομών GPX, οι εργασίες παρασκηνίου και οι αυτοματοποιημένες δοκιμές(testing).

**Κεφάλαιο 7**: Παρουσίαση του Moto-Vault. Παρουσιάζεται το τελικό σύστημα από την πλευρά του χρήστη (ui walkthrough). Μέσα από στιγμιότυπα οθόνης περιγράφονται οι βασικές σελίδες, οι κύριες ροές χρήσης και οι λειτουργίες που υποστηρίζει η πλατφόρμα.

**Κεφάλαιο 8**: Αξιολόγηση και Συζήτηση. Περιγράφονται ο σχεδιασμός και η διαδικασία της αξιολόγησης, τα χαρακτηριστικά των συμμετεχόντων και το ερωτηματολόγιο System Usability Scale και αναλύονται τα αποτελέσματα, οι πρόσθετες ερωτήσεις, οι προτάσεις βελτίωσης και οι περιορισμοί της αξιολόγησης.

**Κεφάλαιο 9**: Συμπεράσματα και Προοπτικές. Συνοψισμός της συνεισφορά της εργασίας και του βαθμού επίτευξης των αρχικών στόχων, ενώ προτείνει συγκεκριμένες τεχνικές επεκτάσεις για το μέλλον.

Μετά τα κεφάλαια ακολουθεί η βιβλιογραφία.

# 2. Ανάλυση Τρέχοντος Επιπέδου

## 2.1 Βασικές Έννοιες και Ορισμοί

Πριν ξεκινήσουμε την παρουσίαση του ανταγωνισμού, χρειάζεται να διευκρινιστούν μερικές βασικές έννοιες της εργασίας. Συγκεκριμένα εξετάζονται η διαχείριση πληροφοριών, η συντήρηση οχήματος, η παρακολούθηση κόστους, το φωτογραφικό υλικό, τα αρχεία διαδρομών και το ψηφιακό ιστορικό μοτοσυκλέτας.

### 2.1.1 Διαχείριση πληροφοριών του οχήματος

Η διαχείριση πληροφοριών οχήματος συλλέγει, αποθηκεύει, οργανώνει, ψάχνει και παρουσιάζει τα δεδομένα του οχήματος. Αυτά μπορεί να είναι βασικά χαρακτηριστικά, χιλιόμετρα, εργασίες συντήρησης, επισκευές, ανεφοδιασμούς, έξοδα, ανταλλακτικά, αξεσουάρ, έγγραφα, φωτογραφίες και διαδρομές.

Η απλή αποθήκευση δεν είναι αρκετή. Οι πληροφορίες πρέπει να συνδέονται με το σωστό όχημα και να συνοδεύονται από στοιχεία που βοηθούν τον χρήστη να καταλάβει το ιστορικό τους. Για παράδειγμα, ένα σέρβις είναι χρήσιμο όταν περιλαμβάνει ημερομηνία, οδόμετρο, περιγραφή και κόστος.

Στο Moto-Vault κάθε καταχώριση συνδέεται με τον λογαριασμό του χρήστη και με συγκεκριμένη μοτοσυκλέτα. Με αυτόν τον τρόπο οι διαφορετικές κατηγορίες δεδομένων οργανώνονται γύρω από το ίδιο όχημα.

### 2.1.2 Το Ιστορικό συντήρησης

Το ιστορικό συντήρησης είναι η λίστα με όλα τα πράγματα που έγιναν στο όχημα και περιλαμβάνει προγραμματισμένες εργασίες, επισκευές, αλλαγές υγρών, αντικαταστάσεις εξαρτημάτων και άλλες δουλειές.

Μια καταχώριση συντήρησης περιλαμβάνει συνήθως την ημερομηνία, τη χιλιομετρική ένδειξη και την περιγραφή της εργασίας. Μπορεί επίσης να περιλαμβάνει το κόστος, το συνεργείο, παρατηρήσεις, φωτογραφίες και στοιχεία για τα ανταλλακτικά που χρησιμοποιήθηκαν.

Το ιστορικό βοηθά τον ιδιοκτήτη να γνωρίζει τι έχει γίνει στη μοτοσυκλέτα, να προγραμματίζει επόμενες εργασίες και να παρακολουθεί τα σχετικά έξοδα. Μπορεί ακόμη να προσφέρει χρήσιμες πληροφορίες σε ένα συνεργείο ή σε έναν μελλοντικό αγοραστή.

Το ψηφιακό αυτό ιστορικό δεν αποτελεί επίσημη πιστοποίηση των service. Η ακρίβειά του εξαρτάται από τα δεδομένα που καταχωρίζει ο χρήστης και λειτουργεί συμπληρωματικά προς τα επίσημα στοιχεία συντήρησης.

### 2.1.3 Προληπτική και διορθωτική συντήρηση

Η προληπτική συντήρηση πραγματοποιείται πριν εμφανιστεί κάποια βλάβη. Στηρίζεται συνήθως στον χρόνο, στα χιλιόμετρα ή στις οδηγίες του κατασκευαστή. Παραδείγματα είναι η αλλαγή λαδιών, ο έλεγχος της αλυσίδας και η αντικατάσταση υγρών σε συγκεκριμένα χρονικά διαστήματα.

Ένα service πραγματοποιείται μετά την εμφάνιση προβλήματος και έχει ως στόχο την αποκατάσταση της σωστής λειτουργίας του οχήματος. Παράδειγμα αποτελεί η αντικατάσταση ενός εξαρτήματος που παρουσίασε βλάβη.

Στην πράξη, μια επίσκεψη στο συνεργείο μπορεί να περιλαμβάνει και τις δύο κατηγορίες εργασιών. Για το Moto-Vault μεγαλύτερη σημασία έχει η σαφής καταγραφή του τι έγινε, πότε έγινε και για ποιον λόγο.

### 2.1.4 Καταγραφή κόστους και κατανάλωσης καυσίμου

Το μακροχρόνιο κόστος χρήσης μιας μοτοσυκλέτας περιλαμβάνει καύσιμα, συντήρηση, επισκευές, ανταλλακτικά, αξεσουάρ και άλλα σχετικά έξοδα. Η συνεπής και οργανωμένη καταγραφή βοηθά τον χρήστη να έχει πιο καθαρή εικόνα της συνολικής οικονομικής επιβάρυνσης.

Μια εγγραφή ανεφοδιασμού μπορεί να περιλαμβάνει την ημερομηνία, τα χιλιόμετρα, τα λίτρα καυσίμου, το κόστος ανά λίτρο και το συνολικό ποσό. Για τον υπολογισμό της μέσης κατανάλωσης, πρέπει να υπάρχουν συνεχόμενες καταχωρίσεις.

Η ακρίβεια των αποτελεσμάτων εξαρτάται από την ποιότητα των δεδομένων. Αν λείπουν ανεφοδιασμοί ή έχουν καταχωριστεί λανθασμένα χιλιόμετρα, οι υπολογισμοί μπορεί να μην ανταποκρίνονται στην πραγματική κατανάλωση.

Στο Moto-Vault το κόστος συνδέεται με ανεφοδιασμούς, εργασίες συντήρησης, ανταλλακτικά και αξεσουάρ. Έτσι μπορούν να παρουσιαστούν συγκεντρωτικά στοιχεία και βασικές κατηγορίες εξόδων.

### 2.1.5 Φωτογραφικό υλικό και έγγραφα

Οι πληροφορίες μιας μοτοσυκλέτας δεν έχουν μόνο αριθμητική ή περιγραφική μορφή. Οι φωτογραφίες μπορούν να δείχνουν την κατάσταση ενός εξαρτήματος, μια εργασία πριν και μετά την ολοκλήρωσή της, μια ζημιά ή τον εξοπλισμό που έχει τοποθετηθεί.

Στο Moto-Vault οι φωτογραφίες μπορούν να αποθηκεύονται ως γενικό υλικό της μοτοσυκλέτας ή να συνδέονται με εργασίες συντήρησης, ανταλλακτικά και αξεσουάρ. Η σύνδεση αυτή κάνει πιο εύκολη τη μελλοντική αναζήτηση και δίνει περισσότερο νόημα σε κάθε εικόνα.

Η εφαρμογή υποστηρίζει επίσης την αποθήκευση εγγράφων που σχετίζονται με τη μοτοσυκλέτα. Τα έγγραφα μπορούν να συνοδεύονται από βασικά στοιχεία και ημερομηνίες λήξης, ώστε να οργανώνονται μαζί με τις υπόλοιπες πληροφορίες του οχήματος.

### 2.1.6 Αρχεία διαδρομών GPX

Ένα αρχείο GPX (GPS Exchange Format σε μορφή XML) μπορεί να περιλαμβάνει σημεία, διαδρομές και ίχνη κίνησης. Η έκδοση 1.1 οργανώνει το περιεχόμενο σε μεταδεδομένα, waypoints, routes και tracks (Topografix, 2004).

Στο Moto-Vault ο χρήστης μπορεί να αποθηκεύει ένα αρχείο GPX και να βλέπει τη διαδρομή σε χάρτη. Η λειτουργία δεν χρησιμοποιείται για ζωντανή πλοήγηση ή αυτόματη καταγραφή. Σκοπός της είναι η αποθήκευση και η μεταγενέστερη προβολή διαδρομών.

### 2.1.7 Ψηφιακό ιστορικό μοτοσυκλέτας

Στην παρούσα εργασία, ο όρος ψηφιακό ιστορικό μοτοσυκλέτας περιγράφει το σύνολο των πληροφοριών που αποθηκεύονται για ένα συγκεκριμένο όχημα. Σε αυτό περιλαμβάνονται οι εργασίες συντήρησης, οι ανεφοδιασμοί, τα έξοδα, τα ανταλλακτικά, τα αξεσουάρ, τα έγγραφα, οι φωτογραφίες, οι υπενθυμίσεις και οι διαδρομές.

Το ιστορικό δημιουργείται από τον ίδιο τον χρήστη και δεν αντικαθιστά το βιβλίο service, τα παραστατικά των συνεργείων, τα στοιχεία του κατασκευαστή ή τα κρατικά μητρώα. Λειτουργεί ως προσωπικό εργαλείο οργάνωσης και παρακολούθησης.

Αυτή η ιδέα αποτελεί τον πυρήνα του MotoVault, καθώς συνδέει όλες τις λειτουργίες της εφαρμογής γύρω από τη μοτοσυκλέτα, προσφέροντας μια συνολική εικόνα της χρήσης της.

## 2.2 Ψηφιακή καταγραφή πληροφοριών οχήματος

Η μετάβαση από τα παραδοσιακά μέσα παρακολούθησης (έντυπα τιμολόγια, χειρόγραφες σημειώσεις) σε μια ψηφιακή πλατφόρμα δεν είναι απλώς ψηφιοποίηση (digitalization) φυσικών πραγμάτων. Απαιτεί τη δόμηση ενός σχεσιακού μοντέλου δεδομένων ικανού να συσχετίζει διάφορες πληροφορίες.

Στο πλαίσιο μιας μοτοσυκλέτας, η ψηφιακή καταγραφή επιτρέπει την ταυτόχρονη διασύνδεση των χιλιομέτρων με προγραμματισμένες εργασίες, ημερομηνίες λήξης εγγράφων (πχ. ΚΤΕΟ, ασφάλεια) και αρχεία GPX. Η οργάνωση μετατρέπει τα δεδομένα σε αξιοποιήσιμη πληροφορία, διευκολύνοντας την αναζήτηση και την εξαγωγή στατιστικών.

### 2.2.1 Από την έντυπη στην ψηφιακή καταγραφή

Η παραδοσιακή καταγραφή βασίζεται σε βιβλία συντήρησης, τιμολόγια, αποδείξεις και σημειώσεις του ιδιοκτήτη. Τα έγγραφα αυτά είναι χρήσιμα, αλλά μπορούν να χαθούν, να φθαρούν ή να μην είναι διαθέσιμα όταν τα χρειάζεται ο χρήστης. Η αναζήτηση μιας παλαιότερης εργασίας απαιτεί επίσης χειροκίνητο έλεγχο.

Ίσως τα υπολογιστικά φύλλα (Excel) και οι εφαρμογές σημειώσεων μπορούν να χρησιμοποιηθούν για μια βασική καταγραφή, αλλά δεν είναι φτιαγμένα ειδικά για πληροφορίες οχήματος. Ο χρήστης πρέπει να δημιουργήσει μόνος του τη δομή, τους υπολογισμούς και τον τρόπο με τον οποίο θα συνδέονται τα δεδομένα.

Μια εξειδικευμένη εφαρμογή μπορεί να παρέχει έτοιμα πεδία και βασικούς ελέγχους εγκυρότητας. Μπορεί, για παράδειγμα, να ελέγχει το κόστος, τις ημερομηνίες και το οδόμετρο, καθώς και να συνδέει κάθε καταχώριση με συγκεκριμένο όχημα.

Τον Απρίλιο του 2025, η Ευρωπαϊκή Επιτροπή πρότεινε ψηφιακά πιστοποιητικά τεχνικού ελέγχου και ψηφιακά έγγραφα ταξινόμησης. Αν και οι προτάσεις αυτές αφορούν επίσημα έγγραφα και όχι το προσωπικό ιστορικό συντήρησης, δείχνουν τη γενικότερη στροφή προς την ηλεκτρονική διαχείριση πληροφοριών οχήματος (European Commission, 2025a; European Commission, 2025b).

### 2.2.2 Οργάνωση και σύνδεση των δεδομένων

Σε ένα ψηφιακό σύστημα, το όχημα μπορεί να λειτουργεί ως το βασικό σημείο γύρω από το οποίο οργανώνονται όλες οι καταχωρίσεις. Μια εργασία συντήρησης μπορεί να συνδέεται με ημερομηνία, χιλιόμετρα, κόστος, φωτογραφίες και ανταλλακτικά. Αντίστοιχα, ένας ανεφοδιασμός συνδέεται με ποσότητα καυσίμου, τιμή και χιλιομετρική ένδειξη.

Η δομή αυτή επιτρέπει τη δημιουργία μιας γενικότερης εικόνας της μοτοσυκλέτας, όπου ο χρήστης μπορεί να βλέπει πότε έγινε μια εργασία, πότε τοποθετήθηκε ένα εξάρτημα και τα έξοδα μέσα στον χρόνο.

Όταν τα δεδομένα είναι σωστά οργανωμένα, μπορούν να χρησιμοποιηθούν και για πιο σύνθετους σκοπούς. Ορισμένες σύγχρονες μελέτες συνδυάζουν το ιστορικό συντήρησης, τα στοιχεία του οχήματος και τα χιλιόμετρα για να προβλέψουν μελλοντικές εργασίες συντήρησης. Οι μελέτες αυτές είναι πιο σύνθετες από το Moto-Vault, αλλά βασίζονται και αυτές σε σωστά και οργανωμένα δεδομένα (Chen κ.ά., 2024a; Chen κ.ά., 2024b).

### 2.2.3 Οφέλη για τον ιδιοκτήτη

Ο ιδιοκτήτης μπορεί να βρίσκει τις βασικές πληροφορίες της μοτοσυκλέτας άμεσα χωρίς να ελέγχει διαφορετικά αρχεία και εφαρμογές.

Η ψηφιακή καταγραφή βοηθά επίσης στον προγραμματισμό της συντήρησης. Η ημερομηνία και η χιλιομετρική ένδειξη μιας προηγούμενης εργασίας μπορούν να χρησιμοποιηθούν για τη δημιουργία υπενθύμισης. Οι υπενθυμίσεις δεν αντικαθιστούν τις οδηγίες του κατασκευαστή ή την αξιολόγηση ενός τεχνικού, αλλά βοηθούν τον χρήστη να μην ξεχνά προγραμματισμένες εργασίες.

Ένα ακόμη όφελος είναι η καλύτερη εικόνα του κόστους. Οι ανεφοδιασμοί βενζίνης, οι εργασίες συντήρησης και οι αγορές εξαρτημάτων μπορούν να παρουσιαστούν συγκεντρωτικά, ώστε ο χρήστης να καταλαβαίνει καλύτερα τα συνολικά έξοδα μιας μηχανής.

Οι φωτογραφίες και τα έγγραφα προσθέτουν χρήσιμες πληροφορίες στις καταχωρίσεις. Μια φωτογραφία μπορεί να δείχνει μια επισκευή ή την κατάσταση ενός εξαρτήματος, ενώ ένα έγγραφο μπορεί να αφορά την ασφάλεια, τον τεχνικό έλεγχο ή άλλο στοιχείο του οχήματος.

Τέλος, ένα οργανωμένο ιστορικό μπορεί να βοηθήσει κατά τη μεταβίβαση της μοτοσυκλέτας. Παρ’ όλα αυτά, οι εγγραφές ενός προσωπικού συστήματος δεν αποτελούν επίσημη απόδειξη και πρέπει να συνοδεύονται από τα αντίστοιχα παραστατικά όπου αυτό είναι απαραίτητο.

### 2.2.4 Ποιότητα και αξιοπιστία των πληροφοριών

Η ποιότητα του ψηφιακού ιστορικού εξαρτάται από τα δεδομένα που εισάγει ο χρήστης. Η εφαρμογή μπορεί να ελέγχει αν ένα πεδίο είναι κενό, αν ένα ποσό είναι αρνητικό ή αν μια τιμή έχει σωστή μορφή. Δεν μπορεί όμως να επιβεβαιώσει πάντα ότι η ημερομηνία, το κόστος ή τα χιλιόμετρα ανταποκρίνονται στην πραγματικότητα.

Η πληρότητα είναι επίσης σημαντική. Αν λείπει ένας ανεφοδιασμός, ο υπολογισμός κατανάλωσης μπορεί να είναι ανακριβής. Αν δεν ενημερωθούν τα χιλιόμετρα, μια σχετική υπενθύμιση μπορεί να μην ανταποκρίνεται στην πραγματική κατάσταση.

Για αυτό τα στατιστικά μιας προσωπικής εφαρμογής πρέπει να παρουσιάζονται ως αποτελέσματα που βασίζονται στις διαθέσιμες καταχωρίσεις. Το ιστορικό έχει οργανωτικό χαρακτήρα και δεν πρέπει να εμφανίζεται ως επίσημο ή αμετάβλητο μητρώο.

### 2.2.5 Προστασία δεδομένων και ασφάλεια

Ένα διαδικτυακό σύστημα πληροφοριών οχήματος μπορεί να αποθηκεύει προσωπικά στοιχεία, φωτογραφίες, έγγραφα και διαδρομές. Ένα αρχείο GPX, για παράδειγμα, μπορεί να αποκαλύπτει συχνές τοποθεσίες ή το σημείο εκκίνησης μιας διαδρομής.

Για αυτό απαιτούνται αυθεντικοποίηση και έλεγχοι πρόσβασης. Το backend πρέπει να ελέγχει ότι ο χρήστης έχει δικαίωμα πρόσβασης στη μοτοσυκλέτα και στην καταχώριση που ζητά. Ο έλεγχος αυτός είναι απαραίτητος σε λειτουργίες ανάγνωσης, ενημέρωσης και διαγραφής.

Η ασφαλής μεταφορά μέσω HTTPS, η προστασία των διαπιστευτηρίων και η σωστή διαχείριση των αρχείων αποτελούν επίσης βασικές απαιτήσεις. Στο Moto-Vault κάθε καταχώριση συνδέεται με τον κάτοχο του αντίστοιχου οχήματος, ενώ τα αρχεία αποθηκεύονται χωριστά από τα δομημένα δεδομένα της βάσης.

### 2.2.6 Μεταφορά και εξαγωγή δεδομένων

Η δυνατότητα μεταφοράς των δεδομένων είναι σημαντική για τη μακροχρόνια χρήση μιας εφαρμογής. Όταν μια υπηρεσία δεν προσφέρει εξαγωγή, ο χρήστης μπορεί να δυσκολευτεί να κρατήσει αντίγραφο ή να μεταφέρει το ιστορικό του αλλού.

Open source και γνωστές μορφές, όπως CSV, JSON και GPX, διευκολύνουν την ανταλλαγή δεδομένων. Το GPX μπορεί να χρησιμοποιηθεί από πολλές εφαρμογές και συσκευές πλοήγησης, ενώ το CSV είναι κατάλληλο για την εξαγωγή δομημένων καταχωρίσεων.

Το Moto-Vault υποστηρίζει εισαγωγή και εξαγωγή δεδομένων, δημιουργία αντιγράφου ασφαλείας του λογαριασμού και επαναφορά από συμβατό αρχείο. Υποστηρίζει επίσης εξαγωγές CSV για επιμέρους δεδομένα και δημιουργία αναφοράς συντήρησης σε PDF. Οι λειτουργίες αυτές επιτρέπουν στον χρήστη να διατηρεί αντίγραφο του ιστορικού του και να χρησιμοποιεί τα δεδομένα και εκτός της εφαρμογής.

### 2.2.7 Περιορισμοί της ψηφιακής καταγραφής

Η ψηφιακή καταγραφή εξαρτάται από τη διαθεσιμότητα της υπηρεσίας, τη σωστή λειτουργία της υποδομής και τη διατήρηση αντιγράφων ασφαλείας. Ένα τεχνικό πρόβλημα ή λανθασμένη ρύθμιση μπορεί να επηρεάσει την πρόσβαση στα δεδομένα.

Υπάρχει επίσης το ζήτημα της μακροχρόνιας διατήρησης. Μια διαδικτυακή υπηρεσία μπορεί στο μέλλον να αλλάξει ή να σταματήσει να λειτουργεί. Για αυτό είναι σημαντικό να παρέχεται δυνατότητα εξαγωγής και ο ιδιοκτήτης να συνεχίζει να διατηρεί τα επίσημα παραστατικά.

Η εφαρμογή δεν μπορεί να επιβεβαιώσει αυτόματα ότι μια εργασία πραγματοποιήθηκε σωστά ή ότι ένα ανταλλακτικό είναι κατάλληλο. Δεν αντικαθιστά τον μηχανικό, τις οδηγίες του κατασκευαστή ή τα επίσημα συστήματα. Ο ρόλος της είναι να οργανώνει και να παρουσιάζει τις πληροφορίες που εισάγει ο χρήστης.

Παρά τους περιορισμούς αυτούς, η ψηφιακή καταγραφή μπορεί να κάνει σημαντικά ευκολότερη τη διαχείριση του ιστορικού μιας μοτοσυκλέτας. Το βασικό της πλεονέκτημα είναι ότι συγκεντρώνει και συνδέει διαφορετικές κατηγορίες δεδομένων σε ένα κοινό περιβάλλον.

Στην επόμενη ενότητα παρουσιάζονται εφαρμογές και υπηρεσίες που καλύπτουν μέρος αυτών των αναγκών.

## 2.3 Υφιστάμενες εφαρμογές και υπηρεσίες

Για να υπάρξει εικόνα του ανταγωνισμού, εξετάστηκαν εφαρμογές και υπηρεσίες διαχείρισης οχημάτων, όπως εργαλεία συντήρησης, καταγραφής καυσίμων, εξόδων, διαδρομών, καθώς και μια πλατφόρμα συνεργείων.

Η ανάλυση βασίστηκε στις επίσημες περιγραφές των κατασκευαστών στα site τους και στο Google Play, χωρίς ανεξάρτητο τεχνικό έλεγχο. Έτσι, δεν βγήκαν αυθαίρετα συμπεράσματα για την ποιότητα, την ασφάλεια ή την αρχιτεκτονική τους.

### 2.3.1 Motorfolio

Το Motorfolio είναι ελληνική διαδικτυακή πλατφόρμα μηχανογράφησης για συνεργεία οχημάτων. Απευθύνεται κυρίως σε επαγγελματίες και όχι στον ιδιώτη κάτοχο μιας μοτοσυκλέτας.

Σύμφωνα με την επίσημη ιστοσελίδα, υποστηρίζει διαχείριση πελατών, οχημάτων, εργασιών, ραντεβού, ανταλλακτικών, υπενθυμίσεων service και ηλεκτρονικής τιμολόγησης. Περιλαμβάνει επίσης ψηφιακό βιβλίο service, μέσω του οποίου ο πελάτης μπορεί να βλέπει το ιστορικό εργασιών του οχήματός του και να ενημερώνεται για επόμενες προγραμματισμένες εργασίες. Η πλατφόρμα αναφέρει ακόμη αυτόματες ειδοποιήσεις μέσω email ή SMS.

### 2.3.2 Στρίφτεροι

Οι Στρίφτεροι είναι ελληνική εφαρμογή για αναβάτες μοτοσυκλέτας. Η επίσημη ιστοσελίδα παρουσιάζει λειτουργίες κοινότητας, ομαδικών διαδρομών και καταγραφής δραστηριότητας.

Η εφαρμογή διαθέτει και ενότητα γκαράζ. Στην ενότητα αυτή αναφέρονται η διαχείριση μοτοσυκλέτας, οι υπενθυμίσεις για service και αλλαγή λαδιών, η παρακολούθηση ασφάλισης και ΚΤΕΟ, το ιστορικό συντήρησης και οι ειδοποιήσεις λήξης εγγράφων.

### 2.3.3 MotoVault Motorcycle Garage

Η εφαρμογή MotoVault Motorcycle Garage δεν συνδέεται με το Moto-Vault της παρούσας διπλωματικής εργασίας. Πρόκειται για διαφορετικό εμπορικό προϊόν που χρησιμοποιεί παρόμοια ονομασία.

Η επίσημη ιστοσελίδα και η καταχώριση στο Google Play αναφέρουν παρακολούθηση εργασιών συντήρησης, ιστορικό service, υπενθυμίσεις βάσει χρόνου και χιλιομέτρων, καταγραφή εξόδων και καυσίμων, καθώς και σχεδιασμό ταξιδιών. Η εφαρμογή είναι διαθέσιμη για Android και iOS και δεν παρουσιάζεται ως πλήρης εφαρμογή διαχείρισης μέσω browser.

### 2.3.4 Rydful

Το Rydful παρουσιάζεται στο Google Play ως εφαρμογή παρακολούθησης διαδρομών, καυσίμων και συντήρησης.

Η καταχώρισή του αναφέρει αυτόματη ή χειροκίνητη καταγραφή διαδρομών με GPS, καταγραφή ανεφοδιασμών και φορτίσεων ηλεκτρικών οχημάτων, ιστορικό συντήρησης, υπενθυμίσεις βάσει ημερομηνίας ή χιλιομέτρων, στατιστικά διαδρομών και διαχείριση πολλών οχημάτων. Αναφέρεται επίσης αυτόματη έναρξη και διακοπή της καταγραφής

μέσω Bluetooth beacon, καθώς και ανάγνωση στοιχείων καυσίμου από φωτογραφία απόδειξης ή οθόνης αντλίας.

### 2.3.5 RideLog

Το RideLog είναι εφαρμογή κινητού για την καταγραφή διαδρομών και εξόδων μοτοσυκλέτας.

Σύμφωνα με την καταχώριση του Google Play, καταγράφει αυτόματα διαδρομές και αποθηκεύει απόσταση, πορεία, ταχύτητα και μεταβολές υψομέτρου. Υποστηρίζει ακόμη καταχώριση ανεφοδιασμών, υπολογισμό κατανάλωσης, έξοδα συντήρησης, ασφάλισης, επισκευών και μετατροπών, καθώς και υπενθυμίσεις συντήρησης.

### 2.3.6 Fuel Log

Το Fuel Log είναι εφαρμογή καταγραφής καυσίμων και εξόδων για αυτοκίνητα, φορτηγά και μοτοσυκλέτες.

Η καταχώριση στο Google Play αναφέρει καταγραφή ανεφοδιασμών, υπολογισμό κατανάλωσης και MPG, παρακολούθηση τιμών και κόστους καυσίμου, γενικά έξοδα οχήματος, ιστορικό συντήρησης και υπενθυμίσεις βάσει ημερομηνίας ή χιλιομέτρων. Περιλαμβάνει επίσης στατιστικά και γραφήματα.

### 2.3.7 Motohub

Το Motohub είναι εφαρμογή κοινότητας και διαδρομών για μοτοσυκλετιστές.

Η καταχώριση στο Google Play αναφέρει καταγραφή διαδρομών με απόσταση, χρόνο, ταχύτητα, υψόμετρο, γωνίες κλίσης και καιρικά στοιχεία. Υποστηρίζει κοινή χρήση και μεταφόρτωση διαδρομών σε μορφή GPX, δημιουργία τυχαίων διαδρομών, γκαράζ μοτοσυκλετών και λίστες ελέγχου service. Περιλαμβάνει ακόμη εκδηλώσεις, ομαδικές διαδρομές και παρακολούθηση φίλων στον χάρτη.

### 2.3.8 Drivvo

Το Drivvo είναι πλατφόρμα διαχείρισης οχήματος που είναι διαθέσιμη σε Android, iOS και Web.

Η επίσημη ιστοσελίδα αναφέρει παρακολούθηση καυσίμων, εξόδων, εσόδων, προληπτικής συντήρησης και συνολικού κόστους. Παρέχει υπενθυμίσεις, αναφορές, γραφήματα, ιστορικό οχήματος και συγχρονισμό μεταξύ συσκευών. Υποστηρίζει αυτοκίνητα, μοτοσυκλέτες και άλλους τύπους οχημάτων.

### 2.3.9 MotorManage

Το MotorManage είναι εφαρμογή Android και iOS που επικεντρώνεται στη συντήρηση μοτοσυκλέτας.
Η επίσημη ιστοσελίδα αναφέρει καταγραφή εργασιών service, ιστορικό συντήρησης, παρακολούθηση της κατάστασης εξαρτημάτων, υπενθυμίσεις βάσει χρόνου και χιλιομέτρων, διαχείριση περισσότερων μοτοσυκλετών και λειτουργία χωρίς σύνδεση στο διαδίκτυο. Κάθε μοτοσυκλέτα διαθέτει ξεχωριστό ιστορικό και πρόγραμμα συντήρησης.

Η παρούσα ανάλυση δεν περιλαμβάνει λειτουργίες κοινής χρήσης με μηχανικούς, μεταβίβασης ιστορικού ή επαγγελματικής συνεργασίας, επειδή δεν επιβεβαιώθηκαν ως βασικές λειτουργίες από τις πηγές που εξετάστηκαν.

### 2.3.10 MotoLogger

Το MotoLogger είναι εφαρμογή Android και iOS για τη διαχείριση συντήρησης και εγγράφων μοτοσυκλέτας.

Η επίσημη ιστοσελίδα αναφέρει ψηφιακό γκαράζ πολλών μοτοσυκλετών, ιστορικό service, υπενθυμίσεις, παρακολούθηση ασφάλισης και τεχνικού ελέγχου, αποθήκευση εγγράφων και εξαγωγή ιστορικού σε PDF. Αναφέρει επίσης εισαγωγή εγχειριδίου χρήσης, εξαγωγή του προγράμματος συντήρησης από αυτό και βοηθό τεχνητής νοημοσύνης που μπορεί να απαντά με βάση τα στοιχεία της μοτοσυκλέτας και το εγχειρίδιο.

Άλλη επίσημη σελίδα της εφαρμογής αναφέρει καταγραφή διαδρομών και οργάνωση φωτογραφιών ανά μοτοσυκλέτα. Οι λειτουργίες αυτές καταγράφονται ως δηλώσεις του δημιουργού και δεν αξιολογήθηκαν για το σκοπό της εργασίας.

### 2.3.11 Ομοιότητες στη σχεδίαση και η επίδραση των εργαλείων AI

Η μελέτη των εφαρμογών έδειξε ότι οι περισσότερες είναι ολόιδιες, τόσο σε δυνατότητες όσο και σε σχεδιασμό. Στις περισσότερες περιπτώσεις, το όχημα ή το ψηφιακό γκαράζ αποτελεί το κεντρικό σημείο του UI, γύρω από το οποίο οργανώνονται η συντήρηση, τα έξοδα, οι υπενθυμίσεις και οι διαδρομές. Παρόμοια μοτίβα εμφανίζονται στη χρήση καρτών, μεγάλων αριθμητικών ενδείξεων, στρογγυλεμένων πλαισίων, γραμμικών εικονιδίων και dashboards με παρόμοια δομή.

Η ομοιομορφία αυτή είναι ιδιαίτερα εμφανής στις νεότερες εφαρμογές της κατηγορίας. Αρκετές δίνουν την εντύπωση ότι έχουν δημιουργηθεί με κοινές βιβλιοθήκες, έτοιμα πρότυπα UI ή διαδικασίες ανάπτυξης που βασίζονται σε μεγάλο βαθμό σε εργαλεία τεχνητής νοημοσύνης. Στη σύγχρονη ορολογία, εφαρμογές που παράγονται γρήγορα μέσα από διαδοχικές οδηγίες προς AI coding agents χαρακτηρίζονται συχνά ως "vibe coded". Ο όρος δεν αποτελεί αυστηρή τεχνική κατηγορία, αλλά περιγράφει έναν τρόπο ανάπτυξης στον οποίο μεγάλο μέρος του κώδικα και της διεπαφής δημιουργείται με τη βοήθεια γενετικών εργαλείων.

Αρκετές εφαρμογές επαναλαμβάνουν ολόιδια γραφικά στοιχεία και λειτουργίες, παρόμοιες ροές χρήσης και σχεδόν ίδια προωθητική ορολογία. Η τυφλή χρήση του AI δείχνει ότι η ευρεία χρήση κοινών εργαλείων και προτύπων μπορεί να οδηγήσει σε προϊόντα που μοιάζουν αρκετά μεταξύ τους και δεν προσφέρουν κάτι αξιόλογο και μοναδικό.

Χαρακτηριστικό παράδειγμα αποτελεί το Impeccable, το οποίο παρέχει έτοιμες οδηγίες σχεδίασης UI/UX για AI coding agents. Τέτοια εργαλεία μπορούν να επιταχύνουν σημαντικά την ανάπτυξη και να παράγουν καλαίσθητες διεπαφές, αλλά μπορούν επίσης να ενισχύσουν την ομοιομορφία όταν χρησιμοποιούνται οι ίδιες προτάσεις, βιβλιοθήκες και σχεδιαστικές συμβάσεις.

Η αναφορά αυτή αποτελεί παρατήρηση για τη γενικότερη τάση της αγοράς και για τον τρόπο με τον οποίο τα εργαλεία AI μπορούν να επηρεάσουν την εμφάνιση και την οργάνωση σύγχρονων εφαρμογών.

Στην περίπτωση του Moto-Vault, οι λειτουργίες, το μοντέλο δεδομένων, η αρχιτεκτονική και οι βασικές ροές χρήσης σχεδιάστηκαν με βάση τις απαιτήσεις της παρούσας εργασίας. Εργαλεία τεχνητής νοημοσύνης χρησιμοποιήθηκαν υποστηρικτικά κατά την ανάπτυξη, χωρίς να αποτελούν υποκατάστατο του σχεδιασμού, της τεχνικής αξιολόγησης και του ελέγχου του τελικού αποτελέσματος. Η διαφορά της εργασίας με τον υπάρχοντα

ανταγωνισμό δεν βασίζεται μόνο στην εμφάνιση του UI, αλλά στη συνολική διαδικασία ανάλυσης, σχεδιασμού, υλοποίησης και αξιολόγησης του συστήματος.

## 2.4 Συγκριτική ανάλυση υφιστάμενων λύσεων

Η προηγούμενη ενότητα δείχνει ότι οι εφαρμογές ψηφιακής διαχείρισης μοτοσυκλέτας δεν ακολουθούν όλες το ίδιο μοτίβο. Άλλες επικεντρώνονται στη συντήρηση, άλλες στα καύσιμα και στα έξοδα, ενώ αρκετές δίνουν μεγαλύτερη έμφαση στις διαδρομές ή στη social media πλευρά.

Η σύγκριση βασίζεται στις επίσημες ιστοσελίδες και στις καταχωρίσεις του Google Play που εξετάστηκαν στην Ενότητα 2.3. Οι λειτουργίες αποτυπώνονται με τις ακόλουθες ενδείξεις:

- **Ναι**, όταν η λειτουργία αναφέρεται ρητά στην επίσημη πηγή.
- **Μερικώς**, όταν προσφέρεται περιορισμένη ή παρόμοια λειτουργία.
- **Δεν επιβεβαιώθηκε**, όταν δεν εντοπίστηκε σαφής αναφορά στις πηγές που εξετάστηκαν.

### 2.4.1 Κριτήρια σύγκρισης

Για τη σύγκριση επιλέχθηκαν τα ακόλουθα κριτήρια:

1. **Βασικό κοινό**, δηλαδή αν η λύση απευθύνεται σε ιδιώτες, αναβάτες, συνεργεία ή γενικά σε κατόχους οχημάτων.
2. **Πρόσβαση μέσω Web**, δηλαδή αν ο χρήστης μπορεί να διαχειρίζεται τα δεδομένα του από browser.
3. **Ιστορικό συντήρησης**, δηλαδή η δυνατότητα καταγραφής εργασιών service και επισκευών.
4. **Υπενθυμίσεις**, με βάση χρόνο, ημερομηνία ή χιλιομετρική ένδειξη.
5. **Καύσιμα και έξοδα**, δηλαδή η παρακολούθηση ανεφοδιασμών και οικονομικών στοιχείων.
6. **Φωτογραφίες και έγγραφα**, όταν συνδέονται με το όχημα ή με καταχωρίσεις του.
7. **Διαδρομές και GPX**, είτε μέσω αυτόματης καταγραφής είτε μέσω εισαγωγής και εξαγωγής αρχείων.
8. **Πολλαπλά οχήματα**, δηλαδή η διαχείριση περισσότερων από μίας μοτοσυκλετών ή οχημάτων.
9. **Κοινωνικές λειτουργίες**, όπως κοινότητα, ομαδικές βόλτες, διαμοιρασμός και ζωντανή παρακολούθηση.
10. **Εστίαση στη μοτοσυκλέτα**, δηλαδή αν η λύση σχεδιάστηκε ειδικά για αναβάτες και μοτοσυκλέτες.

### 2.4.2 Πίνακας σύγκρισης

| Λύση | Βασικό κοινό | Web | Garage | Media & GPX | Social & Multi-veh. | Moto Specific |
|---|---|---|---|---|---|---|
| **Motorfolio** | Συνεργεία | ✓ | ✓ | ✓ / ? | ✗ / ✓ | ✗ |
| **Στρίφτεροι** | Αναβάτες | ✗ | ~ | ? / ✓ | ✓ / ? | ✓ |
| **MotoVault Motorcycle Garage** | Ιδιοκτήτες Moto | ✗ | ✓ | ~ / ✓ | ? / ✓ | ✓ |
| **Rydful** | Ιδιοκτήτες οχημάτων | ✗ | ✓ | ~ / ✓ | ? / ✓ | ✗ |
| **RideLog** | Αναβάτες | ✗ | ✓ | ? / ✓ | ~ / ✓ | ✓ |
| **Fuel Log** | Ιδιοκτήτες οχημάτων | ✗ | ✓ | ? / ? | ✗ / ? | ✗ |
| **Motohub** | Αναβάτες | ✗ | ~ | ? / ✓ | ✓ / ✓ | ✓ |
| **Drivvo** | Ιδιώτες / Στόλοι | ✓ | ✓ | ? / ~ | ✗ / ✓ | ✗ |
| **MotorManage** | Ιδιοκτήτες Moto | ✗ | ✓ | ✓ / ? | ? / ✓ | ✓ |
| **MotoLogger** | Ιδιοκτήτες Moto | ✗ | ✓ | ✓ / ✓ | ? / ✓ | ✓ |
| **Moto-Vault** | Ιδιοκτήτες Moto | ✓ | ✓ | ✓ / ✓ | ✗ / ✓ | ✓ |

**Πίνακας 2.1**: Συγκριτική αποτύπωση των εφαρμογών και υπηρεσιών που εξετάστηκαν.

Ο πίνακας δεν αξιολογεί την ποιότητα κάθε λειτουργίας. Δύο εφαρμογές μπορεί για παράδειγμα να υποστηρίζουν ιστορικό συντήρησης, αλλά να διαφέρουν σημαντικά ως προς τα πεδία, την ευχρηστία και τη σύνδεση με άλλα δεδομένα. Ο πίνακας δείχνει μόνο αν μια κατηγορία λειτουργιών αναφέρεται στις επίσημες περιγραφές.

### 2.4.3 Κοινό, σκοπός και τρόπος πρόσβασης

Το Motorfolio είναι επαγγελματικό σύστημα για συνεργεία. Η καταγραφή του ιστορικού συντήρησης συνδέεται με τη διαχείριση πελατών, εργασιών, ραντεβού και τιμολόγησης. Ο πελάτης μπορεί να βλέπει το ψηφιακό βιβλίο service, όμως τον βασικό έλεγχο της πληροφορίας έχει το συνεργείο.

Το Drivvo, το Rydful και το Fuel Log απευθύνονται γενικότερα σε ιδιοκτήτες οχημάτων. Υποστηρίζουν μοτοσυκλέτες, αλλά δεν έχουν σχεδιαστεί αποκλειστικά για αναβάτες. Αντίθετα, οι Στρίφτεροι, το RideLog και το Motohub επικεντρώνονται περισσότερο στην εμπειρία της οδήγησης, στις διαδρομές και στην κοινότητα.

Οι πιο άμεσα συγκρίσιμες εφαρμογές ως προς την εστίαση στη μοτοσυκλέτα είναι το MotoVault Motorcycle Garage, το MotorManage και το MotoLogger. Το MotorManage δίνει έμφαση στη συντήρηση και στην κατάσταση των εξαρτημάτων. Το MotoLogger συνδυάζει έγγραφα, εγχειρίδιο χρήσης και βοηθό τεχνητής νοημοσύνης. Το MotoVault Motorcycle Garage συνδυάζει συντήρηση, έξοδα και διαδρομές.

Η πλειονότητα των λύσεων είναι σχεδιασμένη κυρίως για Android και iOS, για λειτουργίες όπως το GPS, η αυτόματη καταγραφή διαδρομών και η χρήση κατά την οδήγηση. Οι

επιβεβαιωμένες λύσεις που προσφέρουν πλήρη πρόσβαση μέσω browser είναι λιγότερες. Το Drivvo διαθέτει Web έκδοση για προσωπική χρήση, ενώ το Motorfolio λειτουργεί επίσης ως web πλατφόρμα αλλά απευθύνεται σε συνεργεία.

Το Moto-Vault της εργασίας λειτουργεί μέσω browser και μπορεί να χρησιμοποιηθεί καλά από υπολογιστή, tablet ή κινητό χωρίς εγκατάσταση από κατάστημα εφαρμογών.

### 2.4.4 Συντήρηση, καύσιμα και οικονομική παρακολούθηση

Η συντήρηση εμφανίζεται στις περισσότερες εφαρμογές, αλλά όχι με το ίδιο επίπεδο λεπτομέρειας. Σε ορισμένες λύσεις περιορίζεται σε λίστες ελέγχου ή υπενθυμίσεις, ενώ σε άλλες αποθηκεύεται πλήρες ιστορικό με ημερομηνία, χιλιόμετρα, κόστος, εργασίες και ανταλλακτικά.

Το MotorManage επικεντρώνεται στην κατάσταση των εξαρτημάτων και στα προγράμματα αντικατάστασης. Το Motorfolio συνδέει τις εργασίες με τις επαγγελματικές διαδικασίες του συνεργείου.

Στο Moto-Vault κάθε εγγραφή συντήρησης συνδέεται με συγκεκριμένη μοτοσυκλέτα και μπορεί να περιλαμβάνει ημερομηνία, χιλιομετρική ένδειξη, περιγραφή, κόστος, φωτογραφίες και σχετικά εξαρτήματα. Οι υπενθυμίσεις αποθηκεύονται ξεχωριστά και μπορούν να βασίζονται σε ημερομηνία ή χιλιόμετρα.

Στην καταγραφή καυσίμων και εξόδων δίνουν ιδιαίτερη έμφαση τα Fuel Log, Drivvo, Rydful, RideLog και MotoVault Motorcycle Garage. Το Fuel Log είναι η πιο εξειδικευμένη λύση ως προς τους ανεφοδιασμούς και την κατανάλωση, ενώ το Drivvo προσφέρει ευρύτερη οικονομική εικόνα, συνδυάζοντας καύσιμα, συντήρηση, ασφάλιση, φόρους και άλλα έξοδα. Τα Rydful και RideLog συνδέουν τους ανεφοδιασμούς με τη χρήση και τις καταγεγραμμένες διαδρομές.

Το Moto-Vault καταγράφει ανεφοδιασμούς με ημερομηνία, χιλιόμετρα, ποσότητα, τιμή και συνολικό κόστος. Το κόστος αποθηκεύεται επίσης στις εργασίες συντήρησης, στα ανταλλακτικά και στα αξεσουάρ, ώστε η οικονομική εικόνα να μην περιορίζεται μόνο στα καύσιμα.

### 2.4.5 Διαδρομές, κοινωνικές λειτουργίες και αρχεία GPX

Οι διαδρομές αποτελούν βασική λειτουργία στα Rydful, RideLog, Motohub, Στρίφτεροι και MotoVault Motorcycle Garage. Οι εφαρμογές αυτές χρησιμοποιούν το GPS του κινητού για αυτόματη ή χειροκίνητη καταγραφή και μπορούν να υπολογίζουν στοιχεία όπως απόσταση, ταχύτητα και υψόμετρο.

Το Motohub και οι Στρίφτεροι προσθέτουν κοινωνικές λειτουργίες, όπως ομαδικές διαδρομές, κοινότητα και παρακολούθηση άλλων αναβατών. Το MotoVault Motorcycle Garage περιλαμβάνει σχεδιασμό ταξιδιών και εξαγωγή GPX.

Το Moto-Vault της εργασίας ακολουθεί διαφορετική προσέγγιση. Δεν καταγράφει ζωντανά τη θέση του χρήστη προς το παρόν, εφόσον δεν υπάρχει υλοποίηση για κινητά. Επιτρέπει μόνο την εισαγωγή υπάρχοντος αρχείου GPX, την αποθήκευση των βασικών πληροφοριών της διαδρομής και την προβολή της σε χάρτη.

### 2.4.6 Φωτογραφίες και έγγραφα

Η υποστήριξη φωτογραφιών και εγγράφων δεν εμφανίζεται με την ίδια σαφήνεια σε όλες τις εφαρμογές. Το MotoLogger προσφέρει χώρο για έγγραφα και φωτογραφίες. Το Motorfolio επιτρέπει τη σύνδεση αρχείων με πελάτες, οχήματα και εργασίες, ενώ το MotorManage αναφέρει αποδείξεις και πληροφορίες ανταλλακτικών.

Στο Moto-Vault οι φωτογραφίες μπορούν να αποθηκεύονται ως γενικό φωτογραφικό υλικό της μοτοσυκλέτας, αλλά και να συνδέονται με εργασίες συντήρησης, αξεσουάρ και ανταλλακτικά. Τα έγγραφα διατηρούνται σε ξεχωριστή ενότητα και μπορούν να συνοδεύονται από ημερομηνίες λήξης. Άρα, δεν γίνεται απλή συλλογή αρχείων, αλλά συνδέεται με το ιστορικό και τις επιμέρους πληροφορίες του οχήματος.

### 2.4.7 Θέση και διαφοροποίηση του Moto-Vault

Η ανάλυση δείχνει ότι υπάρχουν ήδη αρκετές εφαρμογές διαχείρισης μοτοσυκλέτας με σημαντικό εύρος λειτουργιών. Ορισμένες εμπορικές λύσεις προσφέρουν δυνατότητες που δεν περιλαμβάνονται στο σύστημα της εργασίας, όπως αυτόματη καταγραφή GPS, ζωντανή πλοήγηση, κοινότητα, βοηθούς τεχνητής νοημοσύνης και προκαθορισμένη βάση μηχανών.

Η διαφοροποίηση του Moto-Vault βρίσκεται κυρίως στον συνδυασμό των ακόλουθων χαρακτηριστικών:

1. Είναι διαδικτυακή πλατφόρμα που λειτουργεί από browser.
2. Έχει σχεδιαστεί ειδικά για προσωπική διαχείριση μοτοσυκλέτας.
3. Συνδυάζει συντήρηση, καύσιμα, έξοδα, ανταλλακτικά, αξεσουάρ, έγγραφα, φωτογραφίες, υπενθυμίσεις και GPX.
4. Διατηρεί ενιαίο μοντέλο δεδομένων γύρω από τον χρήστη και τη μοτοσυκλέτα.
5. Παρουσιάζει συγκεντρωτικά στοιχεία μέσω dashboard.
6. Η αρχιτεκτονική, οι απαιτήσεις και η διαδικασία ανάπτυξης τεκμηριώνονται στο σκοπό της διπλωματικής εργασίας.
7. Η χρηστικότητα του τελικού συστήματος αξιολογείται με το ερωτηματολόγιο SUS.

Η συνεισφορά της εργασίας δεν παρουσιάζεται ως δημιουργία μιας εντελώς νέας κατηγορίας λογισμικού. Πρόκειται για τον σχεδιασμό και την υλοποίηση μιας ανεξάρτητης web λύσης, η οποία συγκεντρώνει λειτουργίες που συχνά προσφέρονται χωριστά ή κυρίως μέσα από εφαρμογές κινητών συσκευών.

## 2.5 Συμπεράσματα

Η ανάλυση έδειξε ότι η ψηφιακή διαχείριση πληροφοριών οχήματος είναι ήδη ένα αναπτυγμένο πεδίο. Υπάρχουν εφαρμογές που καλύπτουν συντήρηση, υπενθυμίσεις, καύσιμα, έξοδα, διαδρομές, έγγραφα και διαχείριση πολλών οχημάτων.

Οι λύσεις αυτές δεν έχουν όλες τον ίδιο σκοπό. Ορισμένες επικεντρώνονται στην οικονομική παρακολούθηση, άλλες στη συντήρηση, ενώ αρκετές δίνουν μεγαλύτερη έμφαση στις διαδρομές, στην κοινότητα ή στη χρήση από κινητή συσκευή. Υπάρχουν επίσης επαγγελματικές πλατφόρμες για συνεργεία, οι οποίες οργανώνουν τις πληροφορίες από την πλευρά της επιχείρησης και όχι του ιδιοκτήτη.

Η σύγκριση έδειξε ακόμη ότι οι περισσότερες εφαρμογές που απευθύνονται ειδικά σε μοτοσυκλετιστές είναι σχεδιασμένες κυρίως για Android και iOS. Η προσέγγιση αυτή είναι κατάλληλη για λειτουργίες GPS και καταγραφή κατά την οδήγηση. Οι ολοκληρωμένες

λύσεις που λειτουργούν μέσω browser και απευθύνονται σε ιδιώτες κατόχους μοτοσυκλετών είναι λιγότερες.

Το κενό δεν βρίσκεται επομένως στην πλήρη απουσία σχετικών εφαρμογών. Βρίσκεται περισσότερο στον συνδυασμό συγκεκριμένων λειτουργιών μέσα σε ένα ενιαίο web περιβάλλον. Το Moto-Vault σχεδιάστηκε ώστε να συγκεντρώνει:

1. ιστορικό συντήρησης και υπενθυμίσεις,
2. ανεφοδιασμούς και οικονομικά στοιχεία,
3. ανταλλακτικά και αξεσουάρ,
4. φωτογραφίες και έγγραφα,
5. διαδρομές GPX,
6. διαχείριση περισσότερων από μίας μοτοσυκλετών,
7. συνοπτική παρουσίαση πληροφοριών μέσω dashboard.

Η πλατφόρμα δεν επιχειρεί να αντικαταστήσει εφαρμογές πλοήγησης, κοινωνικά δίκτυα αναβατών ή επαγγελματικά συστήματα συνεργείων. Επικεντρώνεται στην προσωπική οργάνωση των πληροφοριών της μοτοσυκλέτας και στην πρόσβαση σε αυτές από browser.

Η εργασία δεν δημιουργεί μια εντελώς νέα κατηγορία λογισμικού. Βασίζεται στον σχεδιασμό, στην υλοποίηση και στην αξιολόγηση μιας ανεξάρτητης διαδικτυακής λύσης που συνδυάζει λειτουργίες οι οποίες συχνά προσφέρονται χωριστά ή κυρίως μέσα από εφαρμογές κινητών συσκευών.

Οι απαιτήσεις, η αρχιτεκτονική, το μοντέλο δεδομένων, οι σχεδιαστικές επιλογές και η αξιολόγηση της χρηστικότητας τεκμηριώνονται στα επόμενα κεφάλαια.

Τα συμπεράσματα της παρούσας ανάλυσης χρησιμοποιήθηκαν για τον καθορισμό των λειτουργικών και μη λειτουργικών απαιτήσεων του Moto-Vault και για την αιτιολόγηση των βασικών επιλογών σχεδιασμού.

# 3. Μεθοδολογία

## 3.1 Design Science Research

Αυτή η διπλωματική εργασία έχει καθαρά σχεδιαστικό και υλοποιητικό χαρακτήρα, καθώς βασικό της αποτέλεσμα είναι η ανάπτυξη ενός λειτουργικού πληροφοριακού συστήματος. Έτσι επιλέχθηκε η Design Science Research, η οποία χρησιμοποιείται στην έρευνα πληροφοριακών συστημάτων όταν ο στόχος είναι η δημιουργία και η αξιολόγηση ενός τεχνολογικού προϊόντος (software artifact).

Ο όρος "ψηφιακό προϊόν" αναφέρεται σε ένα σύστημα που σχεδιάζεται για να λύσει ένα πραγματικό πρόβλημα. Στην πληροφορική, αυτό μπορεί να είναι ένα μοντέλο, μια μέθοδος, μια αρχιτεκτονική ή μια εφαρμογή. Στη δική μας περίπτωση, το προϊόν αυτό είναι η διαδικτυακή πλατφόρμα MotoVault.

Η μεθοδολογία Design Science Research δεν μένει μόνο στο γράψιμο κώδικα. Συνδέει το πρόβλημα, τις απαιτήσεις, τις σχεδιαστικές επιλογές, την υλοποίηση και την αξιολόγηση. Έτσι, το τελικό σύστημα δεν αντιμετωπίζεται απλά ως ένα "εργαλείο", αλλά ως μια ολοκληρωμένη λύση σε μια συγκεκριμένη ανάγκη.

Σύμφωνα με τους Hevner et al. (2004), η έρευνα μέσω σχεδιασμού στοχεύει στη δημιουργία και αξιολόγηση συστημάτων που λύνουν ουσιαστικά τεχνολογικά και οργανωτικά προβλήματα. Η γνώση δηλαδή προκύπτει μέσα από την ίδια τη διαδικασία σχεδιασμού, υλοποίησης και αξιολόγησης στην πράξη.

Η προσέγγιση αυτή είναι κατάλληλη για το Moto-Vault, επειδή η εργασία ξεκινά από ένα πρακτικό πρόβλημα, τη διάσπαρτη διαχείριση πληροφοριών μοτοσυκλέτας, και προχωρά στον σχεδιασμό και στην ανάπτυξη μιας συγκεκριμένης λύσης. Η χρησιμότητα της λύσης εξετάζεται στη συνέχεια μέσα από τεχνικούς ελέγχους και αξιολόγηση χρηστικότητας.

Στην επόμενη ενότητα παρουσιάζεται η Design Science Research Methodology, η οποία οργανώνει τη διαδικασία σε έξι βασικά στάδια.

## 3.2 Design Science Research Methodology

Η Design Science Research Methodology, γνωστή ως DSRM, είναι μια οργανωμένη διαδικασία εφαρμογής της έρευνας μέσω σχεδιασμού στα πληροφοριακά συστήματα. Προτάθηκε από τους Peffers κ.ά. (2007) και περιγράφει τα βασικά στάδια από την αναγνώριση ενός προβλήματος μέχρι την παρουσίαση των αποτελεσμάτων.

Η μεθοδολογία περιλαμβάνει έξι στάδια:

1. αναγνώριση του προβλήματος και αιτιολόγηση της σημασίας του
2. καθορισμό των στόχων της λύσης
3. σχεδιασμό και ανάπτυξη του συστήματος
4. επίδειξη της χρήσης του
5. αξιολόγηση του αποτελέσματος
6. επικοινωνία της έρευνας και των αποτελεσμάτων

Στο πρώτο στάδιο προσδιορίζεται το πρόβλημα, το περιβάλλον στο οποίο εμφανίζεται και οι χρήστες που επηρεάζονται. Στο δεύτερο στάδιο η ανάγκη μετατρέπεται σε

συγκεκριμένους στόχους, οι οποίοι αργότερα μπορούν να χρησιμοποιηθούν για την αξιολόγηση της λύσης.

Το τρίτο στάδιο περιλαμβάνει τον σχεδιασμό και την κατασκευή του συστήματος/προϊόντος. Σε μια εφαρμογή λογισμικού αυτό σημαίνει καθορισμό της αρχιτεκτονικής, του μοντέλου δεδομένων, των βασικών ροών και των τεχνολογιών, καθώς και την υλοποίηση του συστήματος.

Στο τέταρτο στάδιο παρουσιάζεται πώς το σύστημα χρησιμοποιείται για την αντιμετώπιση του προβλήματος. Η επίδειξη μπορεί να γίνει με πραγματική χρήση ή παραδείγματα.

Στο πέμπτο στάδιο εξετάζεται κατά πόσο το αποτέλεσμα ανταποκρίνεται στους στόχους που είχαν τεθεί.

Το τελευταίο στάδιο αφορά την επικοινωνία του προβλήματος, της διαδικασίας, του συστήματος και των αποτελεσμάτων. Στην παρούσα εργασία η επικοινωνία πραγματοποιείται μέσω του κειμένου της διπλωματικής, των διαγραμμάτων, των στιγμιότυπων οθόνης και της τελικής παρουσίασης.

Τα στάδια δεν χρειάζεται να ακολουθούνται πάντα με αυστηρά γραμμικό τρόπο. Κατά την ανάπτυξη μπορεί να εμφανιστούν νέα προβλήματα ή ανάγκη αλλαγής κάποιας σχεδιαστικής απόφασης. Η διαδικασία μπορεί επομένως να επιστρέφει σε προηγούμενα στάδια και να λειτουργεί επαναληπτικά.

Στην περίπτωση του Moto-Vault, η διαδικασία ξεκίνησε από το πρόβλημα της χύμα διαχείρισης πληροφοριών μοτοσυκλέτας. Ο τρόπος με τον οποίο εφαρμόστηκαν τα έξι στάδια παρουσιάζεται αναλυτικά στην Ενότητα 3.5.

## 3.3 Ερευνητικά ερωτήματα

Η παρούσα διπλωματική εργασία έχει κυρίως σχεδιαστικό και υλοποιητικό χαρακτήρα. Τα ερευνητικά ερωτήματα δεν χρησιμοποιούνται για τον στατιστικό έλεγχο μιας θεωρητικής υπόθεσης, αλλά για την οργάνωση της διαδικασίας σχεδιασμού, ανάπτυξης και αξιολόγησης του συστήματος.

Τα ερωτήματα προκύπτουν από το πρόβλημα που παρουσιάστηκε στο Κεφάλαιο 1 και από τη σύγκριση των υφιστάμενων λύσεων στο Κεφάλαιο 2.

### 3.3.1 Κύριο ερευνητικό ερώτημα

**Πώς μπορεί να σχεδιαστεί, να αναπτυχθεί και να αξιολογηθεί μια διαδικτυακή πλατφόρμα που συγκεντρώνει και οργανώνει τις βασικές πληροφορίες μιας μοτοσυκλέτας σε ένα περιβάλλον;**

Το ερώτημα αυτό καλύπτει ολόκληρη τη διαδικασία της εργασίας, από την ανάλυση των αναγκών μέχρι την αξιολόγηση του τελικού συστήματος.

### 3.3.2 Πρώτο επιμέρους ερευνητικό ερώτημα

**Ποιες λειτουργικές και μη λειτουργικές απαιτήσεις είναι απαραίτητες για την προσωπική ψηφιακή διαχείριση πληροφοριών μοτοσυκλέτας;**

Το ερώτημα αφορά τον καθορισμό του περιεχομένου της λύσης. Η απάντησή του βασίζεται στην ανάλυση του προβλήματος, στη μελέτη των υφιστάμενων εφαρμογών και στη μετατροπή των αναγκών σε συγκεκριμένες απαιτήσεις. Παρουσιάζεται κυρίως στο Κεφάλαιο 4.

### 3.3.3 Δεύτερο επιμέρους ερευνητικό ερώτημα

**Ποια αρχιτεκτονική και ποιες τεχνικές επιλογές μπορούν να υποστηρίξουν με ασφαλή και συντηρήσιμο τρόπο τις απαιτήσεις της πλατφόρμας;**

Το ερώτημα αφορά τον σχεδιασμό και την τεχνική υλοποίηση του συστήματος. Η απάντηση αναπτύσσεται στα Κεφάλαια 4, 5 και 6.

### 3.3.4 Τρίτο επιμέρους ερευνητικό ερώτημα

**Σε ποιο βαθμό μπορεί το Moto-Vault να συνδυάσει σε μία web πλατφόρμα λειτουργίες που στις υφιστάμενες λύσεις εμφανίζονται συχνά κατανεμημένες σε διαφορετικές εφαρμογές;**

Το ερώτημα συνδέει τη σύγκριση του Κεφαλαίου 2 με το τελικό προϊόν. Δεν θεωρεί ότι απουσιάζουν άλλες ολοκληρωμένες λύσεις, αλλά εξετάζει αν το Moto-Vault υλοποιεί τον συγκεκριμένο συνδυασμό λειτουργιών που απουσιάζουν.

Η απάντηση προκύπτει από την υλοποίηση των Κεφαλαίων 5 και 6 και από την παρουσίαση του τελικού συστήματος στο Κεφάλαιο 7.

### 3.3.5 Τέταρτο επιμέρους ερευνητικό ερώτημα

**Πώς αντιλαμβάνονται οι χρήστες τη χρηστικότητα του Moto-Vault μετά τη δοκιμή και την ελεύθερη εξερεύνηση της εφαρμογής;**

Το ερώτημα αφορά την αξιολόγηση της αντιλαμβανόμενης χρηστικότητας με το System Usability Scale. Οι πρόσθετες ερωτήσεις εξετάζονται ξεχωριστά και δεν επηρεάζουν την επίσημη βαθμολογία SUS. Η απάντηση παρουσιάζεται στο Κεφάλαιο 8.

### 3.3.6 Πέμπτο επιμέρους ερευνητικό ερώτημα

**Ποιοι περιορισμοί και ποιες δυνατότητες βελτίωσης προκύπτουν από την ανάπτυξη και την αξιολόγηση του τεχνουργήματος;**

Το ερώτημα αφορά τη συνολική αποτίμηση της εργασίας. Περιλαμβάνει τεχνικούς περιορισμούς, περιορισμούς της αξιολόγησης και ανάγκες που δεν καλύφθηκαν στην παρούσα έκδοση. Η απάντηση παρουσιάζεται κυρίως στο Κεφάλαιο 9.

### 3.3.7 Σύνδεση με τη DSRM

**Πίνακας 3.1**: Σύνδεση των ερευνητικών ερωτημάτων με τα στάδια της DSRM

| **Ερευνητικό ερώτημα** | **Στάδια DSRM** |
|---|---|
| Κύριο ερευνητικό ερώτημα | Όλα τα στάδια |
| Απαιτήσεις της πλατφόρμας | Αναγνώριση προβλήματος και καθορισμός στόχων |
| Αρχιτεκτονική και τεχνικές επιλογές | Σχεδιασμός και ανάπτυξη |
| Ενοποίηση λειτουργιών | Σχεδιασμός, ανάπτυξη και επίδειξη |
| Αντιλαμβανόμενη χρηστικότητα | Αξιολόγηση |
| Περιορισμοί και βελτιώσεις | Αξιολόγηση και επικοινωνία |

Τα ερωτήματα απαντώνται συγκεντρωτικά στο τελευταίο κεφάλαιο, με βάση τα αποτελέσματα της ανάλυσης, της υλοποίησης και της αξιολόγησης.

## 3.4 Μεθοδολογικοί στόχοι

Αυτοί καθορίζουν τον τρόπο με τον οποίο οργανώνεται η έρευνα και η ανάπτυξη του Moto-Vault, δεν περιγράφουν μόνο τι πρέπει να κάνει η εφαρμογή αλλά και πώς πρέπει να τεκμηριωθεί και να αξιολογηθεί η διαδικασία δημιουργίας της.

### 3.4.1 Αναγνώριση και τεκμηρίωση του προβλήματος

Ο πρώτος στόχος είναι η σαφής περιγραφή του προβλήματος της διαχείρισης διάσπαρτων πληροφοριών μηχανης. Η ανάλυση περιλαμβάνει τις ανάγκες του ιδιοκτήτη, τις κατηγορίες δεδομένων που δημιουργούνται και τους περιορισμούς των υφιστάμενων λύσεων.

### 3.4.2 Μετατροπή των αναγκών σε απαιτήσεις

Ο δεύτερος στόχος είναι η μετατροπή των αναγκών σε λειτουργικές και μη λειτουργικές απαιτήσεις. Οι απαιτήσεις χρησιμοποιούνται ως βάση για τον σχεδιασμό, την υλοποίηση και την επαλήθευση του συστήματος.

### 3.4.3 Τεκμηριωμένος και επαναληπτικός σχεδιασμός

Ο τρίτος στόχος είναι η σύνδεση των σχεδιαστικών και τεχνικών επιλογών με συγκεκριμένες απαιτήσεις. Εδώ οι αποφάσεις να μπορούν να αναθεωρούνται όταν εμφανίζονται νέα προβλήματα ή περιορισμοί.

### 3.4.4 Τεχνική επαλήθευση και αξιολόγηση χρηστικότητας

Ο τέταρτος στόχος είναι η αξιολόγηση του συστήματος σε δύο επίπεδα. Η τεχνική επαλήθευση εξετάζει την ορθή λειτουργία βασικών τμημάτων του μέσω αυτοματοποιημένων και χειροκίνητων ελέγχων. Η αξιολόγηση χρηστικότητας εξετάζει την εμπειρία των χρηστών με το System Usability Scale.

### 3.4.5 Καταγραφή αποτελεσμάτων, περιορισμών και βελτιώσεων

Ο πέμπτος στόχος είναι η καταγραφή των αποτελεσμάτων χωρίς απόκρυψη των περιορισμών της εργασίας. Οι παρατηρήσεις από την ανάπτυξη και την αξιολόγηση χρησιμοποιούνται για την αποτίμηση του τελικού συστήματος και για τη διατύπωση μελλοντικών επεκτάσεων.

Οι στόχοι αυτοί συνδέουν τα ερευνητικά ερωτήματα με τις δραστηριότητες της εργασίας και με τον τρόπο αξιολόγησης του συστήματος.

## 3.5 Εφαρμογή της DSRM στην παρούσα εργασία

Η παρούσα εργασία ακολουθεί τα έξι στάδια της Design Science Research Methodology, όπως προτείνονται από τους Peffers κ.ά. (2007). Η εφαρμογή τους δεν ήταν αυστηρά γραμμική. Κατά την ανάπτυξη προέκυψαν νέες ανάγκες και τεχνικοί περιορισμοί, με αποτέλεσμα ορισμένες αποφάσεις να αναθεωρηθούν.

### 3.5.1 Αναγνώριση του προβλήματος

Η αρχική ανάγκη προέκυψε από την προσωπική εμπειρία διαχείρισης πληροφοριών μιας μοτοσυκλέτας. Στοιχεία συντήρησης, αποδείξεις, φωτογραφίες, ανεφοδιασμοί, υπενθυμίσεις και διαδρομές βρίσκονταν σε διαφορετικά μέσα ή εξαρτώνταν από συστήματα συνεργείων και κατασκευαστών.

Η μελέτη των τωρινών διαθέσιμων εφαρμογών έδειξε ότι υπάρχουν αρκετές σχετικές λύσεις. Το τελικό πρόβλημα δεν ήταν η πλήρης απουσία εφαρμογών, αλλά η περιορισμένη παρουσία web λύσεων που συγκεντρώνουν τις βασικές πληροφορίες προσωπικής διαχείρισης μοτοσυκλέτας σε ένα περιβάλλον.

### 3.5.2 Καθορισμός των στόχων

Το πρόβλημα μετατράπηκε σε στόχους σχεδιασμού. Βασικός στόχος ήταν η δημιουργία μιας web πλατφόρμας στην οποία ο χρήστης θα μπορούσε να οργανώνει περισσότερες από μία μοτοσυκλέτες και να συνδέει με κάθε όχημα το αντίστοιχο ιστορικό.

Οι στόχοι περιλάμβαναν τη διαχείριση λογαριασμών και οχημάτων, τη συντήρηση, τους ανεφοδιασμούς, τα έξοδα, τα ανταλλακτικά, τα αξεσουάρ, τα έγγραφα, τις φωτογραφίες, τις υπενθυμίσεις, τις διαδρομές GPX και τη συγκεντρωτική παρουσίαση πληροφοριών. Έτσι δημιουργήθηκαν απαιτήσεις για ασφάλεια, συντηρησιμότητα, ιδιωτική/ασφαλή διαχείριση αρχείων και δυνατότητα δοκιμών.

### 3.5.3 Σχεδιασμός και ανάπτυξη

Στο τρίτο στάδιο σχεδιάστηκε η αρχιτεκτονική, το μοντέλο δεδομένων και το UI του Moto-Vault. Η εφαρμογή χωρίστηκε σε τέσσερα βασικά κομμάτια: το frontend, το backend, τη σχεσιακή βάση δεδομένων και έναν ξεχωριστό χώρο για την αποθήκευση των αρχείων.

Η υλοποίηση έγινε βήμα βήμα. Κάθε νέα λειτουργία χτιζόταν ολοκληρωμένα σε όλα τα επίπεδα, από τη βάση και το backend, μέχρι το περιβάλλον του χρήστη και τους απαραίτητους ελέγχους.

Κατά τη διάρκεια της ανάπτυξης έγιναν αρκετές αλλαγές στο αρχικό πλάνο. Προστέθηκαν λειτουργίες όπως τα έγγραφα οχήματος, οι εισαγωγές και εξαγωγές δεδομένων, η δημιουργία PDF αναφορών και διαφορετικές δυνατότητες λογαριασμών. Οι τεχνικές λεπτομέρειες του σχεδιασμού και της υλοποίησης παρουσιάζονται στα Κεφάλαια 4, 5 και 6.

### 3.5.4 Επίδειξη του συστήματος

Η επίδειξη του Moto-Vault πραγματοποιείται μέσα από την παρουσίαση των βασικών λειτουργιών του τελικού συστήματος. Περιλαμβάνει τη δημιουργία λογαριασμού, τη διαχείριση μοτοσυκλετών, την καταχώριση δεδομένων συντήρησης και ανεφοδιασμών, την αποθήκευση αρχείων, τη δημιουργία υπενθυμίσεων και την προβολή διαδρομών GPX.

Το Κεφάλαιο 7 παρουσιάζει αυτές τις λειτουργίες με στιγμιότυπα οθόνης και επεξηγήσεις, ώστε να φαίνεται πώς οι διαφορετικές πληροφορίες συνδέονται με το ίδιο όχημα.

### 3.5.5 Αξιολόγηση

Η αξιολόγηση πραγματοποιήθηκε σε δύο επίπεδα.

Η τεχνική αξιολόγηση βασίστηκε σε αυτοματοποιημένες δοκιμές backend και frontend, καθώς και σε χειροκίνητο έλεγχο βασικών ροών και λειτουργιών.

Η αξιολόγηση χρηστικότητας πραγματοποιήθηκε με 12 μοτοσυκλετιστές. Οι συμμετέχοντες εξερεύνησαν ελεύθερα την εφαρμογή, χωρίς προκαθορισμένες εργασίες και χωρίς συγκεκριμένο χρονικό περιορισμό. Βοήθεια δόθηκε μόνο όταν χρειάστηκε να ενημερωθούν για το πού μπορούν να βρουν αρχεία GPX για δοκιμή.

Μετά τη χρήση της εφαρμογής, οι συμμετέχοντες συμπλήρωσαν το System Usability Scale και δύο πρόσθετες ερωτήσεις. Η αναλυτική διαδικασία και τα αποτελέσματα παρουσιάζονται στα Κεφάλαια 3.6 και 8.

### 3.5.6 Αντιστοίχιση με τη δομή της εργασίας

**Πίνακας 3.2**: Αντιστοίχιση των σταδίων της DSRM με τη δομή της εργασίας

| **Στάδιο DSRM** | **Περιεχόμενο της εργασίας** |
|---|---|
| Αναγνώριση προβλήματος | Κεφάλαια 1 και 2 |
| Καθορισμός στόχων | Κεφάλαια 1, 3 και 4 |
| Σχεδιασμός και ανάπτυξη | Κεφάλαια 4, 5 και 6 |
| Επίδειξη | Κεφάλαιο 7 |
| Αξιολόγηση | Κεφάλαιο 8 |
| Επικοινωνία | Γραπτή εργασία και τελική παρουσίαση |

Η αντιστοίχιση αυτή δείχνει ότι η DSRM χρησιμοποιείται για την οργάνωση ολόκληρης της εργασίας και όχι μόνο ως θεωρητική αναφορά.

## 3.6 Μεθοδολογία αξιολόγησης

Η αξιολόγηση του Moto-Vault οργανώθηκε σε δύο συμπληρωματικά επίπεδα:

1. τεχνική αξιολόγηση της λειτουργίας του συστήματος,
2. αξιολόγηση της αντιλαμβανόμενης χρηστικότητας από χρήστες.

Η διάκριση αυτή είναι απαραίτητη, επειδή η τεχνική ορθότητα δεν αρκεί από μόνη της για να θεωρηθεί επιτυχημένο ένα πληροφοριακό σύστημα. Μια εφαρμογή μπορεί να λειτουργεί σωστά, αλλά να παραμένει δύσκολη στην κατανόηση ή στη χρήση.

### 3.6.1 Τεχνική αξιολόγηση

Η τεχνική αξιολόγηση βασίζεται σε αυτοματοποιημένες και χειροκίνητες δοκιμές.

Οι αυτοματοποιημένες δοκιμές ελέγχουν βασικά τμήματα του backend και του frontend, όπως:

- επιχειρησιακούς κανόνες,
- validation,
- ασφάλεια και ιδιοκτησία πόρων,
- πρόσβαση στη βάση δεδομένων,
- διαχείριση αρχείων,
- συμπεριφορά του REST API,
- βασικά στοιχεία της διεπαφής.

Οι χειροκίνητοι έλεγχοι χρησιμοποιούνται για ολοκληρωμένες ροές, όπως η πλοήγηση, η μεταφόρτωση και λήψη αρχείων, η προβολή χαρτών, η αλλαγή γλώσσας και η χρήση της εφαρμογής σε διαφορετικά μεγέθη οθόνης.

Η υλοποίηση και τα αποτελέσματα των τεχνικών ελέγχων παρουσιάζονται στο Κεφάλαιο 6.

### 3.6.2 Αξιολόγηση χρηστικότητας

Για την αξιολόγηση της αντιλαμβανόμενης χρηστικότητας επιλέχθηκε το System Usability Scale, γνωστό ως SUS (Brooke, 1996).

Το SUS αποτελείται από δέκα δηλώσεις, οι οποίες απαντώνται σε πενταβάθμια κλίμακα συμφωνίας. Οι δηλώσεις εναλλάσσονται ανάμεσα σε θετική και αρνητική διατύπωση και παράγουν συνολική βαθμολογία από 0 έως 100.

Η επιλογή του SUS έγινε επειδή:

- είναι σύντομο,
- μπορεί να εφαρμοστεί σε διαφορετικά πληροφοριακά συστήματα,
- διαθέτει καθιερωμένο τρόπο υπολογισμού,
- επιτρέπει συνολική αποτίμηση της αντιλαμβανόμενης χρηστικότητας.

### 3.6.3 Πρόσθετες ερωτήσεις

Εκτός από τις δέκα δηλώσεις του SUS, προστέθηκαν δύο ερωτήσεις σχετικές με:

- την πρόθεση μελλοντικής χρήσης,
- την πρόθεση σύστασης της εφαρμογής σε άλλους μοτοσικλετιστές.

Οι ερωτήσεις αυτές δεν αποτελούν μέρος του επίσημου SUS και δεν επηρεάζουν τη βαθμολογία του. Αναλύονται ξεχωριστά ως συμπληρωματικές ενδείξεις αποδοχής της εφαρμογής.

### 3.6.4 Υπολογισμός της βαθμολογίας SUS

Η βαθμολογία υπολογίζεται ξεχωριστά για κάθε συμμετέχοντα.

Για τις θετικά διατυπωμένες ερωτήσεις 1, 3, 5, 7 και 9, από την απάντηση αφαιρείται η τιμή 1.

Για τις αρνητικά διατυπωμένες ερωτήσεις 2, 4, 6, 8 και 10, η απάντηση αφαιρείται από την τιμή 5.

Το άθροισμα των δέκα συνεισφορών πολλαπλασιάζεται επί 2,5 και προκύπτει συνολική βαθμολογία από 0 έως 100.

Για τη συνολική παρουσίαση μπορούν να χρησιμοποιηθούν ο μέσος όρος, η διάμεσος, η ελάχιστη και μέγιστη τιμή και η τυπική απόκλιση.

### 3.6.5 Σύνδεση με τα ερευνητικά ερωτήματα

Η τεχνική αξιολόγηση συνδέεται με την ορθή λειτουργία, την ασφάλεια και την αξιοπιστία του συστήματος.

Η αξιολόγηση SUS συνδέεται με το ερευνητικό ερώτημα που αφορά την αντίληψη των χρηστών για τη χρηστικότητα του Moto-Vault.

Η πραγματική διαδικασία, οι συμμετέχοντες, τα αποτελέσματα και οι περιορισμοί της αξιολόγησης παρουσιάζονται στο Κεφάλαιο 8.

# 4. Ανάλυση και Σχεδιασμός

## 4.1 Χρήστες και σενάρια χρήσης

Η ανάλυση χρηστών αποτελεί το πρώτο βήμα για τη μετατροπή του προβλήματος σε συγκεκριμένες απαιτήσεις. Το Moto-Vault έχει σχεδιαστεί ως εφαρμογή προσωπικής διαχείρισης μοτοσυκλέτας και δεν αποτελεί σύστημα συνεργείου, πλατφόρμα κατασκευαστή ή κοινωνικό δίκτυο αναβατών.

### 4.1.1 Κύριοι χρήστες

Ο βασικός χρήστης είναι ο ιδιώτης κάτοχος ή τακτικός χρήστης μίας ή περισσότερων μοτοσυκλετών. Μπορεί να δημιουργεί λογαριασμό, να καταχωρεί οχήματα και να οργανώνει πληροφορίες συντήρησης, ανεφοδιασμών, εξόδων, ανταλλακτικών, αξεσουάρ, εγγράφων, φωτογραφιών, υπενθυμίσεων και διαδρομών GPX.

Ο μη εγγεγραμμένος επισκέπτης έχει πρόσβαση μόνο στην αρχική σελίδα, στην εγγραφή, στην σύνδεση και στην ανάκτηση κωδικού.

Κάθε εγγεγραμμένος χρήστης έχει πρόσβαση αποκλειστικά στα δικά του οχήματα και στις καταχωρίσεις που συνδέονται με αυτά.

### 4.1.2 Εξωτερικά συστήματα

Το Moto-Vault συνεργάζεται με:

- το Google Identity για σύνδεση μέσω Google,
- υπηρεσία email (Resend) για επαλήθευση λογαριασμού και ανάκτηση κωδικού,
- τοπικό ή S3 συμβατό αποθηκευτικό χώρο για ιδιωτικά αρχεία,
- το OpenStreetMap για τα χαρτογραφικά δεδομένα που προβάλλονται μέσω Leaflet.

### 4.1.3 Δευτερεύοντες ενδιαφερόμενοι

Ένα συνεργείο μπορεί να ενημερωθεί από το ιστορικό ή από αναφορά PDF που παρέχει ο ιδιοκτήτης. Δεν διαθέτει όμως ακόμα ξεχωριστό λογαριασμό ή άμεση πρόσβαση στα δεδομένα.

Ένας μελλοντικός αγοραστής μπορεί επίσης να δει οργανωμένο ιστορικό που επιλέγει να του παρουσιάσει ο ιδιοκτήτης. Οι πληροφορίες αυτές δεν αποτελούν επίσημη πιστοποίηση.

### 4.1.4 Χαρακτηριστικά του κοινού

Οι χρήστες μπορεί να έχουν διαφορετικό επίπεδο τεχνικής εξοικείωσης/γνώσης. Οι περισσότεροι δεν καταγράφουν καμία πληροφορία. Πρέπει να χρησιμοποιηθούν συγκεκριμένες ορολογίες και να μην απαιτούνται τεχνικές γνώσεις.

### 4.1.5 Βασικά σενάρια χρήσης

**Σενάριο 1: Δημιουργία λογαριασμού και οχήματος**

Ο χρήστης δημιουργεί λογαριασμό με email και κωδικό ή συνδέεται μέσω Google. Μετά την είσοδο καταχωρεί τη μηχανή του και τα βασικά της στοιχεία.

**Σενάριο 2: Καταγραφή συντήρησης και εξαρτημάτων**
Μετά από service, ο χρήστης προσθέτει ημερομηνία, χιλιόμετρα, περιγραφή, κόστος και προαιρετικά φωτογραφίες ή παραστατικά. Μπορεί επίσης να συνδέσει ανταλλακτικά ή αξεσουάρ με την εργασία.

**Σενάριο 3: Παρακολούθηση καυσίμων και εξόδων**
Σε κάθε ανεφοδιασμό βενζίνης ο χρήστης καταχωρεί χιλιόμετρα, ποσότητα και κόστος. Το σύστημα υπολογίζει κατανάλωση και συγκεντρωτικά οικονομικά στοιχεία όταν υπάρχουν επαρκή δεδομένα.

**Σενάριο 4: Έγγραφα, υπενθυμίσεις και φωτογραφίες**
Ο χρήστης αποθηκεύει έγγραφα οχήματος, ημερομηνίες λήξης και φωτογραφίες. Δημιουργεί επίσης υπενθυμίσεις με βάση ημερομηνία, χιλιόμετρα ή και τα δύο.

**Σενάριο 5: Διαδρομές, αναφορές και μεταφορά δεδομένων**
Ο χρήστης εισάγει αρχείο GPX και βλέπει τη διαδρομή σε χάρτη. Μπορεί ακόμη να εξάγει δεδομένα, να δημιουργεί αντίγραφο ασφαλείας λογαριασμού και να παράγει αναφορά ιστορικού συντήρησης σε PDF.

Οι τυπικές περιπτώσεις χρήσης και οι εναλλακτικές ροές παρουσιάζονται στην Ενότητα 4.5.

## 4.2 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις δυνατότητες που πρέπει να παρέχει το Moto-Vault. Κάθε απαίτηση λαμβάνει μοναδικό κωδικό της μορφής ΛΑ-Χ.Υ και οργανώνεται ανά λειτουργική περιοχή.

### 4.2.1 Λογαριασμός και αυθεντικοποίηση

**ΛΑ-1.1**: Δημιουργία λογαριασμού με email και κωδικό.
**ΛΑ-1.2**: Επικύρωση στοιχείων εγγραφής και αποφυγή διπλών λογαριασμών.
**ΛΑ-1.3**: Επαλήθευση email και δυνατότητα επαναποστολής.
**ΛΑ-1.4**: Σύνδεση με email και κωδικό.
**ΛΑ-1.5**: Σύνδεση μέσω Google με επαλήθευση token στο backend.
**ΛΑ-1.6**: Ασφαλής σύνδεση Google identity με υπάρχοντα λογαριασμό.
**ΛΑ-1.7**: Προβολή και ενημέρωση προφίλ και προτιμήσεων.
**ΛΑ-1.8**: Αλλαγή και ανάκτηση κωδικού.
**ΛΑ-1.9**: Ανάκληση ενεργών συνεδριών.
**ΛΑ-1.10**: Διαγραφή λογαριασμού και δεδομένων.
**ΛΑ-1.11**: Δημιουργία και επαναφορά πλήρους backup λογαριασμού.
**ΛΑ-1.12**: Προβολή πλάνου και χρήσης αποθηκευτικού χώρου.

### 4.2.2 Διαχείριση οχημάτων

**ΛΑ-2.1**: Δημιουργία, προβολή, επεξεργασία και διαγραφή οχήματος.
**ΛΑ-2.2**: Καταχώριση κατασκευαστή, μοντέλου, έτους, κυβισμού, αριθμού κυκλοφορίας και VIN.
**ΛΑ-2.3**: Προαιρετική τρέχουσα χιλιομετρική ένδειξη.
**ΛΑ-2.4**: Αυτόματη ενημέρωση χιλιομέτρων όταν καταχωρίζεται μεγαλύτερη έγκυρη τιμή.
**ΛΑ-2.5**: Διαχείριση περισσότερων από μίας μοτοσυκλετών.
**ΛΑ-2.6**: Ορισμός κύριου οχήματος και κύριας φωτογραφίας.

### 4.2.3 Ιστορικό συντήρησης

**ΛΑ-3.1**: Δημιουργία, προβολή, επεξεργασία και διαγραφή εγγραφών συντήρησης.
**ΛΑ-3.2**: Καταχώριση ημερομηνίας, χιλιομέτρων, κατηγορίας, τίτλου, περιγραφής, συνεργείου, κόστους και νομίσματος.
**ΛΑ-3.3**: Προσθήκη, λήψη και διαγραφή συνημμένων.
**ΛΑ-3.4**: Σύνδεση εγγραφής με ανταλλακτικά και αξεσουάρ.
**ΛΑ-3.5**: Ενημέρωση της χιλιομετρικής ένδειξης όταν χρειάζεται.
**ΛΑ-3.6**: Παραγωγή αναφοράς ιστορικού συντήρησης σε PDF.

### 4.2.4 Καύσιμα και στατιστικά

**ΛΑ-4.1**: Δημιουργία, προβολή, επεξεργασία και διαγραφή ανεφοδιασμών.
**ΛΑ-4.2**: Καταχώριση ημερομηνίας, χιλιομέτρων, λίτρων, κόστους, τιμής ανά λίτρο και πλήρους ή μερικού ανεφοδιασμού.
**ΛΑ-4.3**: Υπολογισμός κατανάλωσης όταν υπάρχουν κατάλληλοι διαδοχικοί πλήρεις ανεφοδιασμοί.
**ΛΑ-4.4**: Συνυπολογισμός μερικών ανεφοδιασμών.
**ΛΑ-4.5**: Υπολογισμός κόστους ανά απόσταση και παρουσίαση στατιστικών.

### 4.2.5 Αξεσουάρ και ανταλλακτικά

**ΛΑ-5.1**: Δημιουργία, προβολή, επεξεργασία και διαγραφή αξεσουάρ.
**ΛΑ-5.2**: Δημιουργία, προβολή, επεξεργασία και διαγραφή ανταλλακτικών.
**ΛΑ-5.3**: Καταχώριση στοιχείων αγοράς, κόστους, ημερομηνιών και σημειώσεων.
**ΛΑ-5.4**: Προσθήκη και διαχείριση φωτογραφιών ή παραστατικών.
**ΛΑ-5.5**: Προαιρετική σύνδεση με εγγραφή συντήρησης.

### 4.2.6 Έγγραφα

**ΛΑ-6.1**: Δημιουργία, προβολή, επεξεργασία και διαγραφή εγγράφου.
**ΛΑ-6.2**: Καταχώριση τύπου, τίτλου, ημερομηνιών και σημειώσεων.
**ΛΑ-6.3**: Αποθήκευση ιδιωτικού αρχείου.
**ΛΑ-6.4**: Αυτόματος χαρακτηρισμός κατάστασης λήξης.

### 4.2.7 Υπενθυμίσεις

**ΛΑ-7.1**: Δημιουργία, προβολή, επεξεργασία και διαγραφή υπενθυμίσεων.
**ΛΑ-7.2**: Ορισμός κριτηρίου ημερομηνίας, χιλιομέτρων ή και των δύο.
**ΛΑ-7.3**: Αυτόματος υπολογισμός κατάστασης.
**ΛΑ-7.4**: Ολοκλήρωση ή επανεργοποίηση υπενθύμισης.

### 4.2.8 Φωτογραφίες οχήματος

**ΛΑ-8.1**: Μεταφόρτωση φωτογραφιών.
**ΛΑ-8.2**: Προβολή και λήψη φωτογραφιών.
**ΛΑ-8.3**: Ορισμός κύριας φωτογραφίας.
**ΛΑ-8.4**: Διαγραφή φωτογραφίας.

### 4.2.9 Διαδρομές GPX

**ΛΑ-9.1**: Μεταφόρτωση και επικύρωση αρχείου GPX.
**ΛΑ-9.2**: Εξαγωγή απόστασης, διάρκειας και αριθμού σημείων.
**ΛΑ-9.3**: Προβολή διαδρομής σε χάρτη.
**ΛΑ-9.4**: Προβολή, επεξεργασία metadata και διαγραφή.
**ΛΑ-9.5**: Λήψη του αρχικού αρχείου.

### 4.2.10 Dashboard και συγκεντρωτικά στοιχεία

**ΛΑ-10.1**: Προβολή βασικών στατιστικών ανά όχημα.
**ΛΑ-10.2**: Προβολή πρόσφατων εργασιών, δραστηριότητας, υπενθυμίσεων και εγγράφων που λήγουν.
**ΛΑ-10.3**: Προβολή συνολικού κόστους και οικονομικών στοιχείων.
**ΛΑ-10.4**: Προβολή συνολικών εξόδων γκαράζ με υποστήριξη διαφορετικών νομισμάτων.

### 4.2.11 Εισαγωγή, εξαγωγή και αναφορές

**ΛΑ-11.1**: Εξαγωγή δεδομένων οχήματος σε ZIP με CSV.
**ΛΑ-11.2**: Προεπισκόπηση εισαγωγής πριν τη δημιουργία δεδομένων.
**ΛΑ-11.3**: Επιβεβαίωση και εκτέλεση εισαγωγής.
**ΛΑ-11.4**: Πλήρες backup και restore λογαριασμού.
**ΛΑ-11.5**: Παραγωγή PDF αναφοράς συντήρησης.

### 4.2.12 Προτιμήσεις και διεθνοποίηση

**ΛΑ-12.1**: Επιλογή ελληνικής ή αγγλικής γλώσσας.
**ΛΑ-12.2**: Επιλογή μονάδων απόστασης, όγκου και κατανάλωσης.
**ΛΑ-12.3**: Επιλογή νομίσματος και μορφής ημερομηνίας.
**ΛΑ-12.4**: Εφαρμογή των προτιμήσεων σε όλη τη διεπαφή.

### 4.2.13 Έλεγχος πρόσβασης και αρχεία

**ΛΑ-13.1**: Έλεγχος ιδιοκτησίας για κάθε όχημα και σχετικό πόρο.
**ΛΑ-13.2**: Απόκρυψη της ύπαρξης πόρων άλλων χρηστών.
**ΛΑ-13.3**: Πρόσβαση σε ιδιωτικά αρχεία μόνο μέσω backend.
**ΛΑ-13.4**: Εφαρμογή ορίων πλήθους, τύπου και μεγέθους αρχείων.
**ΛΑ-13.5**: Παρουσίαση κατανοητών μηνυμάτων επιτυχίας και αποτυχίας.

Οι λεπτομέρειες υλοποίησης παρουσιάζονται στα Κεφάλαια 5 και 6.

## 4.3 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις περιγράφουν τα ποιοτικά χαρακτηριστικά και τους περιορισμούς του Moto-Vault. Κάθε απαίτηση λαμβάνει κωδικό της μορφής ΜΛΑ-Χ.Υ.

### 4.3.1 Ασφάλεια και ιδιωτικότητα

**ΜΛΑ-1.1**: Οι κωδικοί πρέπει να αποθηκεύονται σε ασφαλή κατακερματισμένη μορφή.
**ΜΛΑ-1.2**: Κάθε προστατευμένη λειτουργία πρέπει να απαιτεί έγκυρη αυθεντικοποίηση.
**ΜΛΑ-1.3**: Τα tokens εξωτερικών παρόχων πρέπει να επαληθεύονται στο backend.
**ΜΛΑ-1.4**: Κάθε πρόσβαση σε όχημα ή σχετικό πόρο πρέπει να ελέγχει την ιδιοκτησία.
**ΜΛΑ-1.5**: Το σύστημα δεν πρέπει να αποκαλύπτει την ύπαρξη πόρων άλλων χρηστών.
**ΜΛΑ-1.6**: Τα αρχεία πρέπει να παραμένουν ιδιωτικά.
**ΜΛΑ-1.7**: Η διαχείριση ονομάτων και διαδρομών αρχείων πρέπει να αποτρέπει μη ασφαλείς διαδρομές.
**ΜΛΑ-1.8**: Η επεξεργασία XML πρέπει να προστατεύεται από XXE.
**ΜΛΑ-1.9**: Μυστικά και credentials δεν πρέπει να αποθηκεύονται στον πηγαίο κώδικα.
**ΜΛΑ-1.10**: Οι συνεδρίες πρέπει να μπορούν να ανακληθούν με ελεγχόμενο τρόπο.

### 4.3.2 Ακεραιότητα και προστασία δεδομένων

**ΜΛΑ-2.1**: Όλες οι είσοδοι πρέπει να επικυρώνονται στο backend.
**ΜΛΑ-2.2**: Κρίσιμοι κανόνες πρέπει να υποστηρίζονται και από περιορισμούς της βάσης όπου είναι κατάλληλο.
**ΜΛΑ-2.3**: Οι αλλαγές του schema πρέπει να εκτελούνται με ελεγχόμενη και επαναλήψιμη διαδικασία.
**ΜΛΑ-2.4**: Οι διαγραφές πρέπει να διαχειρίζονται σωστά τις εξαρτώμενες εγγραφές και τα σχετικά αρχεία.
**ΜΛΑ-2.5**: Το σύστημα πρέπει να εντοπίζει ασυνέπειες ανάμεσα σε metadata και πραγματικά αρχεία.
**ΜΛΑ-2.6**: Τα δυαδικά αρχεία δεν πρέπει να αποθηκεύονται στη σχεσιακή βάση.
**ΜΛΑ-2.7**: Η εισαγωγή δεδομένων πρέπει να απαιτεί προεπισκόπηση και επιβεβαίωση.
**ΜΛΑ-2.8**: Οι υπολογισμοί κατανάλωσης, κόστους και λήξεων πρέπει να βασίζονται σε σταθερούς κανόνες.

### 4.3.3 Αξιοπιστία και διαχείριση σφαλμάτων

**ΜΛΑ-3.1**: Τα σφάλματα του API πρέπει να επιστρέφονται σε συνεπή μορφή.
**ΜΛΑ-3.2**: Πρέπει να διακρίνεται η μη εύρεση από την προσωρινή τεχνική αποτυχία.
**ΜΛΑ-3.3**: Το API πρέπει να χρησιμοποιεί κατάλληλους HTTP status codes.
**ΜΛΑ-3.4**: Η μη διαθεσιμότητα προαιρετικής υπηρεσίας δεν πρέπει να διακόπτει τις υπόλοιπες λειτουργίες.
**ΜΛΑ-3.5**: Τα τεχνικά σφάλματα πρέπει να μετατρέπονται σε κατανοητά μηνύματα.
**ΜΛΑ-3.6**: Οι λειτουργίες που συνδυάζουν βάση και αρχεία πρέπει να περιορίζουν τη μερική αποτυχία.
**ΜΛΑ-3.7**: Η εφαρμογή πρέπει να ξεκινά μόνο όταν η βάση βρίσκεται στην αναμενόμενη κατάσταση.

### 4.3.4 Απόδοση

**ΜΛΑ-4.1**: Οι βασικές λειτουργίες πρέπει να ολοκληρώνονται χωρίς αισθητή καθυστέρηση σε κανονικό φορτίο.
**ΜΛΑ-4.2**: Τα μεγάλα αρχεία δεν πρέπει να περιλαμβάνονται σε απλές JSON αποκρίσεις.
**ΜΛΑ-4.3**: Η λήψη αρχείων πρέπει να γίνεται αποδοτικά.
**ΜΛΑ-4.4:** Πρέπει να εφαρμόζονται όρια μεγέθους και πλήθους uploads.
**ΜΛΑ-4.5**: Η επεξεργασία GPX πρέπει να περιορίζεται σε υποστηριζόμενα μεγέθη.
**ΜΛΑ-4.6**: Τα ερωτήματα ιδιοκτησίας και λίστας πρέπει να εκτελούνται αποδοτικά.
**ΜΛΑ-4.7**: Βαριές βιβλιοθήκες της διεπαφής πρέπει να φορτώνονται μόνο όταν χρειάζονται.

### 4.3.5 Χρηστικότητα

**ΜΛΑ-5.1**: Η διεπαφή πρέπει να χρησιμοποιεί σαφείς τίτλους, ετικέτες και ενέργειες.
**ΜΛΑ-5.2**: Οι σελίδες οχήματος πρέπει να ακολουθούν κοινή πλοήγηση.
**ΜΛΑ-5.3**: Ο χρήστης πρέπει να λαμβάνει άμεση ανάδραση.
**ΜΛΑ-5.4**: Οι καταστροφικές ενέργειες πρέπει να απαιτούν επιβεβαίωση.
**ΜΛΑ-5.5**: Δεν πρέπει να εμφανίζονται ακατέργαστα τεχνικά μηνύματα.
**ΜΛΑ-5.6**: Πρέπει να υποστηρίζονται ελληνικά και αγγλικά.
**ΜΛΑ-5.7**: Οι φόρμες πρέπει να παρουσιάζουν σαφή validation errors.
**ΜΛΑ-5.8**: Η διεπαφή πρέπει να λειτουργεί σε υπολογιστή, tablet και κινητό.
**ΜΛΑ-5.9**: Οι κενές καταστάσεις πρέπει να εξηγούν την επόμενη διαθέσιμη ενέργεια.

### 4.3.6 Προσβασιμότητα

**ΜΛΑ-6.1**: Η πλοήγηση πρέπει να είναι δυνατή με πληκτρολόγιο όπου εφαρμόζεται.
**ΜΛΑ-6.2**: Τα πεδία φόρμας πρέπει να διαθέτουν σαφείς ετικέτες.
**ΜΛΑ-6.3**: Τα εικονίδια ενεργειών πρέπει να έχουν προσβάσιμη περιγραφή.
**ΜΛΑ-6.4**: Η χρωματική πληροφορία δεν πρέπει να είναι ο μοναδικός τρόπος μετάδοσης κατάστασης.
**ΜΛΑ-6.5**: Τα modal και τα μηνύματα πρέπει να διαχειρίζονται σωστά την εστίαση.
**ΜΛΑ-6.6**: Το περιεχόμενο πρέπει να παραμένει κατανοητό σε διαφορετικά μεγέθη οθόνης.

### 4.3.7 Συντηρησιμότητα και επεκτασιμότητα

**ΜΛΑ-7.1**: Το backend και το frontend πρέπει να οργανώνονται σε σαφείς λειτουργικές μονάδες.
**ΜΛΑ-7.2**: Τα persistence entities δεν πρέπει να εκτίθενται απευθείας στο API.
**ΜΛΑ-7.3**: Η αποθήκευση αρχείων πρέπει να είναι ανεξάρτητη από συγκεκριμένο πάροχο.
**ΜΛΑ-7.4**: Οι ρυθμίσεις περιβάλλοντος πρέπει να διαχωρίζονται από τον κώδικα.
**ΜΛΑ-7.5**: Η προσθήκη νέας λειτουργικής περιοχής δεν πρέπει να απαιτεί σημαντική αναδόμηση.
**ΜΛΑ-7.6**: Το σύστημα πρέπει να μπορεί να υποστηρίξει νέες αναφορές, παρόχους και κατηγορίες δεδομένων.
**ΜΛΑ-7.7**: Η επέκταση σε άλλους τύπους οχημάτων θεωρείται μελλοντική επιλογή.

### 4.3.8 Φορητότητα και συμβατότητα

**ΜΛΑ-8.1**: Η εφαρμογή πρέπει να λειτουργεί σε όλους τους browsers.
**ΜΛΑ-8.2**: Backend και frontend πρέπει να μπορούν να εκτελούνται σε containers.
**ΜΛΑ-8.3**: Βάση και storage πρέπει να ρυθμίζονται ανά περιβάλλον.
**ΜΛΑ-8.4**: Οι μορφές εξαγωγής πρέπει να είναι ανοικτές και μεταφέρσιμες.
**ΜΛΑ-8.5**: Οι μη ευαίσθητες runtime ρυθμίσεις πρέπει να παρέχονται χωρίς νέο build.

### 4.3.9 Δοκιμασιμότητα

**ΜΛΑ-9.1**: Οι βασικοί κανόνες backend πρέπει να ελέγχονται αυτοματοποιημένα.
**ΜΛΑ-9.2**: Οι βασικές λειτουργίες frontend πρέπει να καλύπτονται από επαναλήψιμους ελέγχους.
**ΜΛΑ-9.3**: Οι αλλαγές πρέπει να μπορούν να ελέγχονται σε CI.
**ΜΛΑ-9.4**: Οι εξωτερικές εξαρτήσεις πρέπει να μπορούν να προσομοιώνονται σε tests.
**ΜΛΑ-9.5**: Τα migrations και οι βασικές ροές εισαγωγής πρέπει να είναι επαναλήψιμα.

### 4.3.10 Λειτουργία και παρατηρησιμότητα

**ΜΛΑ-10.1**: Η εφαρμογή πρέπει να διαθέτει health checks.
**ΜΛΑ-10.2**: Τα logs πρέπει να βοηθούν στη διάγνωση χωρίς να εκθέτουν μυστικά ή προσωπικά δεδομένα.
**ΜΛΑ-10.3**: Οι background εργασίες πρέπει να έχουν σαφή κατάσταση και δυνατότητα επανάληψης.
**ΜΛΑ-10.4**: Η εγκατάσταση πρέπει να υποστηρίζει ξεχωριστές ρυθμίσεις development και production.
**ΜΛΑ-10.5**: Η βάση και τα αρχεία πρέπει να καλύπτονται από σχέδιο αντιγράφων ασφαλείας και ανάκτησης.

Οι τεχνολογίες που χρησιμοποιούνται για την κάλυψη των απαιτήσεων παρουσιάζονται στα Κεφάλαια 5 και 6.

## 4.4 Ιεράρχηση απαιτήσεων

Το ξεκαθάρισμα των λειτουργιών μας βοηθάει να δούμε τι είναι απολύτως απαραίτητο για την εφαρμογή, τι είναι απλά χρήσιμο και τι αφήνουμε για μετά.

Χρησιμοποιείται η μέθοδος **MoSCoW**:

- **Must**: απαραίτητες απαιτήσεις,
- **Should**: σημαντικές απαιτήσεις,
- **Could**: επιθυμητές απαιτήσεις,
- **Won't**: απαιτήσεις εκτός της παρούσας έκδοσης.

Η κατάταξη βασίστηκε στη σύνδεση με το βασικό πρόβλημα, στη χρησιμότητα, στις εξαρτήσεις, στον διαθέσιμο χρόνο και στη δυνατότητα αξιόπιστης ολοκλήρωσης.

### 4.4.1 Ιεράρχηση απαιτήσεων

**Πίνακας 4.1**: Κατάταξη λειτουργικών περιοχών με τη μέθοδο MoSCoW

| **Λειτουργική περιοχή** | **Must** | **Should** | **Could** | **Won't** |
|---|---|---|---|---|
| Λογαριασμός | Εγγραφή, σύνδεση, επαλήθευση, ανάκτηση, διαγραφή | Google, προτιμήσεις | Πλάνα και επιπλέον όρια | Επαγγελματικοί ρόλοι |
| Οχήματα | Δημιουργία, προβολή, επεξεργασία, διαγραφή | Κύριο όχημα και φωτογραφία | Πρόσθετα πεδία | Αυτόματη ανάκτηση VIN |
| Συντήρηση | Ιστορικό, κόστος, χιλιόμετρα, συνημμένα | Σύνδεση με εξαρτήματα, PDF | Πρόσθετες κατηγορίες | Επίσημη πιστοποίηση συνεργείου |
| Καύσιμα | Ανεφοδιασμοί, κατανάλωση, κόστος | Στατιστικά και γραφήματα | Προηγμένες αναλύσεις | Αυτόματη ανάγνωση από όχημα |
| Αξεσουάρ και ανταλλακτικά | Βασική καταγραφή | Φωτογραφίες, παραστατικά, σύνδεση με service | Εμπλουτισμένη βάση δεδομένων | Αυτόματη αναγνώριση προϊόντων |
| Έγγραφα | Αποθήκευση και ημερομηνίες | Κατάσταση λήξης | Ειδοποιήσεις email | Διασύνδεση με κρατικά μητρώα |
| Υπενθυμίσεις | Ημερομηνία και χιλιόμετρα | Συνδυασμένα κριτήρια | Προηγμένες ειδοποιήσεις | Push notifications κινητού |
| Φωτογραφίες | Μεταφόρτωση, προβολή, διαγραφή | Κύρια φωτογραφία | Βελτιστοποίηση εικόνων | Αυτόματη ταξινόμηση AI |

| | | | | |
|---|---|---|---|---|
| GPX | Εισαγωγή και προβολή | Στατιστικά διαδρομής | Πρόσθετη ανάλυση | Ζωντανή καταγραφή GPS |
| Dashboard | Βασικά στοιχεία | Δραστηριότητα και λήξεις | Περισσότερα γραφήματα | Προγνωστική ανάλυση |
| Δεδομένα | Βασική εξαγωγή | Import, backup, restore, PDF | Περισσότερες μορφές | Πλήρης συγχρονισμός τρίτων υπηρεσιών |
| Διεπαφή | Responsive, ελληνικά, αγγλικά | Προτιμήσεις μονάδων και νομίσματος | Επιπλέον γλώσσες | Native εφαρμογή Android/iOS |

### 4.4.2 Μη λειτουργικές προτεραιότητες

Ως **Must** θεωρούνται η ασφάλεια, ο έλεγχος ιδιοκτησίας, η ιδιωτικότητα αρχείων, η ακεραιότητα δεδομένων, η συνεπής διαχείριση σφαλμάτων, η βασική χρηστικότητα και η δοκιμασιμότητα.

Ως **Should** θεωρούνται η προσβασιμότητα, η απόδοση, η συντηρησιμότητα, η ανεξαρτησία storage, η φορητότητα και η παρατηρησιμότητα.

Ως **Could** θεωρούνται οι επιπλέον γλώσσες, τα προηγμένα στατιστικά, οι περισσότερες ειδοποιήσεις και η περαιτέρω βελτιστοποίηση πολυμέσων.

Εκτός έκδοσης παραμένουν οι λογαριασμοί συνεργείων, η native εφαρμογή, η live πλοήγηση, η κοινότητα, η σύνδεση με OBD ή telematics, η αυτόματη ανάκτηση μέσω VIN και η καταχώριση μέσω OCR ή AI.

Η τελική υλοποίηση ξεπέρασε σε ορισμένα σημεία τον αρχικό ελάχιστο πυρήνα, καθώς περιλαμβάνει import/export, backup/restore, PDF αναφορά, έγγραφα και επιπλέον λειτουργίες λογαριασμού.

## 4.5 Περιπτώσεις χρήσης

Οι περιπτώσεις χρήσης περιγράφουν πώς αλληλεπιδρά ο χρήστης με το Moto-Vault. Ουσιαστικά, συμπληρώνουν τα σενάρια της Ενότητας 4.1 και συνδέονται άμεσα με τις λειτουργικές απαιτήσεις της Ενότητας 4.2.

### 4.5.1 Ρόλοι χρηστών

- **Επισκέπτης**, για εγγραφή, σύνδεση και ανάκτηση κωδικού.
- **Εγγεγραμμένος χρήστης**, για όλες τις λειτουργίες προσωπικής διαχείρισης.
- **Google Identity**, για σύνδεση μέσω Google.
- **Υπηρεσία email**, για επαλήθευση και ανάκτηση.
- **Αποθηκευτικός χώρος**, για ιδιωτικά αρχεία.
- **OpenStreetMap**, για χαρτογραφικά δεδομένα.

### 4.5.2 Κύριες περιπτώσεις χρήσης

**ΠΧ-1 Δημιουργία λογαριασμού**

**Χρήστης**: Επισκέπτης
**Προϋπόθεση**: Δεν υπάρχει λογαριασμός με το ίδιο email.

1. Ο επισκέπτης συμπληρώνει τη φόρμα.
2. Το σύστημα επικυρώνει τα στοιχεία.
3. Δημιουργείται ο λογαριασμός.
4. Αποστέλλεται μήνυμα επαλήθευσης.
5. Ο χρήστης επιβεβαιώνει το email.

**Εναλλακτικές ροές**: μη έγκυρα στοιχεία, ήδη χρησιμοποιημένο email, αποτυχία αποστολής.

**ΠΧ-2 Σύνδεση χρήστη**

1. Ο χρήστης εισάγει email και κωδικό ή επιλέγει Google.
2. Το backend επαληθεύει την ταυτότητα.
3. Δημιουργείται token.
4. Ο χρήστης εισέρχεται στην εφαρμογή.

**ΠΧ-3 Διαχείριση οχήματος**

1. Ο χρήστης ανοίγει το γκαράζ.
2. Δημιουργεί ή επιλέγει όχημα.
3. Προβάλλει ή αλλάζει τα στοιχεία του.
4. Μπορεί να το ορίσει ως κύριο ή να το διαγράψει.

**ΠΧ-4 Καταχώριση συντήρησης**

1. Ο χρήστης ανοίγει το ιστορικό.
2. Συμπληρώνει τα στοιχεία εργασίας.
3. Αποθηκεύει την εγγραφή.
4. Προαιρετικά προσθέτει αρχεία, ανταλλακτικά ή αξεσουάρ.

**Εναλλακτικές ροές**: μη έγκυρα δεδομένα, μη υποστηριζόμενο αρχείο, υπέρβαση ορίων.

**ΠΧ-5 Καταχώριση ανεφοδιασμού**

1. Ο χρήστης καταχωρίζει ημερομηνία, χιλιόμετρα, λίτρα και κόστος.
2. Δηλώνει αν ο ανεφοδιασμός ήταν πλήρης.
3. Το σύστημα αποθηκεύει την εγγραφή.
4. Αν υπάρχουν επαρκή δεδομένα, ενημερώνει τα στατιστικά.

**ΠΧ-6 Διαχείριση αξεσουάρ ή ανταλλακτικού**

1. Ο χρήστης καταχωρίζει στοιχεία αγοράς και κόστους.
2. Προαιρετικά προσθέτει φωτογραφία ή παραστατικό.
3. Μπορεί να συνδέσει την εγγραφή με εργασία συντήρησης.

**ΠΧ-7 Έγγραφο και υπενθύμιση**

1. Ο χρήστης αποθηκεύει έγγραφο και ημερομηνία λήξης.
2. Το σύστημα εμφανίζει την κατάστασή του.
3. Ο χρήστης δημιουργεί υπενθύμιση βάσει ημερομηνίας ή χιλιομέτρων.

**ΠΧ-8 Φωτογραφίες**

1. Ο χρήστης μεταφορτώνει φωτογραφία.
2. Η φωτογραφία συνδέεται με το όχημα.
3. Μπορεί να οριστεί ως κύρια ή να διαγραφεί.

**ΠΧ-9 Εισαγωγή GPX**

1. Ο χρήστης επιλέγει αρχείο GPX.
2. Το backend το επικυρώνει και το αναλύει.
3. Αποθηκεύονται τα βασικά στοιχεία.
4. Η διαδρομή προβάλλεται σε χάρτη.

**ΠΧ-10 Dashboard**

1. Ο χρήστης ανοίγει το dashboard.
2. Το σύστημα συγκεντρώνει στοιχεία από τις σχετικές ενότητες.
3. Παρουσιάζονται κόστη, κατανάλωση, πρόσφατες εργασίες, λήξεις και υπενθυμίσεις.

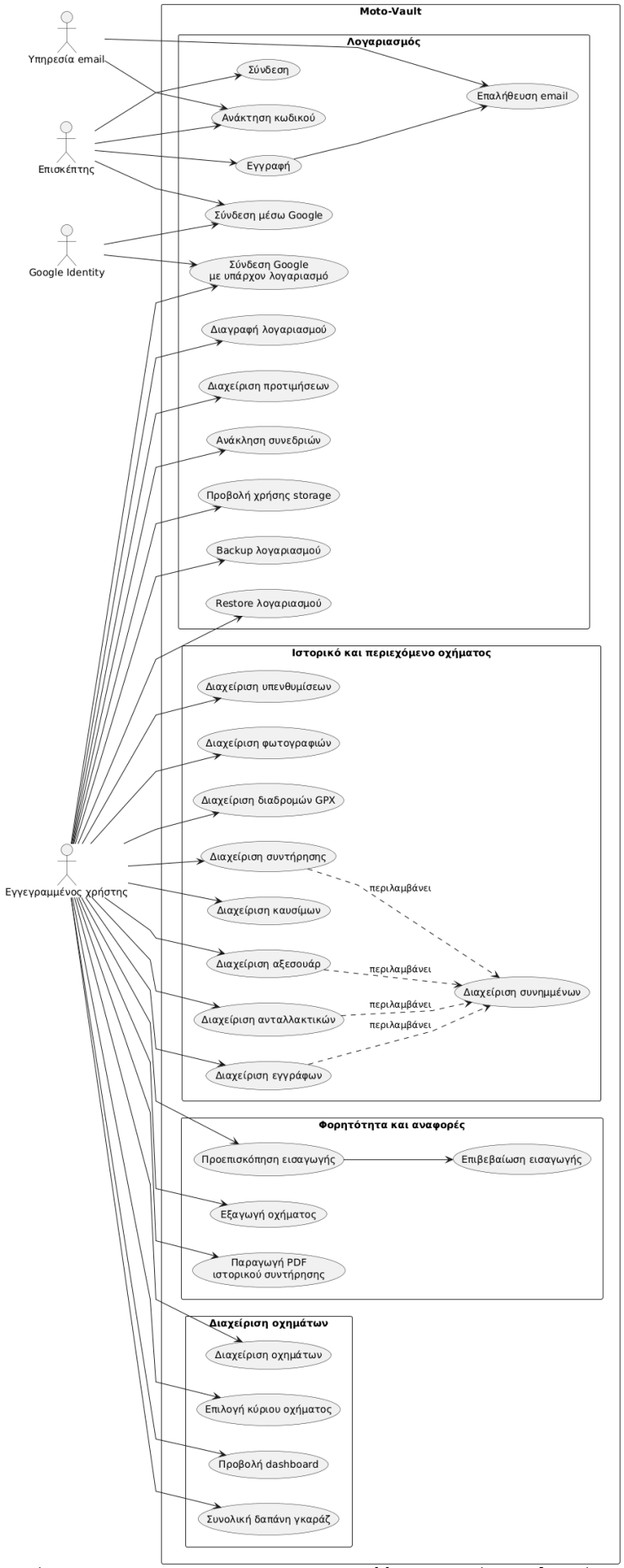
**ΠΧ-11 Εξαγωγή, εισαγωγή και αναφορά**

1. Ο χρήστης ζητά εξαγωγή ή backup.
2. Το σύστημα δημιουργεί το κατάλληλο αρχείο.
3. Για εισαγωγή παρουσιάζεται προεπισκόπηση.
4. Η εισαγωγή εκτελείται μετά από επιβεβαίωση.
5. Ο χρήστης μπορεί να δημιουργήσει PDF ιστορικού.

### 4.5.3 Διάγραμμα περιπτώσεων χρήσης

Το διάγραμμα παρουσιάζει τον επισκέπτη, τον εγγεγραμμένο χρήστη, τις κύριες λειτουργικές περιοχές και τα εξωτερικά συστήματα. Δεν εμφανίζει κάθε ενέργεια δημιουργίας, επεξεργασίας και διαγραφής ξεχωριστά.

**Σχήμα 4.1**: Διάγραμμα βασικών περιπτώσεων χρήσης του Moto-Vault


Moto-Vault
Λογαριασμός
Υπηρεσία email
Επισκέπτης
Google Identity
Σύνδεση
Ανάκτηση κωδικού
Εγγραφή
Σύνδεση μέσω Google
Σύνδεση Google με υπάρχον λογαριασμό
Επαλήθευση email
Διαγραφή λογαριασμού
Διαχείριση προτιμήσεων
Ανάκληση συνεδριών
Προβολή χρήσης storage
Backup λογαριασμού
Restore λογαριασμού
Ιστορικό και περιεχόμενο οχήματος
Διαχείριση υπενθυμίσεων
Διαχείριση φωτογραφιών
Διαχείριση διαδρομών GPX
Διαχείριση συντήρησης
περιλαμβάνει
Διαχείριση καυσίμων
Διαχείριση αξεσουάρ
περιλαμβάνει
Διαχείριση συνημμένων
Διαχείριση ανταλλακτικών
περιλαμβάνει
περιλαμβάνει
Διαχείριση εγγράφων
Εγγεγραμμένος χρήστης
Φορητότητα και αναφορές
Προεπισκόπηση εισαγωγής
Επιβεβαίωση εισαγωγής
Εξαγωγή οχήματος
Παραγωγή PDF ιστορικού συντήρησης
Διαχείριση οχημάτων
Διαχείριση οχημάτων
Επιλογή κύριου οχήματος
Προβολή dashboard
Συνολική δαπάνη γκαράζ

## 4.6 Αρχιτεκτονικός σχεδιασμός

Το Moto-Vault ακολουθεί αρχιτεκτονική διαδικτυακής εφαρμογής με διακριτό frontend, backend, σχεσιακή βάση δεδομένων και υπηρεσία αποθήκευσης αρχείων.

### 4.6.1 Συνολική αρχιτεκτονική

Ο χρήστης αλληλεπιδρά με το Angular frontend μέσω του browser. Το frontend επικοινωνεί με το Spring Boot backend μέσω REST API. Το backend εφαρμόζει την επιχειρησιακή λογική, την αυθεντικοποίηση, τους ελέγχους ιδιοκτησίας, την πρόσβαση στη βάση και τη διαχείριση αρχείων.

Η PostgreSQL αποθηκεύει τα δομημένα δεδομένα και τα metadata των αρχείων. Το πραγματικό περιεχόμενο των αρχείων αποθηκεύεται σε τοπικό filesystem ή σε S3 συμβατό object storage.

### 4.6.2 Monorepo και διαχωρισμός εφαρμογών

Το repository περιλαμβάνει:

- backend/, για το Spring Boot REST API,
- frontend/, για την Angular εφαρμογή.

Η κοινή αποθήκευση διευκολύνει τον συντονισμό των αλλαγών, αλλά frontend και backend παραμένουν ξεχωριστές εφαρμογές και εκτελούνται ανεξάρτητα.

### 4.6.3 Frontend

Το frontend είναι υπεύθυνο για:

- παρουσίαση της διεπαφής (UI),
- πλοήγηση,
- φόρμες και τοπική επικύρωση,
- επικοινωνία με το API,
- διαχείριση της κατάστασης σύνδεσης,
- διεθνοποίηση,
- responsive προβολή,
- εμφάνιση χαρτών και διαδρομών.

Το frontend δεν θεωρείται έμπιστο επίπεδο για την ασφάλεια. Οι κρίσιμοι και τελικοί έλεγχοι γίνονται στο backend.

### 4.6.4 Backend ως αρθρωτός μονόλιθος

Το backend εκτελείται ως μία Spring Boot εφαρμογή, αλλά οργανώνεται σε λειτουργικές μονάδες, όπως user, vehicle, maintenance, fuel, accessory, part, document, reminder, route, dashboard, import, export, storage και security.

Η επιλογή αρθρωτού μονόλιθου επιτρέπει:

- απλούστερη εγκατάσταση,
- κοινή βάση δεδομένων,
- καθαρή εσωτερική οργάνωση,
- μικρότερη πολυπλοκότητα από μια αρχιτεκτονική micro service.

Μια τυπική λειτουργική μονάδα περιλαμβάνει controller, service, repository, entity, DTOs, validation και tests.

### 4.6.5 Βάση δεδομένων

Η PostgreSQL χρησιμοποιείται για χρήστες, οχήματα, λειτουργικές εγγραφές, tokens, metadata αρχείων και δεδομένα background εργασιών.

Η εξέλιξη του schema πραγματοποιείται με migrations, ώστε οι αλλαγές να είναι ελεγχόμενες και επαναλήψιμες.

### 4.6.6 Αποθήκευση αρχείων

Η διαχείριση αρχείων γίνεται μέσω κοινής διεπαφής αποθήκευσης. Έτσι οι λειτουργικές μονάδες δεν εξαρτώνται άμεσα από το αν το αρχείο αποθηκεύεται τοπικά ή σε object storage.

Τα αρχεία παραμένουν ιδιωτικά. Η λήψη τους πραγματοποιείται μέσω του backend μετά από έλεγχο ιδιοκτησίας.

### 4.6.7 Εξωτερικές υπηρεσίες

Η εφαρμογή μπορεί να επικοινωνεί με:

- Google Identity για σύνδεση,
- SMTP provider για email,
- OpenStreetMap για χάρτες,
- υπηρεσία συναλλαγματικών ισοτιμιών,
- object storage για αρχεία.

Η μη διαθεσιμότητα μιας προαιρετικής υπηρεσίας δεν πρέπει να διακόπτει τις υπόλοιπες λειτουργίες.

### 4.6.8 Τυπική ροή αιτήματος

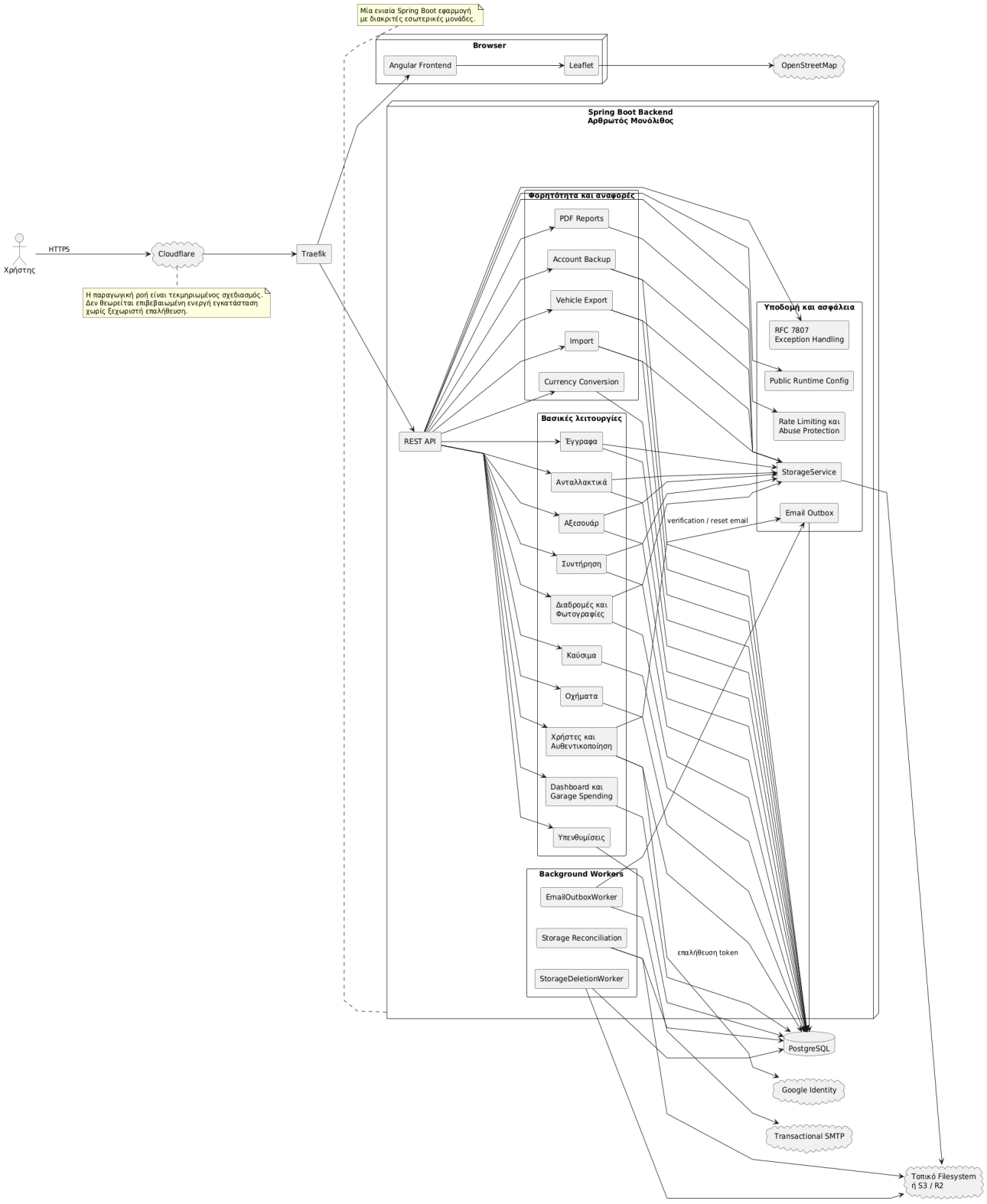
1. Ο χρήστης εκτελεί ενέργεια στο frontend.
2. Το frontend αποστέλλει HTTP αίτημα.
3. Το backend επαληθεύει την ταυτότητα και την ιδιοκτησία.
4. Το service εφαρμόζει τους επιχειρησιακούς κανόνες.
5. Το repository ή το storage εκτελεί την απαραίτητη ενέργεια.
6. Το backend επιστρέφει DTO ή συνεπές μήνυμα σφάλματος.
7. Το frontend ενημερώνει τη διεπαφή.

### 4.6.9 Επιλογές και περιορισμοί

Η αρχιτεκτονική επιλέχθηκε ώστε να είναι κατανοητή, συντηρήσιμη και επεκτάσιμη για το μέλλον.

Οι λεπτομέρειες υλοποίησης παρουσιάζονται στο Κεφάλαιο 6, ενώ η φυσική εγκατάσταση παρουσιάζεται στην Ενότητα 4.10.

**Σχήμα 4.2**: Συνολική αρχιτεκτονική του Moto-Vault



## 4.7 Σχεδιασμός δεδομένων

Το Moto-Vault χρησιμοποιεί σχεσιακό μοντέλο δεδομένων, επειδή οι πληροφορίες έχουν σαφείς σχέσεις και απαιτούν ελέγχους ακεραιότητας.

### 4.7.1 Βασικές αρχές

- Κάθε οντότητα διαθέτει μοναδικό αριθμητικό αναγνωριστικό.
- Κάθε όχημα ανήκει σε έναν χρήστη.
- Οι λειτουργικές εγγραφές συνδέονται με συγκεκριμένο όχημα.
- Οι σχέσεις εκφράζονται με ξένα κλειδιά.
- Τα πραγματικά αρχεία αποθηκεύονται εκτός βάσης.
- Στη βάση αποθηκεύονται μόνο metadata αρχείων.
- Η εξέλιξη του schema γίνεται μέσω migrations.
- Τα entities δεν εκτίθενται απευθείας στο API.

### 4.7.2 Χρήστης και όχημα

Η οντότητα User αποτελεί τη ρίζα ιδιοκτησίας. Περιλαμβάνει email, στοιχεία αυθεντικοποίησης, κατάσταση επαλήθευσης, πλάνο, προτιμήσεις, κύριο όχημα και χρονικές σημάνσεις.

Η οντότητα Vehicle περιλαμβάνει κατασκευαστή, μοντέλο, έτος, κυβισμό, αριθμό κυκλοφορίας, VIN, τρέχοντα χιλιόμετρα και κύρια φωτογραφία.

Ένας χρήστης μπορεί να διαθέτει πολλά οχήματα, αλλά κάθε όχημα ανήκει σε έναν μόνο χρήστη.

### 4.7.3 Συντήρηση και συνημμένα

Η οντότητα MaintenanceRecord περιλαμβάνει ημερομηνία, χιλιόμετρα, κατηγορία, τίτλο, περιγραφή, συνεργείο, κόστος και νόμισμα.

Μια εγγραφή συντήρησης μπορεί να έχει πολλά συνημμένα και να συνδέεται με ανταλλακτικά ή αξεσουάρ. Τα συνημμένα αποθηκεύουν μόνο metadata και storage path.

### 4.7.4 Ανεφοδιασμοί

Η οντότητα ανεφοδιασμού περιλαμβάνει ημερομηνία, χιλιόμετρα, ποσότητα καυσίμου, κόστος, τιμή ανά μονάδα και ένδειξη πλήρους ή μερικού ανεφοδιασμού.

Οι εγγραφές χρησιμοποιούνται για τους υπολογισμούς κατανάλωσης και κόστους.

### 4.7.5 Αξεσουάρ και ανταλλακτικά

Οι δύο οντότητες αποθηκεύουν πληροφορίες αγοράς, ημερομηνίες, κόστος, νόμισμα, σημειώσεις και προαιρετική σύνδεση με εγγραφή συντήρησης.

Μπορούν επίσης να έχουν σχετικά αρχεία ή φωτογραφίες.

### 4.7.6 Έγγραφα και υπενθυμίσεις

Η οντότητα εγγράφου συνδέεται με όχημα και περιλαμβάνει τύπο, τίτλο, ημερομηνίες, σημειώσεις και metadata αρχείου.

Η οντότητα υπενθύμισης περιλαμβάνει τίτλο, περιγραφή, ημερομηνία, χιλιομετρικό όριο, κατάσταση και ένδειξη ολοκλήρωσης.

### 4.7.7 Φωτογραφίες και διαδρομές

Οι φωτογραφίες οχήματος αποθηκεύουν metadata, σειρά προβολής και σχέση με το όχημα. Το όχημα διατηρεί αναφορά στην κύρια φωτογραφία.

Η οντότητα διαδρομής αποθηκεύει τίτλο, ημερομηνία, περιγραφή, storage path και βασικά στατιστικά που προκύπτουν από το GPX.

### 4.7.8 Λογαριασμός, tokens και background δεδομένα

Το μοντέλο περιλαμβάνει επίσης δεδομένα για:

- επαλήθευση email,
- ανάκτηση κωδικού,
- ανάκληση συνεδριών,
- email outbox,
- εργασίες διαγραφής αρχείων,
- εισαγωγές και backup.

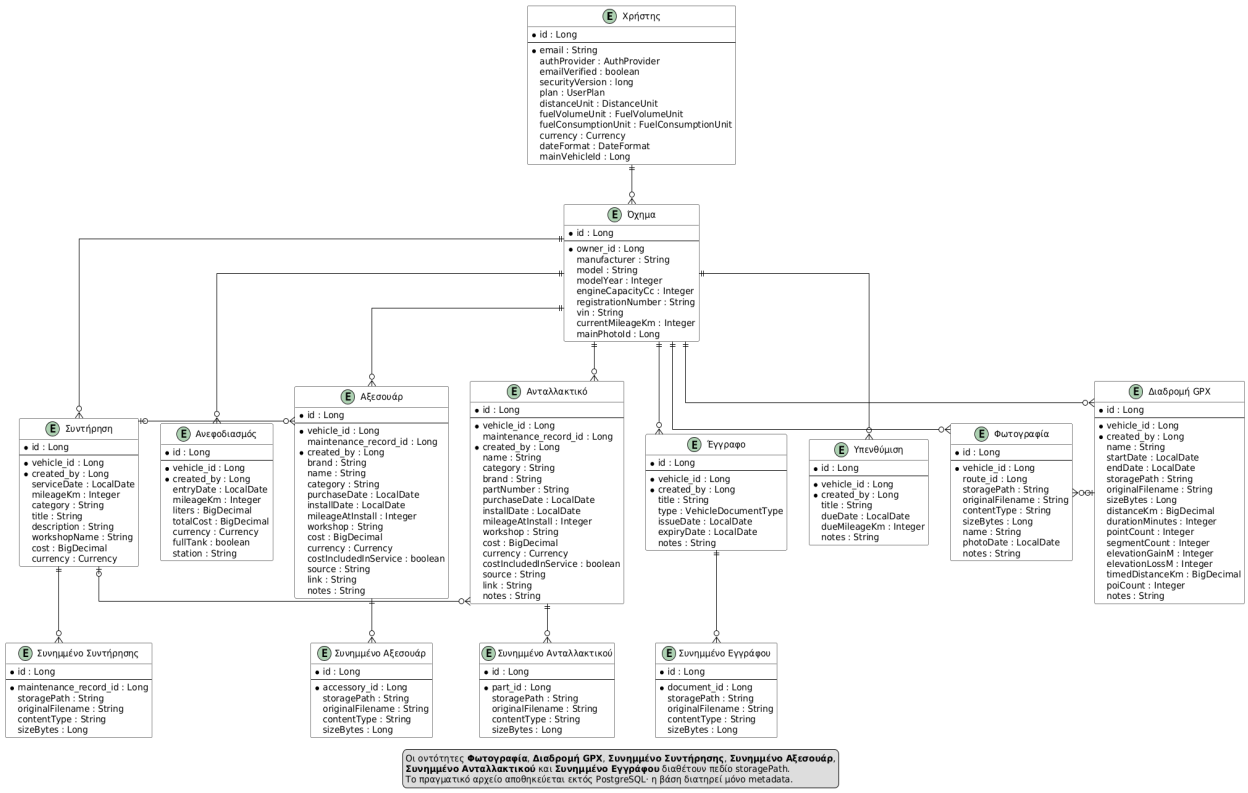
Οι οντότητες αυτές υποστηρίζουν τη λειτουργία του συστήματος, χωρίς να αποτελούν βασικές πληροφορίες μοτοσυκλέτας.

### 4.7.9 Ακεραιότητα και διαγραφές

Η βάση εφαρμόζει μοναδικότητα όπου χρειάζεται, μη μηδενικά πεδία και ξένα κλειδιά. Οι διαγραφές σχεδιάζονται ώστε να καθαρίζονται σωστά οι αντίστοιχες εγγραφές και αρχεία.

Η ασφάλεια δεν στηρίζεται στο να μαντεύει κανείς δύσκολα IDs. Στηρίζεται στο ότι το backend ελέγχει πάντα αν το αντικείμενο ανήκει πράγματι στον χρήστη που το ζητάει.

**Σχήμα 4.3**: Διάγραμμα οντοτήτων και σχέσεων του Moto-Vault




## 4.8 Σχεδιασμός REST API

Το REST API αποτελεί το σημείο επικοινωνίας ανάμεσα στο Angular frontend και στο Spring Boot backend.

### 4.8.1 Βασικές αρχές

Το REST API σχεδιάστηκε με βάση τις σύγχρονες τεχνικές προδιαγραφές:

**Δομή & Διαδρομές**: Χρησιμοποιεί πληθυντικό αριθμό για τις συλλογές (π.χ. /vehicles), nested routes για τα δεδομένα των οχημάτων και τις κατάλληλες HTTP μεθόδους (GET, POST, PUT, DELETE).

**Ασφάλεια και Έλεγχοι**: Η πρόσβαση προστατεύεται με JWT authentication, ενώ το backend εκτελεί πάντα ελέγχους ιδιοκτησίας και επικύρωση (validation) των δεδομένων.

**Διαχείριση Δεδομένων και Αρχείων**: Γίνεται χρήση DTOs, τα δομημένα δεδομένα ανταλλάσσονται σε μορφή JSON, τα αρχεία ανεβαίνουν με multipart requests και κατεβαίνουν μέσω streaming.

**Συνέπεια**: Όλα τα σφάλματα επιστρέφονται σε συγκεκριμένη μορφή.

### 4.8.2 Κύριες ομάδες endpoints

/api/auth
/api/users
/api/vehicles
/api/vehicles/{vehicleId}/maintenance-records
/api/vehicles/{vehicleId}/fuel-entries
/api/vehicles/{vehicleId}/accessories
/api/vehicles/{vehicleId}/parts
/api/vehicles/{vehicleId}/documents
/api/vehicles/{vehicleId}/reminders
/api/vehicles/{vehicleId}/photos
/api/vehicles/{vehicleId}/routes
/api/vehicles/{vehicleId}/dashboard
/api/vehicles/{vehicleId}/export
/api/import
/api/garage/spending
/api/exchange-rates
/api/public/config

Για το vehicleId, το backend ελέγχει ότι το όχημα ανήκει στον συνδεδεμένο χρήστη.

### 4.8.3 HTTP μέθοδοι

**Πίνακας 4.2:** Χρήση των βασικών HTTP μεθόδων στο REST API

| **Μέθοδος** | **Χρήση** |
|---|---|
| GET | Ανάκτηση λίστας, δεδομένου, αναφοράς ή αρχείου |
| POST | Δημιουργία, upload ή ειδική ενέργεια |
| PUT | Ενημέρωση ή αλλαγή κατάστασης |
| DELETE | Διαγραφή δεδομένου |

### 4.8.4 Αντιπροσωπευτικά endpoints

POST /api/auth/register
POST /api/auth/login
POST /api/auth/google
GET /api/auth/me

GET /api/vehicles
POST /api/vehicles
GET /api/vehicles/{id}
PUT /api/vehicles/{id}
DELETE /api/vehicles/{id}

GET /api/vehicles/{vehicleId}/maintenance-records
POST /api/vehicles/{vehicleId}/maintenance-records
PUT /api/vehicles/{vehicleId}/maintenance-records/{id}
DELETE /api/vehicles/{vehicleId}/maintenance-records/{id}

POST /api/vehicles/{vehicleId}/routes
GET /api/vehicles/{vehicleId}/routes/{id}/file

GET /api/users/me/account-backup
POST /api/users/me/account-backup

### 4.8.5 DTOs και validation

Τα entities δεν επιστρέφονται απευθείας. Τα DTOs ορίζουν τα δεδομένα που επιτρέπεται να εισέρχονται και να εξέρχονται από το API.

Η επικύρωση πραγματοποιείται στο backend ακόμη και όταν υπάρχει αντίστοιχος έλεγχος στη διεπαφή. Τα validation errors επιστρέφονται με σαφή πεδία και μηνύματα.

### 4.8.6 Αυθεντικοποίηση και ιδιοκτησία

Η αυθεντικοποίηση επιβεβαιώνει την ταυτότητα του χρήστη. Ο έλεγχος ιδιοκτησίας επιβεβαιώνει ότι ο χρήστης έχει δικαίωμα πρόσβασης στο συγκεκριμένο δεδομένο.

Η κατοχή έγκυρου token δεν αρκεί για πρόσβαση σε οποιοδήποτε αναγνωριστικό. Κάθε λειτουργία που αφορά όχημα περιορίζεται στον ιδιοκτήτη.

### 4.8.7 Αρχεία

Τα uploads χρησιμοποιούν `multipart/form-data`. Τα αρχεία αποθηκεύονται ιδιωτικά και επιστρέφονται μέσω streaming, αφού προηγηθεί έλεγχος ιδιοκτησίας.

Οι JSON αποκρίσεις περιλαμβάνουν μόνο metadata και όχι το δυαδικό περιεχόμενο.

### 4.8.8 Status codes και σφάλματα

Το API χρησιμοποιεί ενδεικτικά:

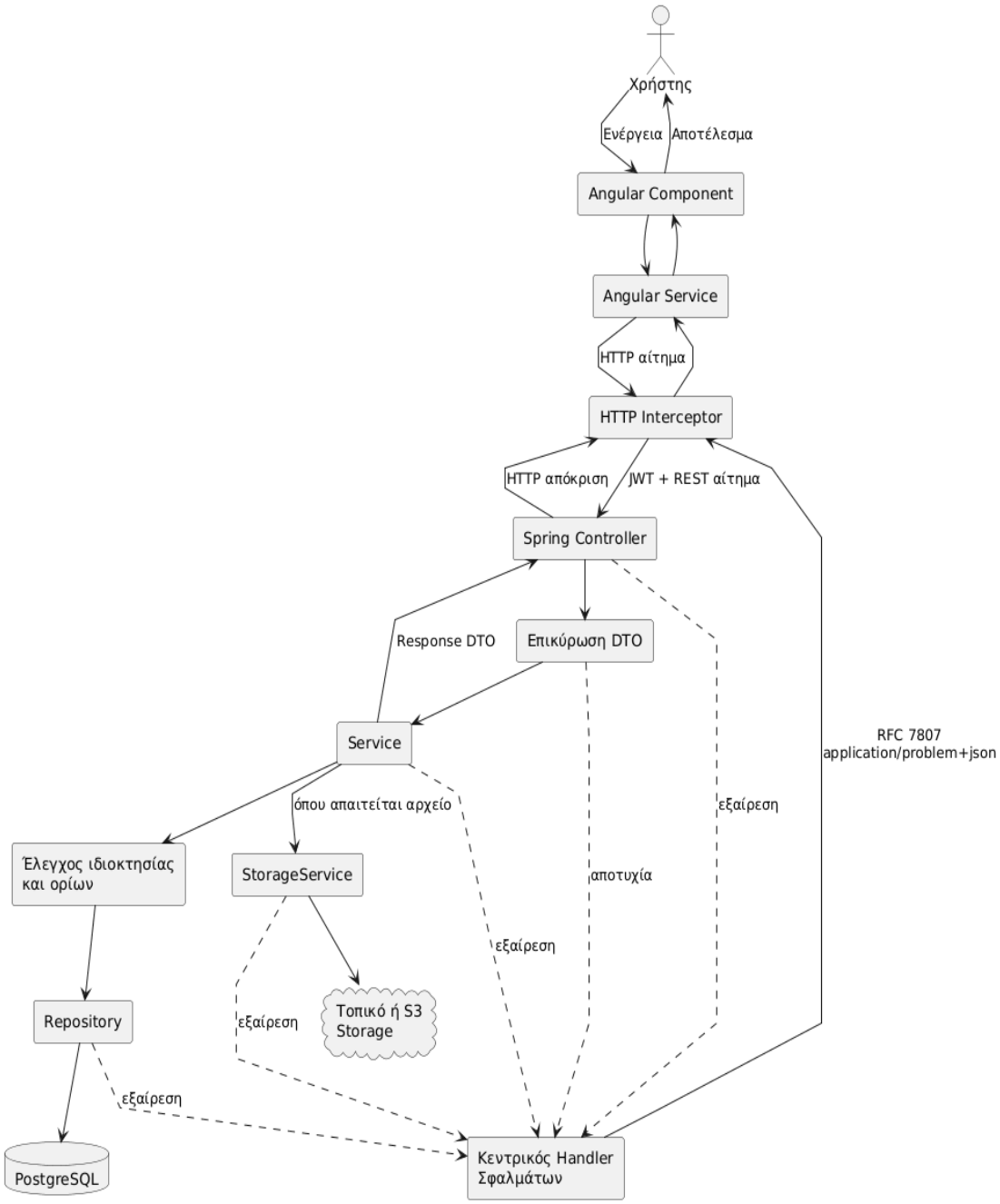
- 200 OK
- 201 Created
- 202 Accepted
- 204 No Content
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found
- 409 Conflict
- 413 Payload Too Large
- 500 Internal Server Error
- 503 Service Unavailable

Τα σφάλματα επιστρέφονται σε συνεπή μορφή RFC 7807, ώστε το frontend να μπορεί να τα παρουσιάζει με κοινό τρόπο.

### 4.8.9 Δημόσια endpoints

Ορισμένα endpoints, όπως το public configuration, παρέχουν μόνο πληροφορίες που είναι ασφαλές να εκτεθούν. Δεν επιστρέφουν credentials ή άλλες ευαίσθητες ρυθμίσεις.

**Σχήμα 4.4**: Ροή αιτήματος από το frontend προς το REST API



## 4.9 Σχεδιασμός διεπαφής χρήστη

Η σχεδίαση του UI βασίστηκε στο πρότυπο της πληροφοριακής αρχιτεκτονικής με επίκεντρο το αντικείμενο (object centric design) όπου το όχημα αποτελεί τη βασική μονάδα. Θέλουμε ο χρήστης να μπορεί να βρίσκει αμέσως την πληροφορία που χρειάζεται και να ολοκληρώνει κάθε ενέργεια χωρίς έξτρα βήματα.

### 4.9.1 Βασικοί στόχοι

- εύκολη πλοήγηση
- επιβεβαίωση "καταστροφικών" ενεργειών (διαγραφή λογαριασμού)
- responsive λειτουργία
- ελληνικό και αγγλικό περιβάλλον
- βασική προσβασιμότητα

### 4.9.2 Πλοήγηση με βάση το όχημα

Η βασική ροή είναι:

1. σύνδεση
2. επιλογή ή δημιουργία οχήματος
3. μετάβαση στη λεπτομερή προβολή
4. επιλογή λειτουργικής περιοχής
5. προβολή λίστας ή φόρμας
6. επιστροφή στο overview page

Οι κύριες περιοχές είναι dashboard, συντήρηση, καύσιμα, αξεσουάρ, ανταλλακτικά, έγγραφα, υπενθυμίσεις, φωτογραφίες και διαδρομές.

### 4.9.3 Δημόσιες οθόνες

Οι δημόσιες οθόνες περιλαμβάνουν αρχική σελίδα, εγγραφή, σύνδεση, επαλήθευση email και ανάκτηση κωδικού. Κάθε οθόνη εστιάζει σε μία βασική ενέργεια.

### 4.9.4 Γκαράζ και λεπτομέρειες οχήματος

Η λίστα οχημάτων αποτελεί το κύριο σημείο εισόδου μετά τη σύνδεση. Κάθε κάρτα μπορεί να εμφανίζει φωτογραφία, κατασκευαστή, μοντέλο, έτος, χιλιόμετρα και ένδειξη κύριου οχήματος.

Η λεπτομερής προβολή λειτουργεί ως κεντρικός κόμβος και παρέχει βασικά στοιχεία, συνοπτικές πληροφορίες και πλοήγηση προς τις επιμέρους λειτουργίες.

### 4.9.5 Dashboard

Το dashboard παρουσιάζει συνοπτικά στοιχεία όπως:

- συνολικό και πρόσφατο κόστος
- μέση κατανάλωση
- πρόσφατες εργασίες
- επερχόμενες υπενθυμίσεις
- έγγραφα που λήγουν
- πρόσφατη δραστηριότητα

Οι κάρτες λειτουργούν ως σύντομες περιλήψεις και οδηγούν στις αναλυτικές σελίδες.

### 4.9.6 Λίστες και κενές καταστάσεις
Οι λίστες ακολουθούν κοινή δομή με τίτλο, σύντομη περιγραφή, βασική ενέργεια, φίλτρα όπου χρειάζονται και ενέργειες επεξεργασίας ή διαγραφής.

Όταν δεν υπάρχουν δεδομένα, η κενή κατάσταση εξηγεί τι μπορεί να προσθέσει ο χρήστης και παρέχει άμεσο κουμπί δημιουργίας.

### 4.9.7 Φόρμες
Οι φόρμες χρησιμοποιούν:

- σαφή τίτλο
- λογική ομαδοποίηση πεδίων
- εμφανή υποχρεωτικά πεδία
- βοήθεια όπου χρειάζεται
- validation κοντά στο σχετικό πεδίο
- σαφή κουμπιά αποθήκευσης και ακύρωσης

Η ίδια δομή επαναχρησιμοποιείται σε διαφορετικές λειτουργικές περιοχές.

### 4.9.8 Ανάδραση και σφάλματα
Οι επιτυχημένες ενέργειες επιβεβαιώνονται με σύντομα μηνύματα. Τα σφάλματα εμφανίζονται με κατανοητή γλώσσα και χωρίς τεχνικές λεπτομέρειες.

Οι διαγραφές και άλλες μη αναστρέψιμες ενέργειες απαιτούν modal επιβεβαίωσης.

### 4.9.9 Αρχεία και πολυμέσα
Οι μεταφορτώσεις παρουσιάζουν υποστηριζόμενους τύπους, όρια μεγέθους και κατάσταση προόδου όπου εφαρμόζεται. Οι φωτογραφίες εμφανίζονται με προεπισκόπηση, ενώ τα έγγραφα και τα συνημμένα με όνομα, μέγεθος και δυνατότητα download.

### 4.9.10 Χάρτες και GPX
Η οθόνη routes παρουσιάζει τον χάρτη, τη γραμμή της διαδρομής και βασικά στατιστικά. Η βιβλιοθήκη χαρτών φορτώνεται μόνο όταν ο χρήστης ανοίξει σχετική σελίδα.

### 4.9.11 Responsive σχεδιασμός
Σε μικρές οθόνες οι πίνακες μπορούν να μετατρέπονται σε κάρτες, οι ενέργειες να ομαδοποιούνται και η πλοήγηση να προσαρμόζεται χωρίς να χάνονται λειτουργίες.

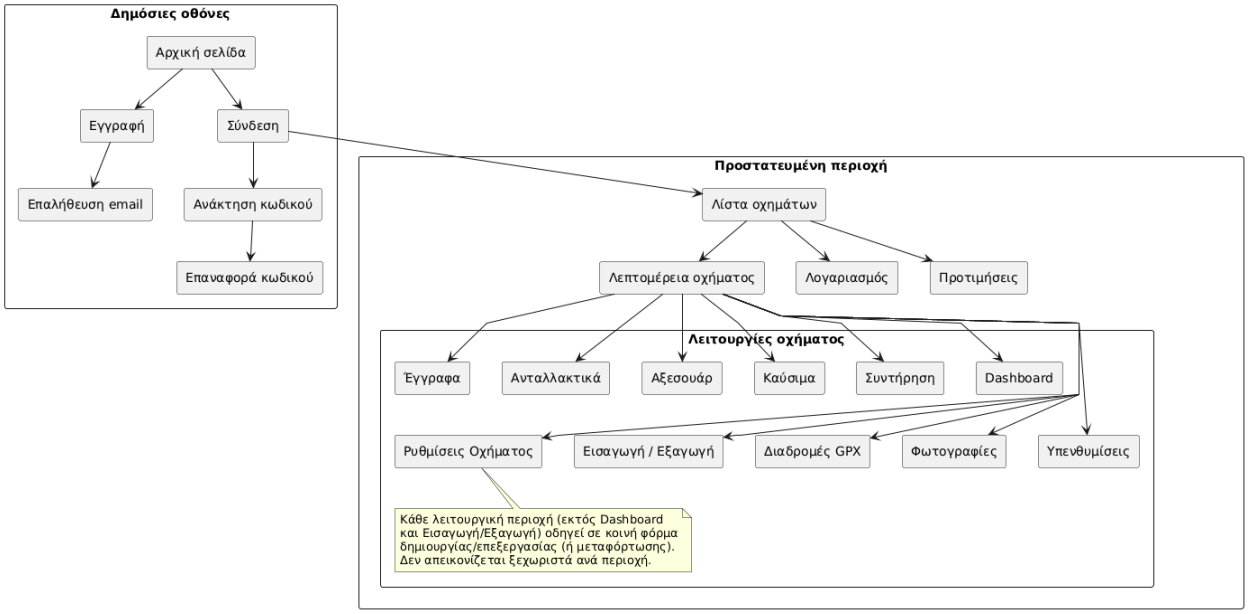
### 4.9.12 Διεθνοποίηση και προτιμήσεις
Όλα τα κείμενα χρησιμοποιούν translation keys. Οι μονάδες, το νόμισμα και οι ημερομηνίες εμφανίζονται σύμφωνα με τις προτιμήσεις του χρήστη.

### 4.9.13 Προσβασιμότητα
Χρησιμοποιούνται labels, σαφή focus states, περιγραφές εικονιδίων και κατανοητή δομή επικεφαλίδων.

### 4.9.14 Διάγραμμα πλοήγησης

Το διάγραμμα πλοήγησης αποτυπώνει τις δημόσιες οθόνες, το γκαράζ, τη λεπτομερή προβολή οχήματος και τις κύριες λειτουργικές περιοχές.

**Σχήμα 4.5**: Διάγραμμα πλοήγησης του Moto-Vault




## 4.10 Σχεδιασμός εγκατάστασης και διάγραμμα ανάπτυξης

Το deployment diagram παρουσιάζει τη φυσική κατανομή των βασικών τμημάτων του Moto-Vault σε περιβάλλον παραγωγής.

### 4.10.1 Βασικοί κόμβοι

Ο σχεδιασμός περιλαμβάνει:

- συσκευή χρήστη
- Cloudflare
- VPS
- reverse proxy
- frontend container
- backend container
- PostgreSQL
- migration job
- object storage
- SMTP provider
- Google Identity
- OpenStreetMap

### 4.10.2 Πρόσβαση χρήστη και Cloudflare

Ο χρήστης χρησιμοποιεί την εφαρμογή μέσω browser και HTTPS.

Η Cloudflare παρέχει DNS, TLS, reverse proxying, βασική προστασία από επιθέσεις και περιορισμό της άμεσης έκθεσης του origin.

### 4.10.3 VPS και containers

Το VPS φιλοξενεί το Docker περιβάλλον, το Dokploy, τον Traefik και τα containers της εφαρμογής.

Ο Traefik δρομολογεί:

- τα αιτήματα frontend προς το Angular container
- τα αιτήματα `/api` προς το Spring Boot backend

Το frontend container παρέχει τα στατικά αρχεία της Angular εφαρμογής. Το backend container εκτελεί το REST API, την επιχειρησιακή λογική και τις background εργασίες.

### 4.10.4 PostgreSQL και migrations

Η PostgreSQL δεν πρέπει να είναι δημόσια προσβάσιμη. Συνδέεται μόνο με το backend και με το migration job.

Τα migrations εκτελούνται από ξεχωριστό process πριν από την εκκίνηση της νέας έκδοσης του backed.

### 4.10.5 Object storage

Το S3 συμβατό object storage χρησιμοποιείται για φωτογραφίες, έγγραφα, συνημμένα και αρχεία GPX.

Τα αντικείμενα παραμένουν ιδιωτικά και το backend είναι υπεύθυνο για τον έλεγχο πρόσβασης και τη λήψη τους.

### 4.10.6 Email και εξωτερικές υπηρεσίες

Ο SMTP provider χρησιμοποιείται για email επαλήθευσης και ανάκτησης. Το Google Identity χρησιμοποιείται για σύνδεση μέσω Google και το OpenStreetMap για χαρτογραφικά δεδομένα.
Οι εξωτερικές υπηρεσίες δεν αποκτούν άμεση πρόσβαση στη βάση δεδομένων.

### 4.10.7 Ασφάλεια δικτύου

Ο σχεδιασμός προβλέπει:

- HTTPS
- firewall στο VPS
- SSH keys
- μη δημόσια PostgreSQL
- περιορισμένη έκθεση θυρών
- σωστή διαχείριση forwarded headers
- secrets μέσω environment variables
- περιορισμένα database credentials για το runtime

### 4.10.8 Ροή εγκατάστασης

1. Ανακτάται ή κατασκευάζεται η νέα έκδοση
2. Εκτελούνται τα migrations
3. Εκκινεί το backend
4. Εκκινεί ή ενημερώνεται το frontend
5. Ο reverse proxy ελέγχει την υγεία των υπηρεσιών
6. Η νέα έκδοση γίνεται διαθέσιμη μέσω του domain

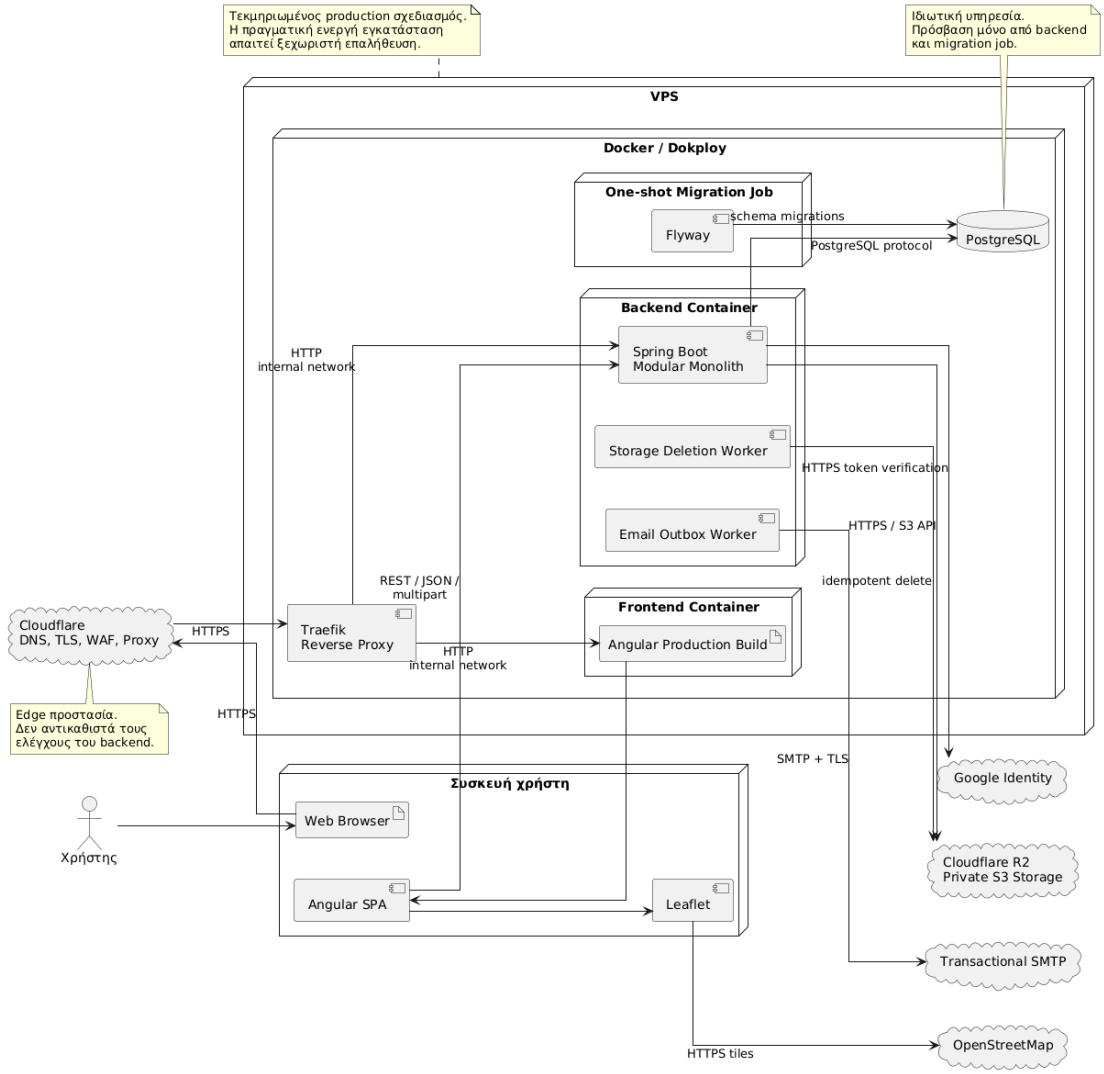
### 4.10.9 Αντίγραφα ασφαλείας και ανάκτηση

Ο σχεδιασμός πρέπει να καλύπτει ξεχωριστά:

- backup της PostgreSQL,
- διατήρηση object storage,
- προστασία environment configuration,
- δοκιμή διαδικασίας επαναφοράς.

Η λειτουργία backup λογαριασμού του χρήστη δεν αντικαθιστά τα υποδομικά αντίγραφα ασφαλείας.

**Σχήμα 4.6**: Διάγραμμα ανάπτυξης του Moto-Vault



# 5. Τεχνολογίες και εργαλεία ανάπτυξης

Το παρόν κεφάλαιο παρουσιάζει τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του Moto-Vault και αιτιολογεί την επιλογή τους.

Το Κεφάλαιο 4 επικεντρώθηκε στις απαιτήσεις και στον σχεδιασμό του συστήματος. Το Κεφάλαιο 6 περιγράφει τον τρόπο με τον οποίο οι τεχνολογίες εφαρμόστηκαν στον πηγαίο κώδικα.

Οι επιλογές εξετάζονται με βάση:

- την καταλληλότητά τους για μια διαδικτυακή εφαρμογή πλήρους στοίβας
- τη συντηρησιμότητα και τη δυνατότητα επέκτασης
- την ασφάλεια και την προστασία δεδομένων
- την υποστήριξη αυτοματοποιημένων δοκιμών
- τη διαχείριση σχεσιακών δεδομένων και ιδιωτικών αρχείων
- την επαναληψιμότητα του περιβάλλοντος ανάπτυξης
- τη δυνατότητα εγκατάστασης σε διαδικτυακή υποδομή

Το βασικό tech stack αποτελείται από Java 21, Spring Boot, Angular, TypeScript, PostgreSQL, Flyway, Docker, GitHub Actions, Leaflet και OpenStreetMap. Συμπληρώνεται από εργαλεία ανάπτυξης, δοκιμών, αποθήκευσης αρχείων και διαχείρισης της υποδομής.

## 5.1 Java και Spring Boot

Το backend του Moto-Vault αναπτύχθηκε με Java 21 και Spring Boot 3.5.16. Ο συνδυασμός αυτός επιλέχθηκε επειδή είναι σταθερό και Long Term Support (LTS) tech stack για την ανάπτυξη REST API, πρόσβαση σε σχεσιακή βάση, ασφάλεια, validation και αυτοματοποιημένες δοκιμές.

### 5.1.1 Java 21

Η Java 21 προσφέρει στατική τυποποίηση, ώριμα εργαλεία ανάπτυξης και ισχυρή υποστήριξη από το οικοσύστημα Spring.

Η στατική τυποποίηση βοηθά στη μοντελοποίηση των οντοτήτων, των DTOs και των επιχειρησιακών κανόνων. Η Java διαθέτει στάνταρ εργαλεία build, testing, debugging και παρακολούθησης.

Το Maven Enforcer απαιτεί Java 21 ή νεότερη, ώστε το τοπικό περιβάλλον και το CI να χρησιμοποιούν συμβατή έκδοση.

### 5.1.2 Spring Boot

Το Spring Boot αποτελεί το βασικό framework του backend. Παρέχει ίσως τον πιο διάσημο τρόπο διαχείρισης configuration, dependency injection, web endpoints, persistence, validation, email και tests.
Χρησιμοποιούνται κυρίως τα ακόλουθα Spring modules:

- Spring Web για το REST API,
- Spring Data JPA για πρόσβαση στην PostgreSQL,
- Spring Security για αυθεντικοποίηση και προστασία endpoints,
- Bean Validation για έλεγχο των αιτημάτων,
- Spring Mail για σύνδεση με υπηρεσία SMTP.

Η χρήση ολοκληρωμένου framework μειώνει την ανάγκη χειροκίνητης σύνδεσης ανεξάρτητων βιβλιοθηκών και προσφέρει απίστευη ευκολία σε όλο το backend.

### 5.1.3 Persistence και ασφάλεια

Το Spring Data JPA και το Hibernate χρησιμοποιούνται για τη χαρτογράφηση των οντοτήτων και την πρόσβαση στη βάση. Το Spring Security υποστηρίζει την αυθεντικοποίηση με JWT, τη σύνδεση μέσω Google και την προστασία των endpoints.

Οι ακριβείς ροές αυθεντικοποίησης, οι έλεγχοι ιδιοκτησίας και η οργάνωση των repositories παρουσιάζονται στο Κεφάλαιο 6.

### 5.1.4 Βοηθητικές βιβλιοθήκες

Το backend χρησιμοποιεί επίσης:

- JJWT για δημιουργία και επαλήθευση JWT,
- Google API Client για server-side επαλήθευση Google ID tokens,
- AWS SDK για S3 compatible storage,
- OpenPDF για δημιουργία PDF,
- Apache PDFBox για έλεγχο παραγόμενων PDF στα tests,
- Jackson για JSON,
- Lombok για περιορισμό επαναλαμβανόμενου κώδικα.

Οι βιβλιοθήκες αυτές καλύπτουν συγκεκριμένες ανάγκες χωρίς να αλλάζουν τη βασική αρχιτεκτονική του Spring Boot backend.

### 5.1.5 Αιτιολόγηση επιλογής

Η Java και το Spring Boot ήταν κατάλληλες επιλογές επειδή:

- υποστηρίζουν άψογα backend,
- συνεργάζονται άμεσα με PostgreSQL και Flyway,
- παρέχουν σταθερούς μηχανισμούς ασφάλειας,
- διαθέτουν ισχυρή υποστήριξη,
- επιτρέπουν την ανάπτυξη monolith χωρίς την πολυπλοκότητα microservice.

## 5.2 Angular και TypeScript

Το frontend του Moto-Vault αναπτύχθηκε με Angular 21.2.18 και TypeScript 5.9.2.

### 5.2.1 Angular

Το Angular επιλέχθηκε επειδή προσφέρει ενιαίο framework για:

- routing,
- dependency injection,
- HTTP επικοινωνία,
- reactive forms,
- standalone components,
- reactive state,
- lazy loading,
- testing.

Επειδή η εφαρμογή έχει πολλές λειτουργίες και σύνθετες φόρμες, η χρήση ενός πλήρους framework μας βοηθάει να κρατήσουμε μια καθαρή, ενιαία δομή σε όλο τον κώδικα.

### 5.2.2 TypeScript

Η TypeScript προσθέτει τύπους στη JavaScript (στατική τυποποίηση) και επιτρέπει τη χρήση interfaces και type aliases.

Στο Moto-Vault χρησιμοποιείται για τη μοντελοποίηση των DTOs, των αποκρίσεων API, της κατάστασης των components και των κοινών utilities. Η τυποποίηση μειώνει τις ασυνέπειες ανάμεσα στα services και στα components και βελτιώνει την υποστήριξη του IDE.

### 5.2.3 Βασικοί μηχανισμοί του frontend

Η εφαρμογή χρησιμοποιεί:

- standalone components,
- Angular Router,
- HttpClient και interceptors,
- reactive forms,
- Angular signals,
- RxJS,
- zoneless λειτουργία.

Τα signals διαχειρίζονται το reactive state και υπολογίζουν αυτόματα τιμές που εξαρτώνται από αυτό, ενώ η RxJS χρησιμοποιείται για ασύγχρονες ροές, HTTP requests και route streams.

Οι λεπτομέρειες της οργάνωσης σε `core` και `features`, του vehicle shell και των frontend services παρουσιάζονται στην Ενότητα 6.3.

### 5.2.4 Διεθνοποίηση και εμφάνιση

Η ngx-translate 18 χρησιμοποιείται για ελληνικό και αγγλικό περιβάλλον. Τα κείμενα αποθηκεύονται σε translation files και δεν ενσωματώνονται απευθείας στα components.

Η μορφοποίηση βασίζεται σε SCSS και το UI προσαρμόζεται σωστά σε desktop, tablet και κινητό browser. Ο σχεδιασμός λαμβάνει επίσης υπόψη βασικές αρχές προσβασιμότητας, όπως labels, keyboard navigation και focus states.

### 5.2.5 Δοκιμές και διαχείριση dependencies

Το Vitest 4.1.10 χρησιμοποιείται ως test runner του frontend. Το npm διαχειρίζεται τα dependencies, ενώ το `package-lock.json` και το `npm ci` επιτρέπουν επαναλήψιμη εγκατάσταση στο CI.

### 5.2.6 Αιτιολόγηση επιλογής

Η Angular και η TypeScript ήταν κατάλληλες για το Moto-Vault επειδή υποστηρίζουν σύνθετες φόρμες, οργανωμένο routing, reactive διεπαφή, ισχυρή τυποποίηση, διεθνοποίηση(γλώσσες και ημερομηνίες) και αυτοματοποιημένες δοκιμές.

## 5.3 PostgreSQL και Flyway

Η PostgreSQL χρησιμοποιείται ως κύρια σχεσιακή βάση δεδομένων, ενώ το Flyway διαχειρίζεται την εξέλιξη του schema.

### 5.3.1 PostgreSQL

Η PostgreSQL επιλέχθηκε επειδή οι πληροφορίες του Moto-Vault έχουν συγκεκριμένες σχέσεις ιδιοκτησίας και εξάρτησης. Η βάση παρέχει συναλλαγές, foreign keys, constraints, indexes και δυνατότητα συγκεντρωτικών ερωτημάτων.

Αποθηκεύει χρήστες, οχήματα, λειτουργικές εγγραφές, tokens, metadata αρχείων και δεδομένα που υποστηρίζουν background εργασίες.

Στο development περιβάλλον χρησιμοποιείται PostgreSQL 16 μέσω Docker Compose, ώστε όλοι οι κύκλοι ανάπτυξης να βασίζονται στην ίδια έκδοση.

### 5.3.2 Σχεσιακή ακεραιότητα

Η ακεραιότητα υποστηρίζεται με:

- primary και foreign keys,
- unique constraints,
- check constraints,
- indexes,
- συναλλαγές.

Οι περιορισμοί (constraints) της PostgreSQL λειτουργούν σαν δεύτερη δικλείδα ασφαλείας, συμπληρώνοντας το validation και τους κανόνες που τρέχουν στο backend.

Οι κύριες οντότητες χρησιμοποιούν αριθμητικά αναγνωριστικά τύπου `Long`, τα οποία αντιστοιχούν σε `BIGINT`.

### 5.3.3 Flyway

Κάθε αλλαγή του schema αποθηκεύεται ως versioned migration. Με αυτόν τον τρόπο η δομή της βάσης μπορεί να αναπαραχθεί σε διαφορετικά περιβάλλοντα και να ελεγχθεί μαζί με τον κώδικα.

Το Flyway επιτρέπει:

- σταθερή σειρά αλλαγών,
- κοινή κατάσταση schema,
- code review των μεταβολών,
- αποφυγή ανεξέλεγκτου automatic update από το Hibernate.

Στο development τα migrations μπορούν να εκτελούνται κατά την εκκίνηση. Στον σχεδιασμό προβλέπεται ελεγχόμενη εκτέλεσή τους πριν από την εκκίνηση της νέας έκδοσης.

### 5.3.4 Αιτιολόγηση επιλογής

Ο συνδυασμός PostgreSQL και Flyway καλύπτει τις ανάγκες σχεσιακής ακεραιότητας, συναλλαγών και ελεγχόμενης εξέλιξης της βάσης. Η πραγματική χαρτογράφηση των entities, τα repositories και τα migrations παρουσιάζονται στην Ενότητα 6.5.

## 5.4 Τεχνολογίες αποθήκευσης αρχείων

Το Moto-Vault αποθηκεύει το πραγματικό περιεχόμενο των αρχείων εκτός PostgreSQL. Η βάση διατηρεί μόνο metadata και τις σχέσεις με τις αντίστοιχες οντότητες.

### 5.4.1 Αφαίρεση αποθήκευσης

Η κοινή διεπαφή `StorageService` απομονώνει την επιχειρησιακή λογική από τον συγκεκριμένο provider.

Υποστηρίζονται δύο βασικές προσεγγίσεις:

- τοπικό filesystem
- S3-compatible object storage

Η επιλογή γίνεται μέσω configuration και επιτρέπει διαφορετική λύση ανά περιβάλλον.

### 5.4.2 Τοπικό filesystem

Το local storage βολεύει πολύ στο development και το debugging γιατί στήνεται εύκολα. Σε περιβάλλον με containers όμως, θέλει οπωσδήποτε persistent volume για να μη χάνονται τα αρχεία, και δεν δίνεται η ευελιξία που έχει ένα εξωτερικό object storage.

### 5.4.3 S3 compatible storage

Η S3 υλοποίηση χρησιμοποιεί το AWS SDK for Java και μπορεί να συνεργαστεί με υπηρεσίες που υποστηρίζουν το ίδιο API.

Στο development χρησιμοποιείται MinIO, ενώ για το περιβάλλον παραγωγής (production) ένα ιδιωτικό bucket στο Cloudflare R2. Τα αρχεία δεν είναι προσβάσιμα μέσω δημόσιων URLs, αλλά είναι αποκλειστικά διαθέσιμα μέσω του backend.

### 5.4.4 Διαχωρισμός δεδομένων και αρχείων

Ο διαχωρισμός προσφέρει:

- μικρότερη επιβάρυνση της σχεσιακής βάσης
- δυνατότητα αλλαγής storage provider
- καλύτερη διαχείριση μεγάλων αρχείων
- ανεξάρτητη στρατηγική backup

Αυτό δημιουργεί την ανάγκη συντονισμού ανάμεσα στη βάση και στο storage, επειδή δεν συμμετέχουν στην ίδια ACID συναλλαγή.

Οι μηχανισμοί validation, rollback, durable deletion και reconciliation παρουσιάζονται αναλυτικά στις Ενότητες 6.6 και 6.9.

## 5.5 Εργαλεία ανάπτυξης και δοκιμών

Η ανάπτυξη του Moto-Vault υποστηρίζεται από εργαλεία version control, build, testing και debugging.

### 5.5.1 Περιβάλλον ανάπτυξης και version control

Το IntelliJ IDEA χρησιμοποιείται για την ανάπτυξη του backend, τη διαχείριση Maven, το debugging και την επιθεώρηση της βάσης.

Το Git χρησιμοποιείται για version control και το GitHub για τη φιλοξενία του repository, τα branches, τα pull requests και το code review.

Η δομή monorepo επιτρέπει σε μία αλλαγή να περιλαμβάνει συντονισμένες τροποποιήσεις backend, frontend και configuration.

### 5.5.2 Build και dependency management

Το Maven διαχειρίζεται τα backend dependencies, το compilation, τα tests και το packaging.

Το npm διαχειρίζεται τα frontend dependencies, τα scripts ανάπτυξης, τις δοκιμές και το production build.

Τα lock files και τα καθαρά CI περιβάλλοντα εξασφαλίζουν ότι το build της εφαρμογής είναι πάντα το ίδιο και παίζει παντού χωρίς προβλημάτα.

### 5.5.3 Development υπηρεσίες

Το Docker Compose χρησιμοποιείται για την εκκίνηση μερών του development περιβάλλοντος, όπως PostgreSQL, MinIO και Mailpit.

Με τη χρήση των containers εξαφανίζονται οι ασυμβατότητες ανάμεσα σε διαφορετικά dev συστήματα.

### 5.5.4 Εργαλεία δοκιμών

Το backend χρησιμοποιεί:

- JUnit 5
- Mockito
- Spring Boot Test
- Spring Security Test
- JaCoCo
- Apache PDFBox για έλεγχο PDF

Το frontend χρησιμοποιεί Vitest και Angular testing utilities.

Η ακριβής οργάνωση και κάλυψη των tests παρουσιάζεται στην Ενότητα 6.10.

### 5.5.5 Debugging και logging

Η διάγνωση προβλημάτων βασίζεται σε breakpoints, logs, browser developer tools, network inspection, database tools, MinIO και Mailpit.

Το backend χρησιμοποιεί SLF4J μέσω του logging stack του Spring Boot. Δεν καταγράφονται passwords, JWTs, action tokens ή access keys.

## 5.6 Συνεχής ολοκλήρωση και έλεγχοι ποιότητας

Το repository χρησιμοποιεί GitHub Actions για αυτοματοποιημένους ελέγχους backend, frontend και dependencies.

### 5.6.1 Backend CI

Το backend workflow ρυθμίζει Java 21 και εκτελεί το Maven verification lifecycle. Με αυτόν τον τρόπο ελέγχονται το compilation, τα tests και ο κανόνας κάλυψης τους JaCoCo.

Η κάλυψη αυτή δεν αποτελεί δείκτη ποιότητας, αλλά βοηθά να αποφεύγεται η σταδιακή απώλεια tests.

### 5.6.2 Frontend CI

Το frontend workflow ρυθμίζει Node.js 22 και εκτελεί:

1. npm ci (εντολή)
2. Vitest tests
3. Angular production build

Έτσι επιβεβαιώνεται ότι η εφαρμογή μπορεί να εγκατασταθεί και να κατασκευαστεί σε καθαρό περιβάλλον.

### 5.6.3 Security checks

Ξεχωριστό workflow εκτελεί ελέγχους για γνωστές ευπάθειες dependencies με OSV Scanner.

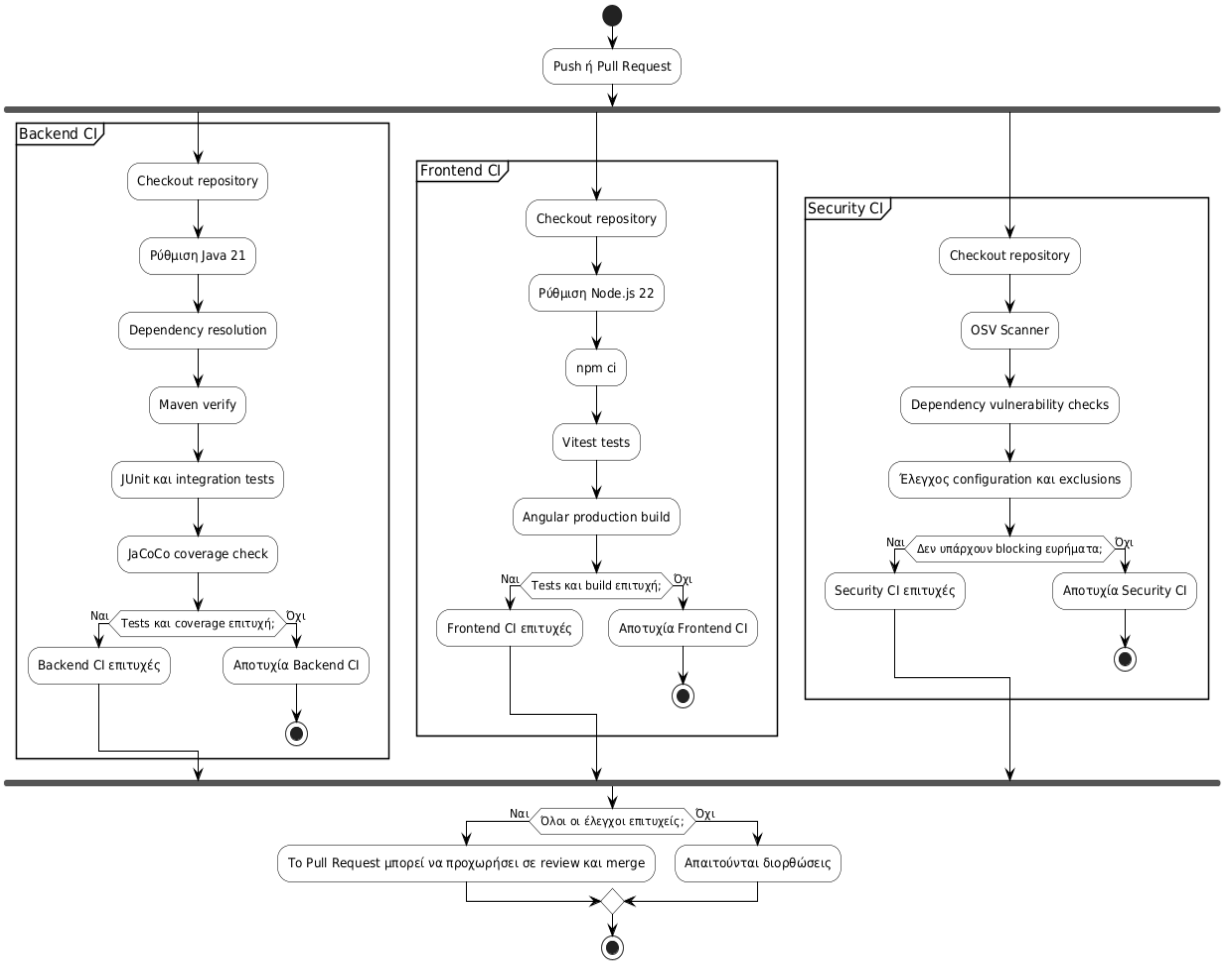
Το Dependabot δημιουργεί αυτόματες ενημερώσεις μέσω Pull Request (PR) για Maven, npm και GitHub Actions dependencies. Κάθε αλλαγή πρέπει να περάσει ελέγχους και review.

### 5.6.4 Pull request flow

Ένα pull request μπορεί να γίνει review και merge όταν οι απαιτούμενοι έλεγχοι έχουν ολοκληρωθεί επιτυχώς. Η επιτυχία των workflows δεν εκτελεί αυτόματα το merge, αλλά επιβεβαιώνει ότι ο κώδικας περνά τις βασικές τεχνικές προϋποθέσεις.

Το repository καλύπτει πλήρως το κομμάτι του CI. Επειδή όμως το deployment στο production δεν γίνεται εντελώς αυτόματα, μιλάμε για Continuous Integration με μια καθορισμένη διαδικασία deployment, και όχι για πλήρως αυτοματοποιημένο CD.

**Σχήμα 5.1**: Ροή συνεχούς ολοκλήρωσης και ελέγχων ποιότητας




## 5.7 Υποδομή και εργαλεία εγκατάστασης

Η υποδομή του MotoVault βασίζεται σε VPS και containers. Σε αυτή την ενότητα αναλύονται τα εργαλεία που χρησιμοποιήθηκαν και το σκεπτικό πίσω από κάθε επιλογή, ενώ η αρχιτεκτονική των κόμβων έχει παρουσιαστεί ήδη στην Ενότητα 4.10.

### 5.7.1 OVH VPS

Το OVH VPS επιλέχθηκε ως υποδομή επειδή παρέχει:

- έλεγχο του server,
- σταθερό κόστος,
- δυνατότητα εκτέλεσης Docker workloads,
- επαρκείς πόρους για την αρχική χρήση,
- ευρωπαϊκή υποδομή.

Η λύση απαιτεί περισσότερη διαχείριση από μια πλήρως managed πλατφόρμα και δεν πρέπει να παρουσιάζεται ως high availability χωρίς πρόσθετη υποδομή.

### 5.7.2 Docker

Με το Docker έχουμε ένα σταθερό runtime που παίζει παντού το ίδιο, χωρίς να μπερδεύονται οι εξαρτήσεις.

Αυτό μας επιτρέπει να σπάσουμε την εφαρμογή σε αυτόνομα containers για το frontend, το backend και τις υπόλοιπες υπηρεσίες.

### 5.7.3 Dokploy και Traefik

Το Dokploy χρησιμοποιείται για τη διαχείριση builds, environment variables, domains, logs, restart policies και health checks.

Ο Traefik λειτουργεί ως reverse proxy και δρομολογεί την κίνηση προς τα κατάλληλα containers.

### 5.7.4 Cloudflare

Η Cloudflare παρέχει DNS, TLS, proxying και edge protection. Οι λειτουργίες αυτές συμπληρώνουν την ασφάλεια του backend και δεν αντικαθιστούν τους ελέγχους της εφαρμογής.

### 5.7.5 R2 και SMTP

Για την αποθήκευση αρχείων (object storage) στο production επιλέχθηκε το Cloudflare R2. Αντίστοιχα, για την αποστολή των transactional emails απαιτείται ένας πραγματικός SMTP provider, καθώς το Mailpit χρησιμοποιείται αποκλειστικά στο τοπικό περιβάλλον ανάπτυξης.

### 5.7.6 Trade-offs

Αυτή η υποδομή δίνει μια πολύ καλή ισορροπία ανάμεσα σε κόστος, έλεγχο και δυνατότητα επέκτασης (scalability). Από την άλλη, απαιτεί τακτική διαχείριση του server, δηλαδή ενημερώσεις, monitoring και backups.

Η ακριβής εικόνα του production περιβάλλοντος βασίζεται αποκλειστικά στα στοιχεία που έχουν επιβεβαιωθεί κατά την τελική παράδοση του συστήματος.

## 5.8 Leaflet και OpenStreetMap

Για την απεικόνιση των GPX διαδρομών στον χάρτη χρησιμοποιούμε τη βιβλιοθήκη Leaflet (v1.9.4) με δωρεάν map tiles από το OpenStreetMap.

### 5.8.1 Leaflet

Η Leaflet είναι ελαφριά βιβλιοθήκη JavaScript για διαδραστικούς χάρτες.

Στο Moto-Vault χρησιμοποιείται για:

- προβολή χάρτη
- σχεδίαση polylines
- markers
- zoom και pan
- προσαρμογή του viewport
- responsive εμφάνιση

Η βιβλιοθήκη ενσωματώνεται στο Angular frontend και δεν αποτελεί εξωτερική υπηρεσία.

### 5.8.2 OpenStreetMap
Το OpenStreetMap παρέχει τα χαρτογραφικά tiles που φορτώνει ο browser. Η εφαρμογή εμφανίζει την απαιτούμενη attribution και πρέπει να σέβεται την πολιτική χρήσης της δημόσιας υπηρεσίας.

Το OpenStreetMap δεν αποκτά πρόσβαση στη βάση δεδομένων ή στα ιδιωτικά GPX αρχεία. Λαμβάνει μόνο τα αιτήματα που απαιτούνται για τη φόρτωση των tiles.

### 5.8.3 Κατανομή επεξεργασίας
Το backend αναλύει το GPX και υπολογίζει τα σχετικά metadata. Το frontend χρησιμοποιεί τα επεξεργασμένα σημεία για την οπτική παρουσίαση της διαδρομής.

Ο διαχωρισμός αυτός διατηρεί τους υπολογισμούς στο backend και το rendering στον browser.

### 5.8.4 Περιορισμοί
Η εφαρμογή εξαρτάται από εξωτερική υπηρεσία tiles. Σε περίπτωση σημαντικής αύξησης χρήσης μπορεί να απαιτηθεί διαφορετικός provider ή ξεχωριστή tile υποδομή.

Η πραγματική επεξεργασία GPX και η ενσωμάτωση του χάρτη παρουσιάζονται στην Ενότητα 6.7.

## 5.9 Συνολική αποτίμηση τεχνολογικών επιλογών
Οι τεχνολογικές επιλογές του MotoVault είναι καλά δεμένες μεταξύ τους, δημιουργώντας ένα ολοκληρωμένο stack για όλη την εφαρμογή.

Στο backend, η Java και το Spring Boot προσφέρουν μια στιβαρή δομή για τη λογική της εφαρμογής, την ασφάλεια, τη διαχείριση των δεδομένων και τα tests. Στο frontend, η Angular με την TypeScript αναλαμβάνουν τις σύνθετες φόρμες, το routing, το reactive state και τη διεθνοποίηση (i18n). Η PostgreSQL εξασφαλίζει τη σχεσιακή ακεραιότητα των δεδομένων, ενώ το Flyway αναλαμβάνει την ασφαλή και σταδιακή εξέλιξη της βάσης (database migrations).

Η αρχιτεκτονική αποθήκευσης επιτρέπει τη χρήση είτε τοπικού συστήματος αρχείων είτε S3 compatible υπηρεσίας. Το Docker και το Dokploy απλοποιούν το deployment, ενώ στο περιβάλλον παραγωγής (production) η Cloudflare και το R2 αναλαμβάνουν τις edge λειτουργίες και το object storage. Μέσω του GitHub Actions (CI/CD) γίνονται αυτόματοι και επαναλήψιμοι έλεγχοι για το backend, το frontend και τα dependencies σε βιβλιοθήκες του έργου.

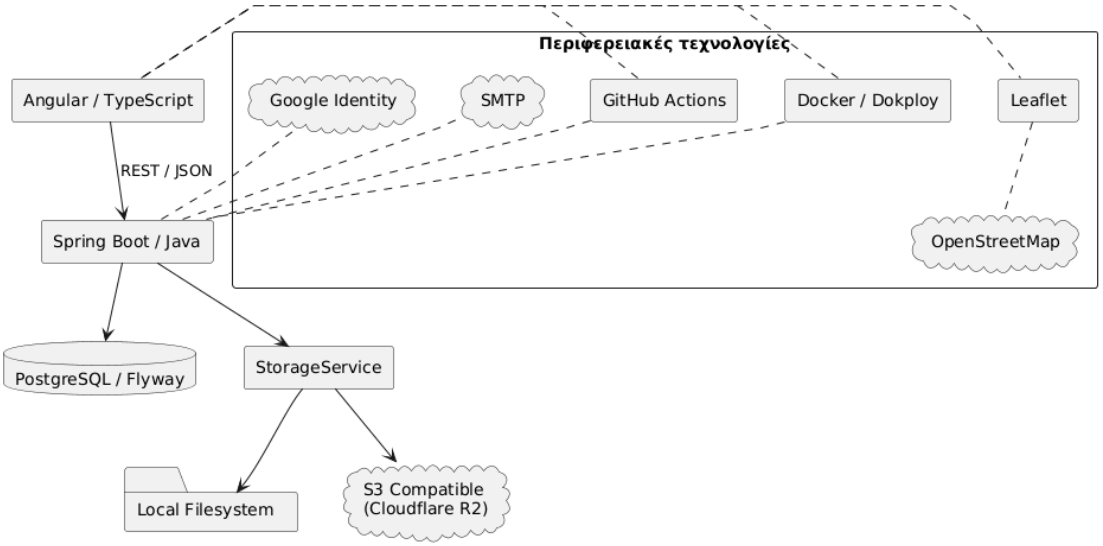
Οι κύριες υποχωρήσεις (trade offs) που χρειάστηκε να γίνουν είναι οι εξής:

- τα ολοκληρωμένα frameworks αυξάνουν την αρχική πολυπλοκότητα
- το VPS προσφέρει έλεγχο αλλά απαιτεί διαχείριση
- ο διαχωρισμός της βάσης δεδομένων από το storage απαιτεί έξτρα μηχανισμούς για τη διατήρηση της συνέπειας (consistency) των δεδομένων
- ο αρθρωτός μονόλιθος (modular monolith) κάνει το deployment πολύ πιο απλό αλλά δεν επιτρέπει την ανεξάρτητη κλιμάκωση (scaling) κάθε επιμέρους λειτουργίας.

Οι αποφάσεις κρίθηκαν σωστές επειδή η εφαρμογή αναπτύχθηκε από ένα άτομο, σε περιορισμένο χρόνο, ασφαλή διαχείριση αρχείων και δυνατότητα μελλοντικής επέκτασης.

Οι λύσεις αυτές δεν παρουσιάζονται ως η μοναδική ή η απόλυτα ιδανική επιλογή, αλλά ως ένας τεκμηριωμένος συνδυασμός που καλύπτει πλήρως τις απαιτήσεις και τους περιορισμούς του έργου.

**Σχήμα 5.2**: Τεχνολογικό οικοσύστημα του Moto-Vault



# 6. Υλοποίηση

## 6.1 Οργάνωση και δομή του πηγαίου κώδικα

Το Moto-Vault οργανώνεται ως monorepo, στο οποίο συνυπάρχουν backend, frontend, αρχεία υποδομής και workflows συνεχούς ολοκλήρωσης. Ο πηγαίος κώδικας διατηρείται σε ιδιωτικό GitHub repository στο https://github.com/IoannisArmamentos/. Πρόσβαση μπορεί να δωθεί, εφόσον απαιτηθεί.
Η βασική δομή είναι:

```
MotoVault/
├── .github/
├── backend/
├── frontend/
├── docker-compose.yml
├── osv-scanner.toml
├── README.md
└── .gitignore
```

Η επιλογή του monorepo μας επιτρέπει να κάνουμε ταυτόχρονες αλλαγές στο API και το frontend, διατηρώντας συγκεντρωμένη την «τεκμηρίωση» και ενιαία τη διαχείριση όλων των dev υπηρεσιών.

### 6.1.1 Δομή backend

Ο βασικός κώδικας βρίσκεται στη διαδρομή:

backend/src/main/java/com/motovault

Η εφαρμογή ξεκινά από την MotoVaultApplication.

Ο κώδικας οργανώνεται ανά λειτουργική περιοχή:

```
com.motovault
├── accessory
├── accountbackup
├── common
├── config
├── currency
├── dashboard
├── dataimport
├── document
├── email
├── export
├── fuel
├── maintenance
├── part
├── reminder
├── route
├── security
├── storage
├── user
└── vehicle
```

Η οργάνωση package by feature συγκεντρώνει στο ίδιο package τα στοιχεία μιας λειτουργίας, όπως entity, repository, service, controller, DTOs και tests.

Οι βασικές περιοχές καλύπτουν λογαριασμούς, οχήματα, συντήρηση, καύσιμα, αξεσουάρ, ανταλλακτικά, έγγραφα, υπενθυμίσεις, διαδρομές, dashboard, import/export, backup, email, storage και security.

### 6.1.2 Εσωτερική δομή backend feature

Μια τυπική λειτουργική περιοχή περιλαμβάνει:

- entity για το persistence model
- repository για πρόσβαση στη βάση
- service για επιχειρησιακή λογική
- controller για HTTP endpoints
- request και response DTOs
- validation
- tests

Η επανάληψη αυτού του pattern κάνει το repository πολύ πιο εύκολο στην κατανόηση και μειώνει την εξάρτηση (coupling) ανάμεσα στις διάφορες λειτουργίες.

### 6.1.3 Δομή frontend

Ο κώδικας του frontend χωρίζεται στο επίπεδο core για τα κοινόχρηστα στοιχεία και στα features για τις επιμέρους λειτουργίες της εφαρμογής.

Το core περιλαμβάνει:

- authentication
- HTTP services
- interceptors
- guards
- κοινά models
- utilities
- επαναχρησιμοποιήσιμα UI στοιχεία

Το features περιλαμβάνει τις σελίδες και τις φόρμες για λογαριασμό, γκαράζ, συντήρηση, καύσιμα, αξεσουάρ, ανταλλακτικά, έγγραφα, υπενθυμίσεις, διαδρομές και dashboard.

Η δομή επιτρέπει στις σελίδες να χρησιμοποιούν κοινές υπηρεσίες χωρίς να αντιγράφουν λογική επικοινωνίας ή μετατροπών.

### 6.1.4 Tests και migrations

Τα backend tests βρίσκονται κάτω από το `backend/src/test`, ενώ τα frontend tests τοποθετούνται δίπλα στα αντίστοιχα components, services ή utilities.

Τα migrations του Flyway βρίσκονται μέσα στα resources του backend, οπότε παρακολουθούνται κανονικά από το version control μαζί με τον υπόλοιπο κώδικα.

### 6.1.5 Αρχεία υποδομής

Το docker-compose.yml χρησιμοποιείται για development υπηρεσίες. Ο φάκελος .github/workflows περιλαμβάνει τα workflows backend, frontend και security.

Αυτή η δομή διευκολύνει το local development, τους αυτοματοποιημένους ελέγχους και το deployment, κρατώντας καθαρό τον διαχωρισμό ανάμεσα στον κώδικα της εφαρμογής και τις υποδομές (infrastructure).

## 6.2 Υλοποίηση του backend

Το backend είναι γραμμένο σε Spring Boot και ακολουθεί αρχιτεκτονική αρθρωτού μονόλιθου (modular monolith). Όλες οι λειτουργικές ενότητες τρέχουν στην ίδια εφαρμογή και μοιράζονται την ίδια PostgreSQL, αλλά διατηρούν καθαρά διαχωρισμένες ευθύνες.

### 6.2.1 Ροή επεξεργασίας αιτήματος

Μια τυπική ροή είναι:

HTTP Request
↓
Controller
↓
Service
↓
Repository ή Storage
↓
DTO Response

Ο controller χειρίζεται το HTTP επίπεδο. Το service εφαρμόζει επιχειρησιακούς κανόνες, ownership checks και συναλλαγές. Το repository επικοινωνεί με την PostgreSQL, ενώ το storage layer διαχειρίζεται τα αρχεία.

### 6.2.2 Controllers και DTOs

Οι controllers χρησιμοποιούν annotations του Spring Web για GET, POST, PUT και DELETE.

Τα JPA entities δεν επιστρέφονται απευθείας. Κάθε λειτουργική περιοχή διαθέτει request και response DTOs, ώστε:

- να μην εκτίθενται εσωτερικά πεδία
- να αποφεύγονται κυκλικές σχέσεις
- να παραμένει σταθερό το API contract
- να επιτρέπονται υπολογισμένα response fields

Τα request DTOs επικυρώνονται με Bean Validation πριν φτάσουν στο service layer.

### 6.2.3 Service layer

Τα services περιέχουν την κύρια επιχειρησιακή λογική:

- έλεγχο ιδιοκτησίας
- εφαρμογή ορίων πλάνου
- mapping μεταξύ DTOs και entities
- συντονισμό πολλών repositories
- συναλλαγές
- ενημέρωση χιλιομέτρων
- συντονισμό database και storage

Οι controllers παραμένουν λεπτοί και δεν περιέχουν επιχειρησιακούς κανόνες.

### 6.2.4 Ownership-aware πρόσβαση

Η πρόσβαση σε vehicle scoped δεδομένα δεν βασίζεται μόνο στο ID. Χρησιμοποιούνται repository queries που συνδυάζουν resource ID και owner ID, όπως:

- findByIdAndOwnerId
- findByIdAndVehicleOwnerId

Αν το στοιχείο δεν υπάρχει ή ανήκει σε άλλον χρήστη, επιστρέφεται `404 Not Found`. Έτσι αποφεύγεται η αποκάλυψη ξένων στοιχείων.

### 6.2.5 Συναλλαγές

Οι λειτουργίες που μεταβάλλουν δεδομένα εκτελούνται σε @Transactional methods. Οι αναγνώσεις μπορούν να χρησιμοποιούν readOnly = true.

Σε λειτουργίες που συνδυάζουν PostgreSQL και storage, η συναλλαγή της βάσης δεν αρκεί. Για αυτό χρησιμοποιούνται πρόσθετοι μηχανισμοί rollback και cleanup, οι οποίοι παρουσιάζονται στις Ενότητες 6.6 και 6.9.

### 6.2.6 Κεντρική διαχείριση σφαλμάτων

Κοινός exception handler μετατρέπει validation, authentication, conflict, not found και τεχνικά σφάλματα σε αποκρίσεις RFC 7807.

Το frontend έχει κοινή μορφή με status, title, detail και πρόσθετα πεδία όπου απαιτείται.

### 6.2.7 Αντιπροσωπευτική λειτουργία οχήματος

Το VehicleService υποστηρίζει δημιουργία, προβολή, ενημέρωση, διαγραφή, ορισμό κύριου οχήματος και ενημέρωση χιλιομέτρων.

Κατά τη δημιουργία:

1. ελέγχεται το επιτρεπόμενο πλήθος οχημάτων
2. δημιουργείται το entity
3. συνδέεται με τον ιδιοκτήτη
4. αποθηκεύεται
5. το πρώτο όχημα μπορεί να οριστεί ως κύριο
6. επιστρέφεται response DTO

Αντίστοιχη δομή χρησιμοποιούν οι υπηρεσίες συντήρησης, καυσίμων, εγγράφων, υπενθυμίσεων και λοιπών λειτουργιών.

## 6.3 Υλοποίηση του frontend

Το frontend είναι αναπτυγμένο σε Angular 21 και TypeScript ως Single Page Application (SPA). Η αρχιτεκτονική του διαχωρίζει καθαρά τις κοινόχρηστες υπηρεσίες από τις σελίδες των επιμέρους λειτουργιών.

### 6.3.1 Ροή αλληλεπίδρασης

Χρήστης
↓
Angular component
↓
Frontend service
↓
HttpClient
↓
REST API
↓
Signal ή component state
↓
Ενημέρωση διεπαφής (UI)

Τα components χειρίζονται την παρουσίαση και τις ενέργειες του χρήστη. Τα services κατασκευάζουν τα αιτήματα και μεταφέρουν typed DTOs. Η απόκριση ενημερώνει reactive state.

### 6.3.2 Standalone components και features

Η εφαρμογή χρησιμοποιεί standalone components. Οι εξαρτήσεις δηλώνονται άμεσα στο component και οι λειτουργικές περιοχές μπορούν να φορτώνονται ανεξάρτητα.

Ο φάκελος `core` συγκεντρώνει authentication, services, interceptors, guards, models και utilities. Ο φάκελος `features` περιλαμβάνει τις σελίδες και τις φόρμες κάθε λειτουργικής περιοχής.

### 6.3.3 Routing και vehicle shell

Το routing ορίζεται κεντρικά και περιλαμβάνει δημόσιες, προστατευμένες και vehicle-scoped διαδρομές.

Οι λειτουργίες συγκεκριμένου οχήματος οργανώνονται μέσα σε κοινό shell με:

- στοιχεία ενεργού οχήματος
- sidebar ή navigation menu
- route outlet
- κοινές ενέργειες
- responsive συμπεριφορά

Οι route guards βελτιώνουν την εμπειρία χρήσης, αλλά δεν αντικαθιστούν την εξουσιοδότηση του backend.

### 6.3.4 Frontend services και interceptors

Κάθε λειτουργική περιοχή διαθέτει service, όπως `VehicleService`, `MaintenanceService`, `FuelService`, `DocumentService` και `RouteService`.

Οι interceptors αναλαμβάνουν:

- προσθήκη JWT
- κοινό χειρισμό unauthorized responses
- μετατροπή API errors
- κοινά headers όπου απαιτούνται

Έτσι η ίδια λογική δεν επαναλαμβάνεται σε κάθε service.

### 6.3.5 Reactive state

Το `AuthService` διατηρεί κεντρικά την κατάσταση σύνδεσης και τον τρέχοντα χρήστη.

Τα Angular signals χρησιμοποιούνται για:

- τρέχοντα χρήστη
- ενεργό όχημα
- λίστες δεδομένων
- loading και error state
- φίλτρα και επιλογές προβολής

Η RxJS χρησιμοποιείται για HTTP observables, route streams και σύνθεση ασύγχρονων ροών. Τα δύο μοντέλα χρησιμοποιούνται συμπληρωματικά.

### 6.3.6 Reactive forms και validation

Οι βασικές φόρμες υλοποιούνται με reactive forms.

Το frontend ελέγχει υποχρεωτικά πεδία, μορφές και βασικά όρια για άμεση ανάδραση. Το backend επαναλαμβάνει πάντοτε τους ελέγχους, επειδή το client-side validation δεν αποτελεί μηχανισμό ασφάλειας.

### 6.3.7 Διεθνοποίηση και προτιμήσεις

Η ngx-translate φορτώνει translation files. Οι προτιμήσεις μονάδων, νομίσματος και ημερομηνίας εφαρμόζονται μέσω κοινών utilities και services.

### 6.3.8 Αρχεία και downloads

Τα uploads αποστέλλονται ως `multipart/form-data`. Για downloads χρησιμοποιείται `blob` response και filename από τα ασφαλή response headers.

Οι σελίδες φωτογραφιών, εγγράφων και συνημμένων χρησιμοποιούν κοινά UI μοτίβα για preview, λήψη και διαγραφή.

### 6.3.9 Responsive διεπαφή

Το SCSS προσαρμόζει φόρμες, λίστες, navigation και κάρτες σε διαφορετικά μεγέθη οθόνης. Σε μικρές οθόνες οι πίνακες μπορούν να αντικαθίστανται από κάρτες χωρίς απώλεια βασικών ενεργειών.

## 6.4 Αυθεντικοποίηση, εξουσιοδότηση και ασφάλεια

Η ασφάλεια του Moto-Vault υλοποιείται σε πολλαπλά επίπεδα. Η ταυτοποίηση του χρήστη δεν αρκεί από μόνη της. Κάθε πρόσβαση σε προστατευμένη εγγραφή (στοιχείο) ελέγχεται και ως προς την ιδιοκτησία.

### 6.4.1 Τοπική εγγραφή και επαλήθευση email

Η εγγραφή με email και κωδικό ακολουθεί τη ροή:

Register request
↓
Validation
↓
Έλεγχος μοναδικότητας email
↓
Hash κωδικού
↓
Δημιουργία χρήστη
↓
Verification token
↓
Email outbox

Το email μετατρέπεται σε κανονική μορφή (normalized) και ο κωδικός αποθηκεύεται αποκλειστικά ως hash.

Το verification token ισχύει για περιορισμένο χρονικό διάστημα και χρησιμοποιείται μόνο μία φορά. Η αποστολή του email γίνεται μέσω του outbox pattern, ώστε μια προσωρινή αποτυχία στον SMTP server να μην επηρεάζει τη δημιουργία του λογαριασμού.

### 6.4.2 Σύνδεση και JWT

Κατά τη σύνδεση:

- αναζητείται ο χρήστης
- συγκρίνεται ο κωδικός μέσω password encoder
- ελέγχεται η κατάσταση του λογαριασμού
- εκδίδεται JWT
- ενημερώνονται τα session metadata

Σε αποτυχία επιστρέφεται γενικό μήνυμα και δεν αποκαλύπτεται αν υπάρχει συγκεκριμένο email.

Το JWT περιλαμβάνει μόνο τις απαραίτητες claims και ελέγχεται ως προς την υπογραφή και τη λήξη του.

### 6.4.3 JwtAuthFilter και SecurityContext

Ο JwtAuthFilter διαβάζει το Authorization header, επαληθεύει το token, ανακτά τον χρήστη και τον τοποθετεί στο Spring Security context.

Οι controllers μπορούν έτσι να λαμβάνουν τον χρήστη μέσω:

Στη java:
@AuthenticationPrincipal User user

Η SecurityConfig ορίζει τα δημόσια endpoints, τα προστατευμένα endpoints, το JWT filter, το CORS και το stateless session behavior.

### 6.4.4 Session metadata και ανάκληση

Παρότι η HTTP αυθεντικοποίηση βασίζεται σε stateless JWT, η εφαρμογή διατηρεί session metadata για:

- προβολή ενεργών sessions
- ανάκληση συγκεκριμένης session
- ανάκληση όλων των sessions
- ακύρωση tokens μετά από αλλαγές ασφαλείας

Έτσι ένα token μπορεί να πάψει να γίνεται αποδεκτό πριν από τη κανονική λήξη του.

### 6.4.5 Google Sign-In

Το frontend λαμβάνει Google ID token και το αποστέλλει στο backend.

Το backend επαληθεύει signature, issuer, audience και τα απαραίτητα claims. Στη συνέχεια αντιστοιχίζει ή δημιουργεί τον χρήστη και εκδίδει δικό του JWT.

Η σύνδεση Google identity με υπάρχοντα λογαριασμό γίνεται με ξεχωριστή και ρητή ροή, ώστε να αποφεύγεται η αυτόματη κατάληψη λογαριασμού μόνο από κοινό email.

### 6.4.6 Ανάκτηση και αλλαγή κωδικού

Η ανάκτηση κωδικού χρησιμοποιεί action token περιορισμένης διάρκειας και μίας χρήσης.

Οι αποκρίσεις στο αίτημα ανάκτησης είναι γενικές, ώστε να μην αποκαλύπτεται αν υπάρχει λογαριασμός.

Μετά την αλλαγή κωδικού μπορούν να ανακληθούν οι προηγούμενες sessions.

### 6.4.7 Ownership authorization

Κάθε vehicle scoped λειτουργία ελέγχει ότι το στοιχείο ανήκει στον συνδεδεμένο χρήστη.

Οι repository queries περιλαμβάνουν τον owner και η εφαρμογή επιστρέφει `404 Not Found` τόσο για ανύπαρκτο όσο και για ξένο πόρο. Η προσέγγιση περιορίζει IDOR ευπάθειες.

### 6.4.8 Upload και parser security

Τα uploads ελέγχονται ως προς μέγεθος, extension, content type και κατηγορία. Τα filenames δεν χρησιμοποιούνται ως πραγματικά storage keys.

Για λόγους ασφαλείας, ο GPX parser απενεργοποιεί τα DTDs και τα external entities (αποτρέποντας επιθέσεις XXE). Έτσι, μπορεί και θέτει όρια στον αριθμό των σημείων (trackpoints) και επεξεργάζεται τα XML events με streaming, ώστε να μην εξαντλούνται οι πόροι του συστήματος.

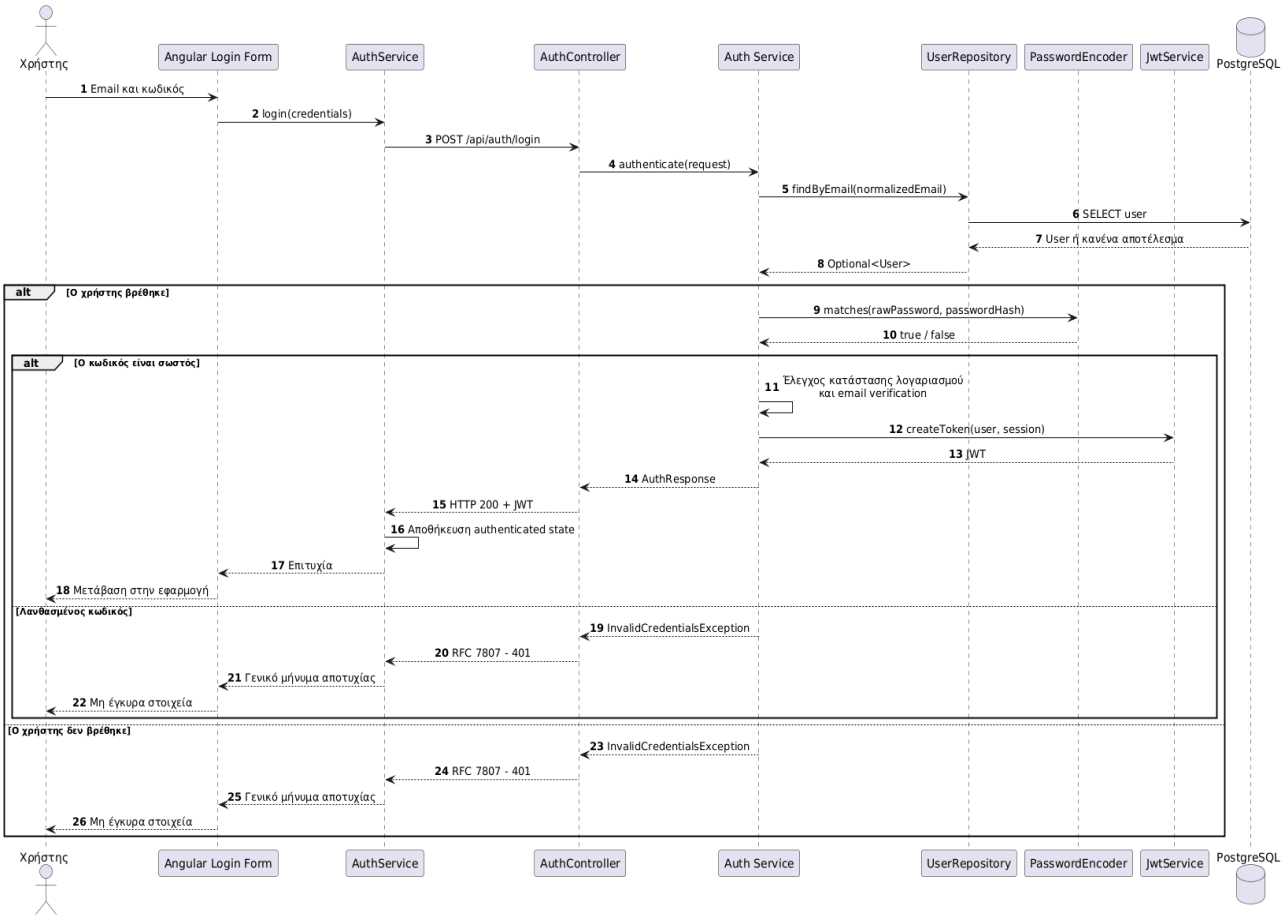
### 6.4.9 Rate limiting και production ασφάλεια

Το backend εφαρμόζει rate limiting σε ευαίσθητες λειτουργίες και εμπιστεύεται forwarded headers μόνο από ρυθμισμένους proxies.

Ο production validator αποτρέπει την εκκίνηση με κενά secrets, μη ασφαλή origins ή ελλιπή storage configuration.

Η ασφάλεια συμπληρώνεται από HTTPS, ιδιωτική βάση, ιδιωτικό object storage και περιορισμένη έκθεση θυρών.

**Σχήμα 6.1**: Διάγραμμα ακολουθίας της διαδικασίας σύνδεσης χρήστη




## 6.5 Υλοποίηση της διαχείρισης δεδομένων

Η διαχείριση δεδομένων γίνεται με την PostgreSQL, το Spring Data JPA και το Flyway.

### 6.5.1 Persistence model

Οι βασικές JPA οντότητες είναι:

- User
- Vehicle
- MaintenanceRecord
- FuelEntry
- Accessory
- Part
- VehicleDocument
- Reminder
- Route
- entities συνημμένων
- session και action token entities
- email outbox και storage deletion entities

Οι κύριες οντότητες χρησιμοποιούν Long και όχι UUID.

### 6.5.2 Σχέσεις ιδιοκτησίας

Η βασική ιεραρχία είναι:

```
User
 ↓
Vehicle
  ├── MaintenanceRecord
  ├── FuelEntry
  ├── Accessory
  ├── Part
  ├── VehicleDocument
  ├── Reminder
  ├── VehiclePhoto
  └── Route
```

Ένας χρήστης μπορεί να διαθέτει πολλά οχήματα, ενώ κάθε όχημα ανήκει σε έναν χρήστη.

Το mainVehicleId αποθηκεύει το κύριο όχημα του χρήστη. Κατά τη διαγραφή του κύριου οχήματος η αναφορά ενημερώνεται ή καθαρίζεται.

### 6.5.3 Repositories και queries

Κάθε λειτουργική περιοχή διαθέτει Spring Data repository.

Τα repositories υποστηρίζουν:

- CRUD λειτουργίες
- ownership-aware αναζητήσεις
- counts και limits
- ταξινόμηση
- συγκεντρωτικά queries
- processing queues

Για απλές περιπτώσεις χρησιμοποιούνται derived query methods. Για σύνθετα aggregations χρησιμοποιούνται JPQL ή native queries.

### 6.5.4 DTO mapping

Η μετατροπή των entities σε response DTOs γίνεται στο service layer.

Μέσω αυτής της πρακτικής αποφεύγουμε προβλήματα με lazy loading, αποκρύπτουμε εσωτερικά πεδία και μπορούμε να προσθέτουμε υπολογισμένα πεδία χωρίς να πειράζουμε το μοντέλο της βάσης.

### 6.5.5 Συναλλαγές και dirty checking

Οι μεταβολές εκτελούνται σε @Transactional methods. Οι read only λειτουργίες χρησιμοποιούν @Transactional(readOnly = true) όπου είναι κατάλληλο.

Το Hibernate dirty checking αποθηκεύει τις αλλαγές σε managed entities μέσα στη συναλλαγή. Σε περιπτώσεις όπου το entity δεν είναι managed γίνεται αποθήκευση.

### 6.5.6 Constraints και indexes
Τα Flyway migrations δημιουργούν foreign keys, unique constraints, check constraints και indexes.

Indexes χρησιμοποιούνται για συχνές αναζητήσεις ανά owner, vehicle, ημερομηνία και κατάσταση queue. Η επιλογή τους βασίζεται στα πραγματικά query patterns.

### 6.5.7 Ενημέρωση χιλιομέτρων
Ορισμένες λειτουργίες, όπως συντήρηση και ανεφοδιασμός, μπορούν να ενημερώσουν την τρέχουσα χιλιομετρική ένδειξη του οχήματος.

Η ενημέρωση γίνεται μόνο όταν η νέα τιμή είναι έγκυρη και μεγαλύτερη από την υπάρχουσα. Η χαμηλότερη ιστορική τιμή διατηρείται στην εγγραφή χωρίς να μειώνει την τρέχουσα ένδειξη.

### 6.5.8 Υπολογισμός κατανάλωσης
Η κατανάλωση υπολογίζεται ανάμεσα σε κατάλληλους πλήρεις ανεφοδιασμούς. Οι ενδιάμεσοι μερικοί ανεφοδιασμοί προστίθενται στα συνολικά λίτρα του διαστήματος.

Αν δεν υπάρχουν επαρκή δεδομένα, το σύστημα δεν εμφανίζει παραπλανητικό αποτέλεσμα.

### 6.5.9 Καταστάσεις εγγράφων και υπενθυμίσεων
Η κατάσταση εγγράφου υπολογίζεται από την ημερομηνία λήξης. Οι υπενθυμίσεις αξιολογούνται με βάση ημερομηνία, χιλιόμετρα ή συνδυασμό τους.

Οι κανόνες βρίσκονται στο backend, ώστε να εφαρμόζονται με τον ίδιο τρόπο ανεξάρτητα από το frontend.

### 6.5.10 Migrations
Κάθε αλλαγή schema προστίθεται ως νέο Flyway migration. Το Hibernate χρησιμοποιείται για validation και όχι για ανεξέλεγκτη δημιουργία ή μεταβολή του production schema.

## 6.6 Διαχείριση αρχείων και φωτογραφιών
Τα αρχεία αποθηκεύονται μέσω ανεξάρτητου storage provider, ενώ η PostgreSQL διατηρεί metadata και σχέσεις.

### 6.6.1 Κατηγορίες αρχείων
Η εφαρμογή διαχειρίζεται:

- φωτογραφίες οχήματος
- συνημμένα συντήρησης
- αρχεία αξεσουάρ και ανταλλακτικών
- έγγραφα οχήματος
- GPX
- αρχεία import, export και backup

Κάθε κατηγορία μπορεί να έχει διαφορετικά όρια και επιτρεπόμενους τύπους.

### 6.6.2 StorageService
Η διεπαφή StorageService ορίζει κοινές λειτουργίες αποθήκευσης, ανάκτησης, ελέγχου ύπαρξης και διαγραφής.

Οι υλοποιήσεις είναι:
- LocalStorageService
- S3StorageService

Η επιλογή provider πραγματοποιείται μέσω configuration.

### 6.6.3 Storage keys και filenames
Το πραγματικό storage key δημιουργείται από την εφαρμογή και δεν βασίζεται στο αρχικό filename.

Το αρχικό όνομα διατηρείται μόνο ως metadata και ως προτεινόμενο filename λήψης.

Ο sanitization μηχανισμός αφαιρεί path separators, control characters, newlines και άλλους μη ασφαλείς χαρακτήρες.

### 6.6.4 Upload validation
Η UploadValidator ελέγχει:
- ύπαρξη αρχείου
- μέγιστο μέγεθος
- extension
- content type
- κατηγορία upload
- επιτρεπόμενο όριο χρήσης storage

Το frontend εκτελεί βασικούς προληπτικούς ελέγχους, αλλά το backend παραμένει η τελική πηγή validation.

### 6.6.5 Metadata
Για κάθε αρχείο αποθηκεύονται:
- storage path
- αρχικό filename
- content type
- μέγεθος
- ημερομηνία δημιουργίας
- σχέση με τη κύρια εγγραφή (Parent entity relationship)

Η βάση αποτελεί την πηγή αλήθειας για το αν ένα storage object θεωρείται ενεργό μέρος του συστήματος.

### 6.6.6 Ιδιωτική λήψη
Τα buckets και τα αρχεία δεν είναι δημόσια.

Η λήψη ακολουθεί τη ροή:
1. αυθεντικοποίηση
2. ownership check
3. ανάκτηση metadata
4. ανάκτηση από storage
5. ασφαλή response headers
6. streaming προς τον client

### 6.6.7 Φωτογραφίες οχήματος
Οι φωτογραφίες συνδέονται με συγκεκριμένο όχημα. Ο χρήστης μπορεί να τις προβάλλει, να τις διαγράφει και να επιλέγει κύρια φωτογραφία.

Αν διαγραφεί η κύρια φωτογραφία, η αναφορά του οχήματος καθαρίζεται ή ενημερώνεται.

### 6.6.8 Αποτυχία κατά το upload
Η αποθήκευση αρχείου και η δημιουργία metadata δεν ανήκουν στην ίδια συναλλαγή.

Αν αποθηκευτεί αρχείο αλλά αποτύχει η database συναλλαγή, ο συντονιστής storage προσπαθεί να αναιρέσει το upload. Αν η άμεση διαγραφή αποτύχει, καταγράφεται εργασία για μεταγενέστερο cleanup.

### 6.6.9 Durable διαγραφή
Κατά τη διαγραφή μιας εγγραφής, τα storage keys καταγράφονται σε durable deletion queue.

Η database συναλλαγή μπορεί να ολοκληρωθεί και ο worker να διαγράψει τα objects με retries. Έτσι προσωρινή αποτυχία provider δεν αφήνει τη βασική λειτουργία σε ασαφή κατάσταση.

### 6.6.10 Reconciliation
Περιοδικές διαδικασίες μπορούν να εντοπίζουν:
- metadata χωρίς πραγματικό object
- objects χωρίς ενεργό metadata
- αποτυχημένες deletion tasks

Οι μηχανισμοί αυτοί παρουσιάζονται συνολικά στην Ενότητα 6.9.

## 6.7 Επεξεργασία και παρουσίαση αρχείων GPX
Το Moto-Vault μεταφορτώνει, αναλύει, αποθηκεύει και παρουσιάζει αρχεία GPX.

### 6.7.1 Ροή επεξεργασίας
Επιλογή αρχείου
↓
Frontend validation
↓
Multipart upload
↓
Ownership και upload validation
↓
Streaming parsing
↓
Υπολογισμός metadata
↓
Αποθήκευση αρχείου και Route entity
↓
Προβολή στον χάρτη

### 6.7.2 GpxParser

Η κλάση GpxParser είναι ανεξάρτητη από Spring και χρησιμοποιεί StAX streaming parser.

Το GPX δεν φορτώνεται ως πλήρες DOM tree και μειώνει τη χρήση μνήμης.

Ο parser απενεργοποιεί DTD και external entities για προστασία από XXE.

### 6.7.3 Όρια

Εφαρμόζονται ανώτατα όρια σε track points, waypoints και XML events. Αρχεία που υπερβαίνουν τα όρια απορρίπτονται πριν προκαλέσουν υπερβολική χρήση resources.

### 6.7.4 Tracks και segments

Ο parser υποστηρίζει περισσότερα tracks και segments.

Στην αρχή κάθε segment μηδενίζονται το προηγούμενο σημείο και ο προηγούμενος χρόνος, ώστε να μην υπολογίζεται λάθος απόσταση ανάμεσα σε ασύνδετα τμήματα.

### 6.7.5 Απόσταση και διάρκεια

Η απόσταση υπολογίζεται ανάμεσα σε διαδοχικά σημεία του ίδιου segment με τον τύπο Haversine.

Ο υπολογισμός της διάρκειας βασίζεται στις χρονικές διαφορές μεταξύ των διαδοχικών σημείων. Σημεία με λανθασμένη χρονική σειρά (μη μονότονα timestamps) ή αδικαιολόγητα μεγάλα κενά απορρίπτονται, καθώς δεν αντιπροσωπεύουν ομαλή κίνηση.

Οι υπολογισμένες τιμές στρογγυλοποιούνται στην κατάλληλη ακρίβεια πριν αποθηκευτούν στη βάση.

### 6.7.6 Υψομετρικά στοιχεία και waypoints

Όταν το αρχείο περιέχει υψομετρικά δεδομένα, η συνολική άνοδος και κάθοδος υπολογίζονται αφού πρώτα φιλτραριστούν οι μικρές διακυμάνσεις (θόρυβος) του υψομέτρου.

Τα waypoints εξάγονται ως σημεία ενδιαφέροντος (POIs) και μπορούν να προβληθούν απευθείας στον χάρτη.

### 6.7.7 Route entity και endpoints

Το Route αποθηκεύει όνομα, ημερομηνίες, απόσταση, διάρκεια, αριθμό σημείων, segments, υψομετρικά στοιχεία, αριθμό POIs, filename, μέγεθος και storage path.

Τα endpoints βρίσκονται κάτω από:

/api/vehicles/{vehicleId}/routes

Υποστηρίζουν λίστα, upload, προβολή, ενημέρωση metadata, λήψη και διαγραφή.

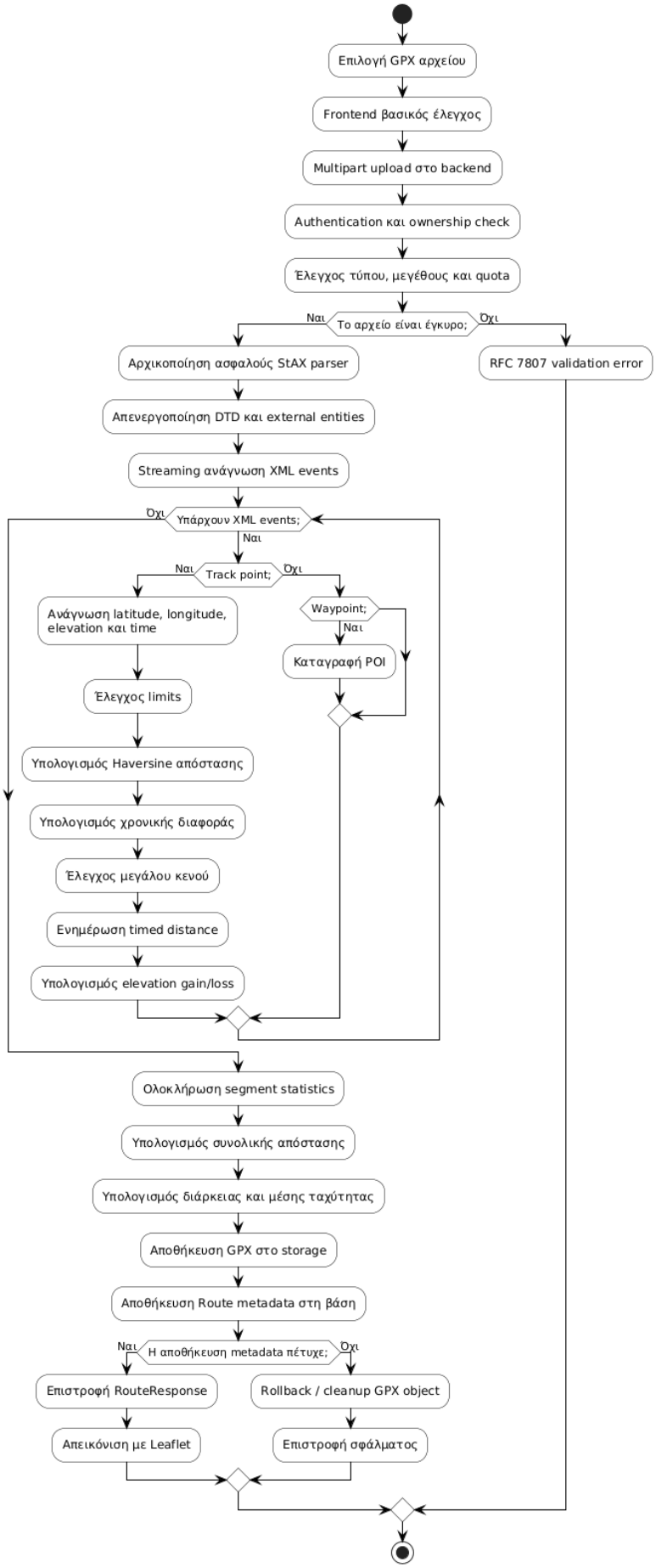
### 6.7.8 Παρουσίαση με Leaflet

Το frontend λαμβάνει τα σημεία και τα metadata και δημιουργεί polyline μέσω Leaflet.

Ο χάρτης προσαρμόζει το viewport στα όρια της διαδρομής και εμφανίζει markers ή σημεία ενδιαφέροντος όπου υπάρχουν.

Τα tiles φορτώνονται από το OpenStreetMap και εμφανίζεται η απαιτούμενη attribution.

**Σχήμα 6.2**: Ροή εισαγωγής, επεξεργασίας και αποθήκευσης αρχείου GPX


Επιλογή GPX αρχείου
Frontend βασικός έλεγχος
Multipart upload στο backend
Authentication και ownership check
Έλεγχος τύπου, μεγέθους και quota
Το αρχείο είναι έγκυρο;
Ναι
Όχι
RFC 7807 validation error
Αρχικοποίηση ασφαλούς StAX parser
Απενεργοποίηση DTD και external entities
Streaming ανάγνωση XML events
Υπάρχουν XML events;
Όχι
Ναι
Track point;
Ναι
Όχι
Ανάγνωση latitude, longitude, elevation και time
Waypoint;
Ναι
Καταγραφή POI
Έλεγχος limits
Υπολογισμός Haversine απόστασης
Υπολογισμός χρονικής διαφοράς
Έλεγχος μεγάλου κενού
Ενημέρωση timed distance
Υπολογισμός elevation gain/loss
Ολοκλήρωση segment statistics
Υπολογισμός συνολικής απόστασης
Υπολογισμός διάρκειας και μέσης ταχύτητας
Αποθήκευση GPX στο storage
Αποθήκευση Route metadata στη βάση
Η αποθήκευση metadata πέτυχε;
Ναι
Όχι
Επιστροφή RouteResponse
Απεικόνιση με Leaflet
Rollback / cleanup GPX object
Επιστροφή σφάλματος

## 6.8 Εισαγωγή, εξαγωγή και δημιουργία αναφορών

Το Moto-Vault υποστηρίζει εξαγωγή οχήματος, προεπισκόπηση και εισαγωγή archive, backup και restore λογαριασμού και δημιουργία PDF ιστορικού συντήρησης.

### 6.8.1 Εξαγωγή οχήματος

Η εξαγωγή δημιουργεί ZIP που περιλαμβάνει:

- manifest.json
- data.json
- φωτογραφίες
- συνημμένα
- έγγραφα
- GPX αρχεία

Το archive δεν εκθέτει database IDs ή πραγματικά storage paths. Χρησιμοποιεί archive local αναγνωριστικά για να διατηρεί τις σχέσεις.

Το manifest περιλαμβάνει format identifier, έκδοση, σχέσεις, μεγέθη και SHA-256 checksums.

### 6.8.2 Streaming δημιουργία archive

Η υπηρεσία δημιουργεί προσωρινό ZIP μέσω `ZipOutputStream` και το επιστρέφει ως stream. Αμέσως μόλις κλείσει το HTTP response, το προσωρινό αρχείο διαγράφεται αυτόματα για να μη μένουν σκουπίδια στο σύστημα.

Η προσέγγιση αποφεύγει τη διατήρηση ολόκληρου archive στη μνήμη.

### 6.8.3 Προεπισκόπηση εισαγωγής

Το endpoint preview:

- δεν γράφει δεδομένα
- ελέγχει manifest και format
- επικυρώνει sizes και checksums
- ελέγχει σχέσεις
- εμφανίζει counts και βασικά στοιχεία

Η πραγματική εισαγωγή εκτελείται μόνο μετά από επιβεβαίωση.

### 6.8.4 Ασφάλεια ZIP

Ο reader απορρίπτει:

- unsafe paths και Zip Slip
- duplicate entries
- ασύμβατο format
- μη έγκυρα checksums
- υπερβολικό πλήθος ή συνολικό μέγεθος entries
- μη έγκυρες σχέσεις

Τα όρια μειώνουν τον κίνδυνο ZIP bomb και εξάντλησης resources.

### 6.8.5 Εισαγωγή

Κατά την εισαγωγή:

1. δημιουργείται νέο όχημα
2. αντιστοιχίζονται archive-local IDs σε νέα database IDs
3. δημιουργούνται οι εξαρτώμενες εγγραφές με σωστή σειρά
4. αποθηκεύονται τα αρχεία
5. εφαρμόζονται limits και ownership
6. σε αποτυχία εκτελείται cleanup

Η εισαγωγή δεν αντικαθιστά υπάρχον όχημα αλλά δημιουργεί νέο σύνολο δεδομένων.

### 6.8.6 Backup και restore λογαριασμού

Το account backup περιλαμβάνει όλα τα υποστηριζόμενα οχήματα και δεδομένα του χρήστη.

Κατά το restore, οι συσχετίσεις ανακατασκευάζονται με νέα IDs, ενώ εξαιρούνται ευαίσθητα δεδομένα ταυτοποίησης (όπως credentials) που δεν πρέπει να μεταφέρονται.

### 6.8.7 PDF ιστορικού συντήρησης

Η αναφορά PDF δημιουργείται με OpenPDF και περιλαμβάνει βασικά στοιχεία οχήματος και οργανωμένο ιστορικό εργασιών.

Το παραγόμενο αρχείο ελέγχεται στα tests με PDFBox, ώστε να επιβεβαιώνεται ότι είναι έγκυρο και περιλαμβάνει το αναμενόμενο περιεχόμενο.

## 6.9 Background εργασίες και μηχανισμοί αξιοπιστίας

Ορισμένες λειτουργίες δεν είναι κατάλληλο να εξαρτώνται από την άμεση επιτυχία μιας εξωτερικής υπηρεσίας. Για αυτό το backend χρησιμοποιεί durable queues, retries και reconciliation.

### 6.9.1 Βασική αρχή

Η κύρια επιχειρησιακή συναλλαγή ολοκληρώνεται στη βάση και οι εκκρεμείς εξωτερικές ενέργειες αποθηκεύονται ως εγγραφές προς επεξεργασία.

Με αυτόν τον τρόπο:

- προσωρινή αποτυχία SMTP δεν ακυρώνει τη δημιουργία λογαριασμού,
- προσωρινή αποτυχία object storage δεν ακυρώνει απαραίτητα τη διαγραφή metadata,
- οι εκκρεμείς εργασίες επιβιώνουν από restart,
- οι retries εκτελούνται ελεγχόμενα.

### 6.9.2 Email outbox

Οι λειτουργίες που απαιτούν transactional email δημιουργούν εγγραφή στο `email_outbox` μέσα στην ίδια database συναλλαγή.

Ο EmailOutboxWorker αναζητά pending εγγραφές, αποστέλλει τα email μέσω SMTP και ενημερώνει την κατάστασή τους.

Σε αποτυχία αυξάνεται ο αριθμός προσπαθειών και προγραμματίζεται επόμενη εκτέλεση με καθυστέρηση.

Η προσέγγιση στοχεύει σε ασφαλή at-least-once επεξεργασία και περιορισμό διπλών αποστολών μέσω status transitions και locking.

### 6.9.3 Storage deletion queue

Η διαγραφή metadata και object storage δεν αποτελεί κοινή συναλλαγή.

Κατά τη διαγραφή μιας λειτουργικής εγγραφής, τα storage keys καταγράφονται σε queue. Ο StorageDeletionWorker προσπαθεί να διαγράψει τα objects και επαναλαμβάνει προσωρινές αποτυχίες.

Η διαγραφή θεωρείται idempotent, ώστε η επανάληψη σε ήδη διαγραμμένο object να μην προκαλεί νέα ασυνέπεια.

### 6.9.4 Rollback tracking

Κατά τη δημιουργία νέων αρχείων, η υπηρεσία παρακολουθεί ποια objects αποθηκεύτηκαν πριν ολοκληρωθεί η database συναλλαγή.

Αν η συναλλαγή αποτύχει, γίνεται προσπάθεια άμεσης διαγραφής. Αν αποτύχει και αυτή, δημιουργείται durable cleanup εργασία.

### 6.9.5 Reconciliation

Η StorageReconciliationService συγκρίνει metadata και storage και εντοπίζει:

- metadata χωρίς object
- objects χωρίς ενεργή αναφορά
- παλιές αποτυχημένες deletion tasks

Η διαδικασία πρέπει να αποφεύγει τη διαγραφή object που μπορεί ακόμη να χρησιμοποιείται. Για αυτό εφαρμόζονται χρονικά όρια, έλεγχοι αναφορών και λειτουργία ασφαλούς καταγραφής.

### 6.9.6 Concurrency και locking

Για την αποφυγή race conditions, εξασφαλίζουμε ότι καμία εργασία δεν θα περάσει από ταυτόχρονη επεξεργασία από περισσότερους από έναν workers.

Ανάλογα με το σενάριο, αυτό επιτυγχάνεται με ατομικές αλλαγές κατάστασης (atomic status transitions), locking σε επίπεδο repository ή περιορισμό του batch size.

### 6.9.7 Παρατηρησιμότητα

Οι επαναλαμβανόμενες αποτυχίες καταγράφονται χωρίς να διαρρέουν ευαίσθητα δεδομένα (credentials). Οι καταστάσεις pending, completed και failed επιτρέπουν την εύκολη παρακολούθηση και τον έλεγχο των workers.

Αν και αυτοί οι μηχανισμοί αυξάνουν την αξιοπιστία, δεν αντικαθιστούν ένα πλήρες distributed messaging σύστημα. Παραμένουν σκόπιμα ενσωματωμένοι στον αρθρωτό μονόλιθο (modular monolith), διατηρώντας την αρχιτεκτονική απλή.

## 6.10 Αυτοματοποιημένες δοκιμές

Το MotoVault περιλαμβάνει ένα ολοκληρωμένο σετ δοκιμών (backend & frontend) που καλύπτει τους επιχειρησιακούς κανόνες, την ασφάλεια, το persistence, τη διαχείριση αρχείων, την επεξεργασία GPX, τις λειτουργίες import/export, την παραγωγή PDF και τις background εργασίες.

Τα τελικά αριθμητικά στοιχεία (κάλυψη κώδικα, πλήθος tests) θα επικαιροποιηθούν απευθείας από την τελική κατάσταση του repository πριν την παράδοση.

### 6.10.1 Backend tests

Το backend χρησιμοποιεί JUnit 5, Mockito, Spring Boot Test και Spring Security Test.

Οι βασικές κατηγορίες είναι:

- unit tests
- service tests
- repository tests
- controller tests
- security tests
- integration tests

Τα unit tests ελέγχουν calculators, validators, parsers και utilities. Τα service tests ελέγχουν επιχειρησιακές ροές με mocked dependencies. Τα repository tests ελέγχουν mappings, ownership-aware queries και constraints. Τα controller tests ελέγχουν endpoints, status codes, validation και RFC 7807 errors.

### 6.10.2 Security tests

Τα security tests καλύπτουν:

- αιτήματα χωρίς JWT
- άκυρα ή ανακλημένα tokens
- πρόσβαση σε πόρους άλλου χρήστη
- Google authentication
- protected endpoints
- rate limiting
- ευαίσθητα account actions

### 6.10.3 Storage και file tests

Οι δοκιμές ελέγχουν:

- upload και download
- filename sanitization
- extensions και size limits
- quotas
- rollback
- deletion queue
- reconciliation
- provider failures

### 6.10.4 GPX tests

Ο GpxParser ελέγχεται ανεξάρτητα από Spring.

Οι δοκιμές καλύπτουν:

- Haversine distance
- πολλά segments
- timestamps
- διάρκεια
- elevation gain και loss
- waypoints
- malformed XML
- parser limits
- απενεργοποίηση DTD και external entities

### 6.10.5 Import, export και PDF tests

Οι δοκιμές archive καλύπτουν:

- έγκυρο ZIP
- manifest
- checksums
- unsafe paths
- duplicate entries
- limits
- σχέσεις
- cleanup σε αποτυχία

Τα PDF tests χρησιμοποιούν Apache PDFBox για να επιβεβαιώσουν ότι το παραγόμενο αρχείο είναι έγκυρο και περιλαμβάνει αναμενόμενο περιεχόμενο.

### 6.10.6 Background worker tests

Οι workers email και storage ελέγχονται ως προς:

- επιτυχή επεξεργασία
- retries
- αλλαγές status
- idempotency
- αποτυχία provider
- αποφυγή διαγραφής ενεργών αναφορών

### 6.10.7 Frontend tests

Το frontend χρησιμοποιεί Vitest και Angular testing utilities.

Καλύπτονται:

- services και HTTP calls
- components
- validators
- utilities μονάδων και μορφοποίησης
- reactive state
- guards και interceptors
- βασικές αλληλεπιδράσεις φορμών

### 6.10.8 Κάλυψη και CI

Η JaCoCo μετρά την κάλυψη του backend και εφαρμόζει ελάχιστο threshold στο 70%. Τα tests backend και frontend εκτελούνται στα αντίστοιχα GitHub Actions workflows.

Η κάλυψη δεν αποδεικνύει την απουσία σφαλμάτων. Λειτουργεί ως ένας από τους ελέγχους ποιότητας μαζί με τα integration tests, τα security checks και το production build.

## 6.11 Ρυθμίσεις περιβάλλοντος και λειτουργία της εφαρμογής

Το Moto-Vault υποστηρίζει διαφορετικό configuration για development και production.

### 6.11.1 Spring profiles

Τα κύρια αρχεία backend είναι:

- application.yml
- application-dev.yml
- application-prod.yml

Το κοινό αρχείο περιλαμβάνει τις μη ευαίσθητες ρυθμίσεις. Αντίστοιχα, τα αρχεία για κάθε περιβάλλον (environment specific) ορίζουν τις επιμέρους υπηρεσίες και χρησιμοποιούν μεταβλητές περιβάλλοντος (environment variables).

### 6.11.2 Development περιβάλλον

Το development profile υποστηρίζει:

- PostgreSQL
- MinIO
- Mailpit
- τοπικό ή S3 storage
- Angular development server και proxy

Το Docker Compose διασφαλίζει ότι οι τοπικές υπηρεσίες τρέχουν πάντα με τον ίδιο, επαναλήψιμο τρόπο στο περιβάλλον ανάπτυξης.

### 6.11.3 Production περιβάλλον

Οι production ρυθμίσεις παρέχονται μέσω environment variables και καλύπτουν:

- database connection
- JWT και action token secrets
- SMTP
- storage provider
- S3 endpoint και credentials
- frontend origin και CORS
- trusted proxies
- Google client configuration

Οι τιμές αυτές δεν αποθηκεύονται στο Git.

### 6.11.4 Typed configuration

Σχετικές ρυθμίσεις οργανώνονται σε typed configuration properties για CORS, email, Google, action tokens, storage και rate limiting. Η typed μορφή επιτρέπει validation και αποφεύγει διάσπαρτα string lookups.

### 6.11.5 ProductionConfigurationValidator

Ο validator ελέγχει τις κρίσιμες ρυθμίσεις κατά την εκκίνηση και μπορεί να απορρίψει:

- κενά ή default secrets
- μη ασφαλή origins
- ελλιπή storage settings
- μη έγκυρο production provider
- ασύμβατες τιμές

Η fail fast συμπεριφορά εμποδίζει την εκκίνηση σε εμφανώς μη ασφαλή κατάσταση.

### 6.11.6 Βάση και migrations

Στο development τα Flyway migrations μπορούν να εκτελούνται μαζί με την εφαρμογή.

Στον production σχεδιασμό εκτελούνται πριν από το runtime backend από ξεχωριστό migration process. Το Hibernate χρησιμοποιεί schema validation και το Open Session in View παραμένει απενεργοποιημένο.

### 6.11.7 Storage και email

Ο storage provider επιλέγεται μέσω configuration:

- local
- s3

Το local path πρέπει να είναι writable και εκτός source tree. Η S3 ρύθμιση περιλαμβάνει endpoint, region, bucket και credentials.

Στο development το Mailpit λαμβάνει τα email. Σε production απαιτείται πραγματικός transactional SMTP provider.

### 6.11.8 Frontend configuration

Το frontend χρησιμοποιεί development proxy για τα τοπικά API requests.

Μη ευαίσθητες runtime πληροφορίες, όπως η διαθεσιμότητα Google Sign-In ή άλλων features, μπορούν να ανακτηθούν από δημόσιο configuration endpoint.

### 6.11.9 Container λειτουργία

Το backend εκτελείται εσωτερικά στην port 8080, χωρίς να χρειάζεται δημόσια έκθεση. Ο reverse proxy δρομολογεί τα εξωτερικά αιτήματα.

Health checks, restart policies και logs χρησιμοποιούνται για τη βασική λειτουργική παρακολούθηση.

### 6.11.10 Μυστικά και logs

Secrets, tokens και credentials παρέχονται μόνο μέσω ασφαλούς configuration.

Τα logs δεν πρέπει να περιλαμβάνουν passwords, JWTs, reset tokens, access keys ή πλήρες ευαίσθητο περιεχόμενο αρχείων.

## 6.12 Συνολική αποτίμηση της υλοποίησης

Η υλοποίηση μετατρέπει τις απαιτήσεις και τον σχεδιασμό του Κεφαλαίου 4 σε λειτουργική εφαρμογή, χρησιμοποιώντας το τεχνολογικό οικοσύστημα του Κεφαλαίου 5.

### 6.12.1 Κάλυψη βασικών λειτουργιών

Η εφαρμογή υποστηρίζει:

- λογαριασμούς και σύνδεση μέσω email ή Google
- περισσότερες από μία μοτοσυκλέτες
- συντήρηση και συνημμένα
- ανεφοδιασμούς και στατιστικά
- αξεσουάρ και ανταλλακτικά
- έγγραφα και υπενθυμίσεις
- φωτογραφίες
- GPX διαδρομές
- dashboard
- εισαγωγή και εξαγωγή
- PDF αναφορά
- backup και restore λογαριασμού

### 6.12.2 Αρχιτεκτονική

Η μοντέρνα αρχιτεκτονική του αρθρωτού μονόλιθου (modular monolith) είναι πολύ επιλογή για ανάπτυξη από ένα άτομο. Εξασφαλίζει ACID συναλλαγές, απλοποιεί το deployment και επιτρέπει την άμεση επικοινωνία μεταξύ των modules.

Η οργάνωση σε package by feature και ο καθαρός διαχωρισμός σε controller, service, repository, entity και DTO αποτρέπουν τη δημιουργία χαοτικού και μη δομημένου κώδικα.

### 6.12.3 Ασφάλεια και αξιοπιστία

Η υλοποίηση περιλαμβάνει:

- JWT και Google token verification
- ownership-aware authorization
- session revocation
- email verification και password recovery
- private file access
- upload και parser validation
- rate limiting
- production configuration validation

Η αξιοπιστία ενισχύεται από database transactions, email outbox, storage deletion queue, retries, rollback tracking και reconciliation.

### 6.12.4 Συντηρησιμότητα

Η συντηρησιμότητα υποστηρίζεται από:

- feature-based οργάνωση backend και frontend
- DTOs
- κοινές υπηρεσίες και UI components
- typed configuration
- Flyway migrations

- automated tests
- CI workflows

### 6.12.5 Περιορισμοί

Η υλοποίηση εξακολουθεί να έχει περιορισμούς:

- δεν έχει αξιολογηθεί με μεγάλης κλίμακας load tests
- το single VPS μοντέλο δεν προσφέρει high availability
- η database και object storage απαιτεί background μηχανισμούς
- η αξιολόγηση αφορά κυρίως την πρώτη εμπειρία χρηστών
- ορισμένες παραγωγικές δυνατότητες εξαρτώνται από την τελική εγκατάσταση και το configuration

### 6.12.6 Συμπέρασμα

Στην πράξη, το MotoVault αποτυπώνει καλά όλες τις σχεδιαστικές αποφάσεις που ορίστηκαν αρχικά.

Η εφαρμογή συνδυάζει οργανωμένο backend, reactive frontend, σχεσιακή βάση, ιδιωτική αποθήκευση αρχείων, μηχανισμούς αξιοπιστίας και αυτοματοποιημένες δοκιμές.

Το επόμενο κεφάλαιο παρουσιάζει το τελικό σύστημα από την πλευρά του χρήστη.

# 7. Παρουσίαση του Moto-Vault

## 7.1 Εισαγωγή

Το παρόν κεφάλαιο παρουσιάζει το τελικό αποτέλεσμα της υλοποίησης του Moto-Vault από την πλευρά του χρήστη. Σε αντίθεση με το προηγούμενο κεφάλαιο, όπου εξετάστηκαν η εσωτερική δομή και οι τεχνικοί μηχανισμοί, εδώ το ενδιαφέρον μεταφέρεται στις διαθέσιμες λειτουργίες και στον τρόπο με τον οποίο παρουσιάζονται μέσω του UI.

Η τελική έκδοση της εφαρμογής είναι προσβάσιμη μέσω της διεύθυνσης: https://moto-vault.app/

Η παρουσίαση ακολουθεί ένα αντιπροσωπευτικό σενάριο χρήσης της εφαρμογής (user flow). Ξεκινά από την εγγραφή και τη σύνδεση, συνεχίζει με το ψηφιακό γκαράζ και την επιλογή μοτοσυκλέτας και έπειτα εξετάζει τις βασικές λειτουργικές περιοχές, όπως συντήρηση, καύσιμα, αξεσουάρ, ανταλλακτικά, έγγραφα, υπενθυμίσεις, διαδρομές GPX και φωτογραφίες. Παρουσιάζονται επίσης οι λειτουργίες εισαγωγής, εξαγωγής, δημιουργίας αναφορών και προσαρμογής της διεπαφής.

Χρησιμοποιούνται στιγμιότυπα οθόνης από την εφαρμογή, τα οποία συνοδεύονται από περιγραφή της λειτουργίας που παρουσιάζουν. Στόχος δεν είναι η δημιουργία πλήρους εγχειριδίου χρήσης, αλλά η τεκμηρίωση των βασικών ροών και του τελικού αποτελέσματος.

Τα περισσότερα στιγμιότυπα προέρχονται από το desktop browser view, ώστε οι πληροφορίες να είναι ευανάγνωστες. Στην Ενότητα 7.12 παρουσιάζονται προβολές από μικρότερη οθόνη κινητού.

Το κεφάλαιο ολοκληρώνεται με συνολική αποτίμηση του UI και μετάβαση στην αξιολόγηση χρηστικότητας του Κεφαλαίου 8.

## 7.2 Εγγραφή, σύνδεση και διαχείριση λογαριασμού

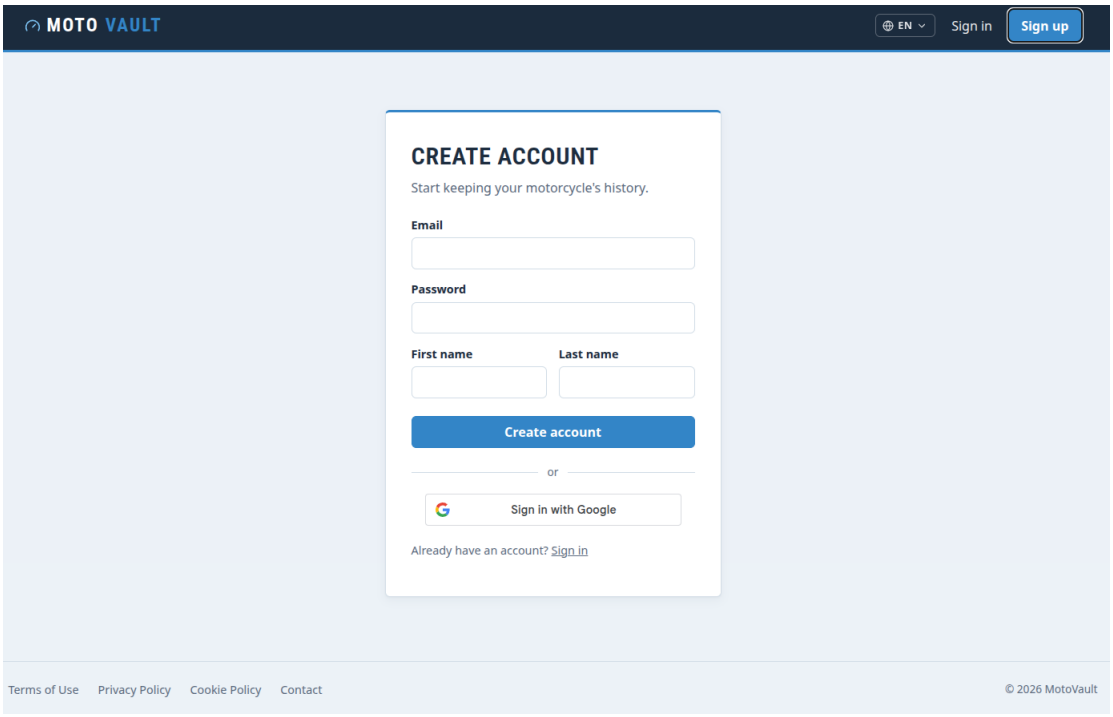
Η πρώτη επαφή του χρήστη με το Moto-Vault πραγματοποιείται μέσω των σελίδων εγγραφής και σύνδεσης. Οι δημόσιες αυτές οθόνες χρησιμοποιούν απλή διάταξη και εστιάζουν στις βασικές ενέργειες.

### 7.2.1 Δημιουργία λογαριασμού

Η σελίδα εγγραφής επιτρέπει τη δημιουργία λογαριασμού με email, κωδικό πρόσβασης, όνομα και επώνυμο. Παρέχεται επίσης δυνατότητα σύνδεσης μέσω Google.

Τα υποχρεωτικά πεδία και τα μηνύματα επικύρωσης εμφανίζονται μέσα στη φόρμα, ενώ στο κάτω μέρος υπάρχει σύνδεσμος προς τη σελίδα σύνδεσης για χρήστες που διαθέτουν ήδη λογαριασμό.

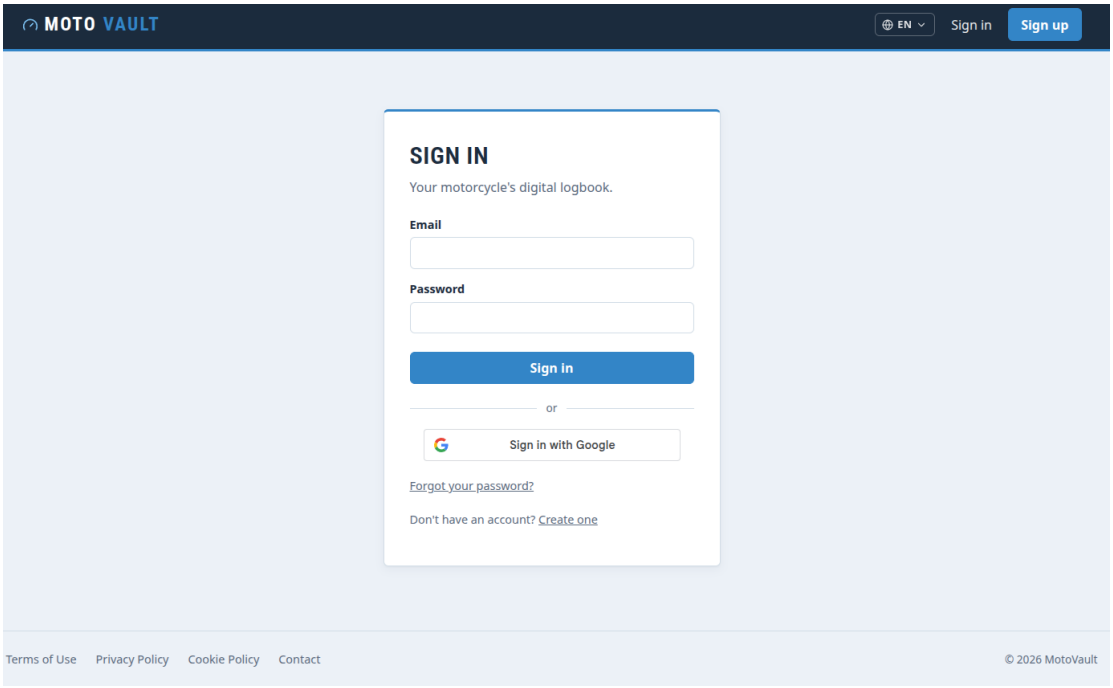
**Εικόνα 7.1**: Σελίδα δημιουργίας λογαριασμού στο Moto-Vault




### 7.2.2 Σύνδεση στην εφαρμογή

Η σελίδα σύνδεσης περιλαμβάνει πεδία email και κωδικού, επιλογή Google Sign-In και συνδέσμους για ανάκτηση κωδικού και δημιουργία νέου λογαριασμού.
Οι σελίδες εγγραφής και σύνδεσης ακολουθούν κοινή οπτική μορφή, ώστε η μετάβαση ανάμεσά τους να είναι προβλέψιμη.

**Εικόνα 7.2**: Σελίδα σύνδεσης στο Moto-Vault



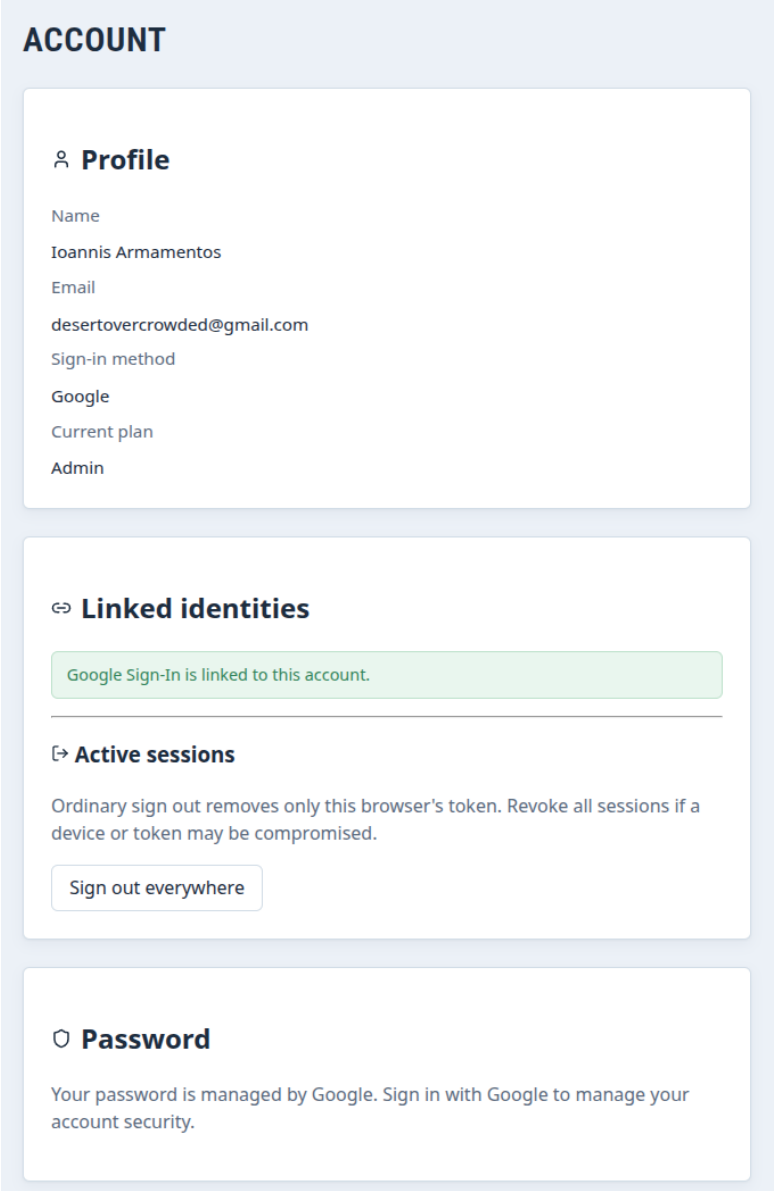
### 7.2.3 Προφίλ, ταυτότητες και συνεδρίες

Μετά τη σύνδεση, η σελίδα λογαριασμού παρουσιάζει το ονοματεπώνυμο, το email, τη μέθοδο σύνδεσης και το ενεργό πλάνο.

Στην ίδια ενότητα εμφανίζονται οι συνδεδεμένοι πάροχοι ταυτοποίησης (OAuth). Αν ο χρήστης συνδέεται μέσω Google Sign In, η εφαρμογή επισημαίνει ότι ο λογαριασμός Google είναι ήδη συνδεδεμένος.

Ο χρήστης μπορεί να ανακαλέσει τις ενεργές συνεδρίες (active sessions) και να αποσυνδεθεί από όλες τις συσκευές. Η ενότητα διαχείρισης κωδικού προσαρμόζεται ανάλογα με τη μέθοδο σύνδεσης, ενημερώνοντας τον χρήστη αν η αλλαγή κωδικού πρέπει να γίνει μέσω του εξωτερικού παρόχου.

**Εικόνα 7.3**: Προφίλ, συνδεδεμένες ταυτότητες και ενεργές συνεδρίες



### 7.2.4 Προτιμήσεις και χρήση αποθηκευτικού χώρου

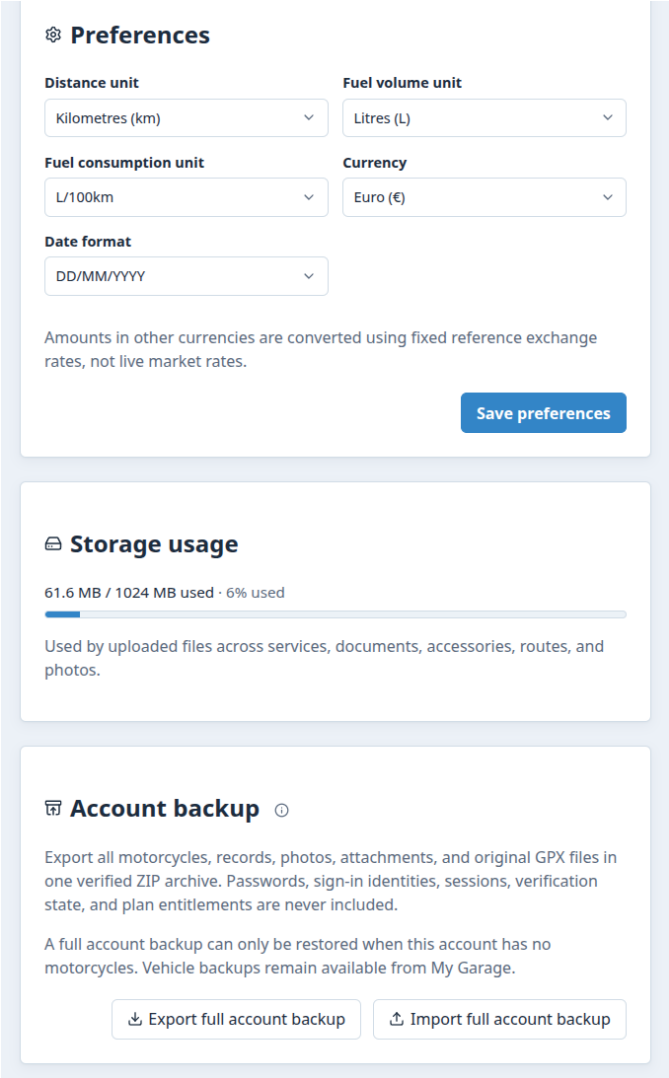
Η σελίδα λογαριασμού επιτρέπει την επιλογή:

- μονάδας απόστασης
- ονάδας όγκου καυσίμου
- μονάδας κατανάλωσης
- νομίσματος
- μορφής ημερομηνίας

Οι επιλογές αυτές επηρεάζουν την παρουσίαση των σχετικών δεδομένων σε όλη την εφαρμογή. Η συνολική προσαρμογή μονάδων και γλώσσας παρουσιάζεται στην Ενότητα 7.12.

Στην ίδια οθόνη εμφανίζεται η χρήση αποθηκευτικού χώρου με αριθμητική ένδειξη και γραμμή προόδου. Ο χρήστης μπορεί έτσι να βλέπει πόσο από το διαθέσιμο όριο χρησιμοποιείται από φωτογραφίες, έγγραφα, συνημμένα και αρχεία GPX. Παρέχονται ακόμη ενέργειες δημιουργίας και επαναφοράς πλήρους αντιγράφου ασφαλείας. Η λειτουργία παρουσιάζεται αναλυτικότερα στην Ενότητα 7.11.

**Εικόνα 7.4**: Προτιμήσεις, χρήση αποθηκευτικού χώρου και διαχείριση αντιγράφου ασφαλείας

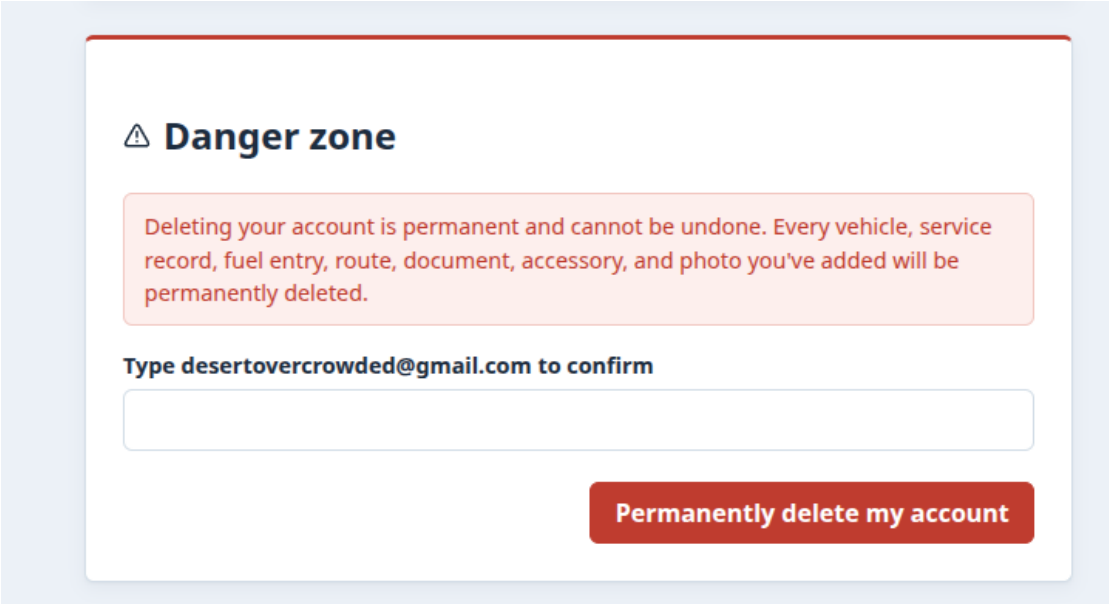


### 7.2.5 Διαγραφή λογαριασμού

Η επιλογή οριστικής διαγραφής βρίσκεται σε διακριτή "Danger Zone" (ενότητα υψηλού κινδύνου). Η εφαρμογή ενημερώνει ξεκάθαρα τον χρήστη ότι η ενέργεια αυτή θα διαγράψει οριστικά τον λογαριασμό, τις μοτοσυκλέτες, τις εγγραφές και όλα τα σχετικά αρχεία.

Για την αποφυγή τυχαίας απώλειας δεδομένων, η διαγραφή απαιτεί έξτρα βήμα επιβεβαίωσης.

**Εικόνα 7.5**: Διαγραφή λογαριασμού χρήστη




## 7.3 Αρχική σελίδα και πίνακας ελέγχου

Μετά τη σύνδεση, ο χρήστης μεταφέρεται στην αρχική σελίδα, η οποία λειτουργεί ως ψηφιακό γκαράζ και ως συγκεντρωτικός πίνακας οικονομικών στοιχείων.

### 7.3.1 Ψηφιακό γκαράζ

Οι μοτοσυκλέτες εμφανίζονται σε μορφή καρτών. Κάθε κάρτα περιλαμβάνει:

- κύρια φωτογραφία
- κατασκευαστή και μοντέλο
- έτος
- κυβισμό
- αριθμό κυκλοφορίας
- τρέχουσα χιλιομετρική ένδειξη

Η κύρια μοτοσυκλέτα επισημαίνεται με ειδική ένδειξη (main). Από το επάνω μέρος της σελίδας ο χρήστης μπορεί να δημιουργήσει νέο όχημα ή να εισαγάγει archive οχήματος.

### 7.3.2 Άμεση πρόσβαση στις λειτουργίες

Κάθε κάρτα παρέχει συντομεύσεις προς:

- επισκόπηση
- συντήρηση
- καύσιμα
- έγγραφα
- υπενθυμίσεις
- διαδρομές
- αξεσουάρ
- ανταλλακτικά
- φωτογραφίες

Η πλοήγηση ξεκινά επομένως από τη μοτοσυκλέτα που ενδιαφέρει τον χρήστη και δεν απαιτεί ενδιάμεση αναζήτηση.

### 7.3.3 Οικονομικά στοιχεία γκαράζ

Κάτω από τις κάρτες εμφανίζονται συνοπτικοί δείκτες για:

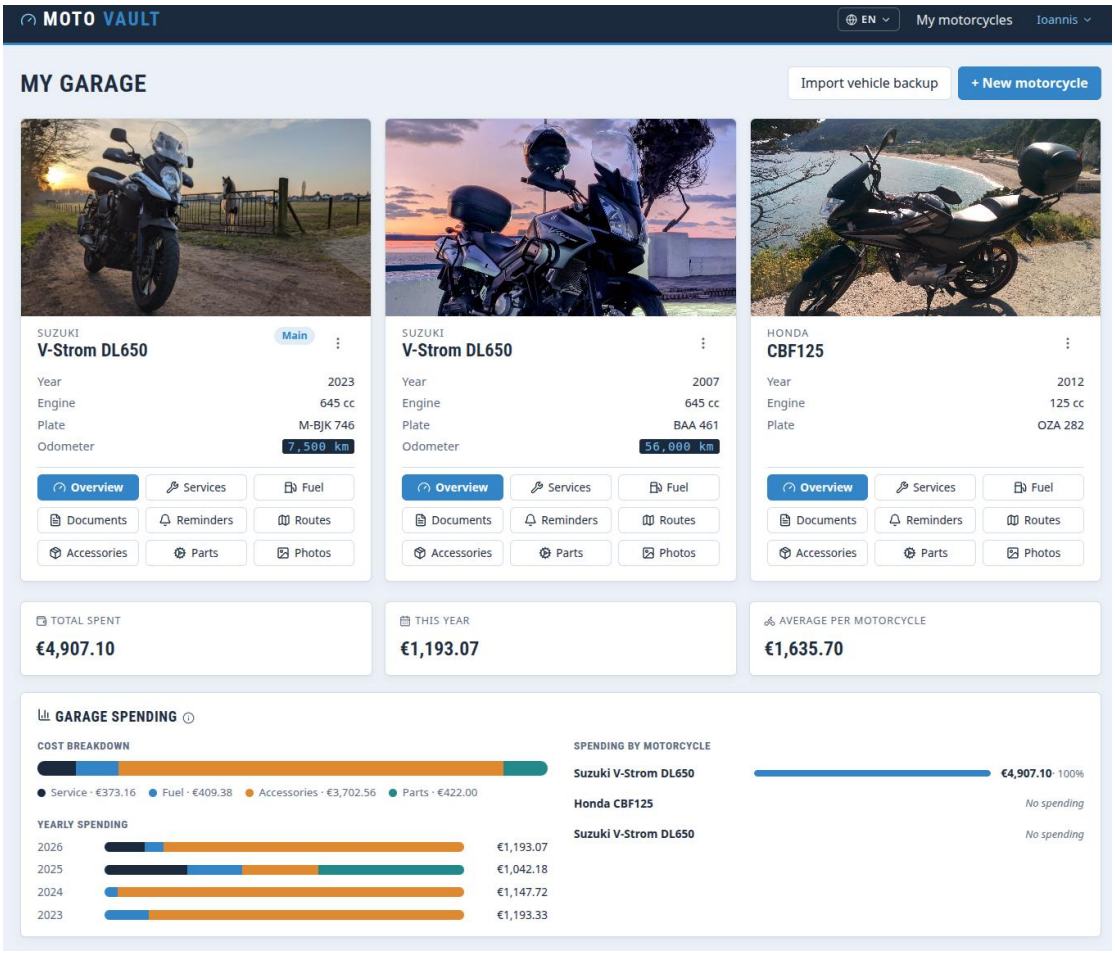
- συνολικό ποσό δαπανών
- δαπάνες του τρέχοντος έτους
- μέση δαπάνη ανά μοτοσυκλέτα

Η περιοχή «Garage spending» αναλύει τις δαπάνες:

- ανά κατηγορία
- ανά έτος
- ανά μοτοσυκλέτα

Έτσι η αρχική σελίδα συνδυάζει την πρόσβαση στα οχήματα με μια συγκεντρωτική εικόνα του κόστους ολόκληρου του γκαράζ.

**Εικόνα 7.6**: Αρχική σελίδα του Moto-Vault με το ψηφιακό γκαράζ και τη συγκεντρωτική ανάλυση δαπανών




## 7.4 Διαχείριση μοτοσυκλετών

Η μοτοσυκλέτα αποτελεί την κεντρική οντότητα της εφαρμογής, καθώς όλες οι λειτουργικές εγγραφές οργανώνονται γύρω από συγκεκριμένο όχημα.

### 7.4.1 Δημιουργία και επεξεργασία οχήματος

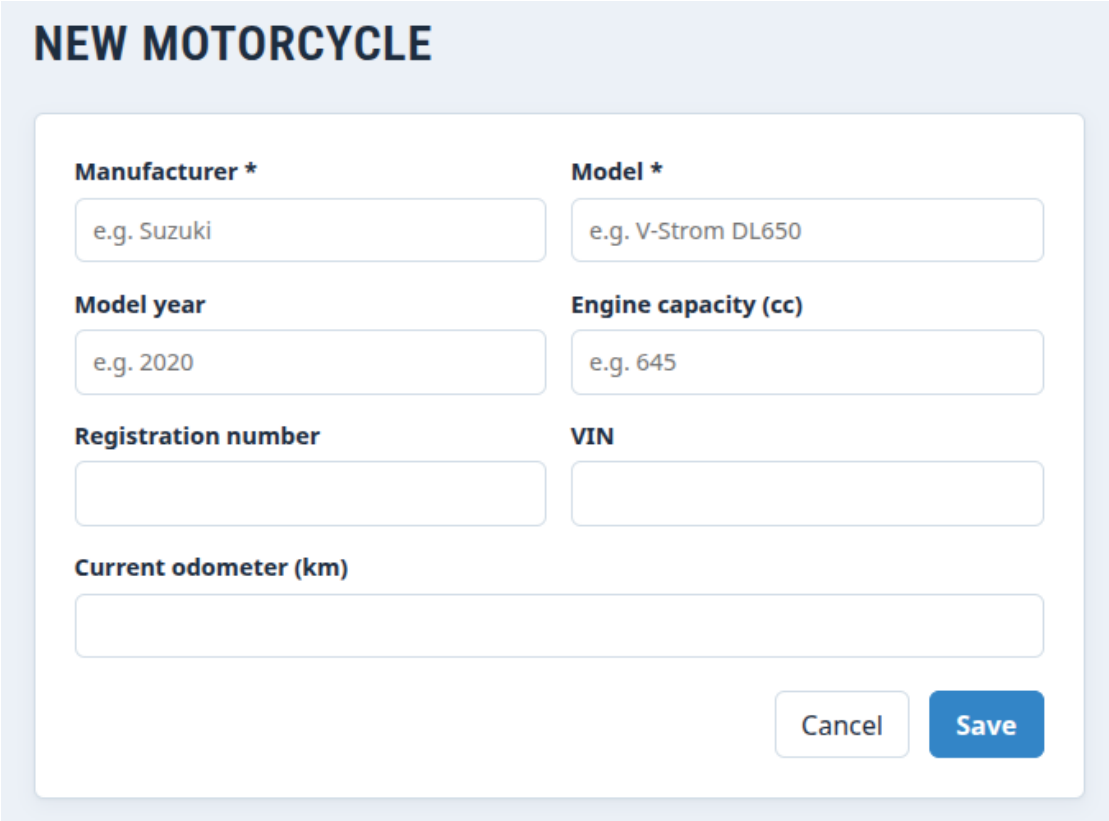
Η φόρμα νέας μοτοσυκλέτας περιλαμβάνει:

- κατασκευαστή
- μοντέλο
- έτος
- κυβισμό
- αριθμό κυκλοφορίας
- αριθμό πλαισίου
- τρέχουσα ένδειξη χιλιομετρητή

Τα υποχρεωτικά πεδία επισημαίνονται και οι ενέργειες αποθήκευσης και ακύρωσης βρίσκονται στο τέλος της φόρμας.

Η αρχική χιλιομετρική ένδειξη χρησιμοποιείται αργότερα σε συντηρήσεις, ανεφοδιασμούς, υπενθυμίσεις και οικονομικούς δείκτες.

**Εικόνα 7.7**: Φόρμα δημιουργίας νέας μοτοσυκλέτας




### 7.4.2 Κοινό πλαίσιο και πλοήγηση

Μετά την επιλογή οχήματος εμφανίζεται η sidebar στο αριστερά με τη φωτογραφία και τα βασικά στοιχεία της μοτοσυκλέτας.

Το μενού παρέχει πρόσβαση σε όλες τις λειτουργικές περιοχές:

- επισκόπηση
- συντηρήσεις
- καύσιμα
- έγγραφα
- υπενθυμίσεις
- διαδρομές
- αξεσουάρ
- ανταλλακτικά
- φωτογραφίες

Η ίδια δομή παραμένει διαθέσιμη καθώς ο χρήστης μετακινείται από ενότητα σε ενότητα.

### 7.4.3 Επισκόπηση οχήματος

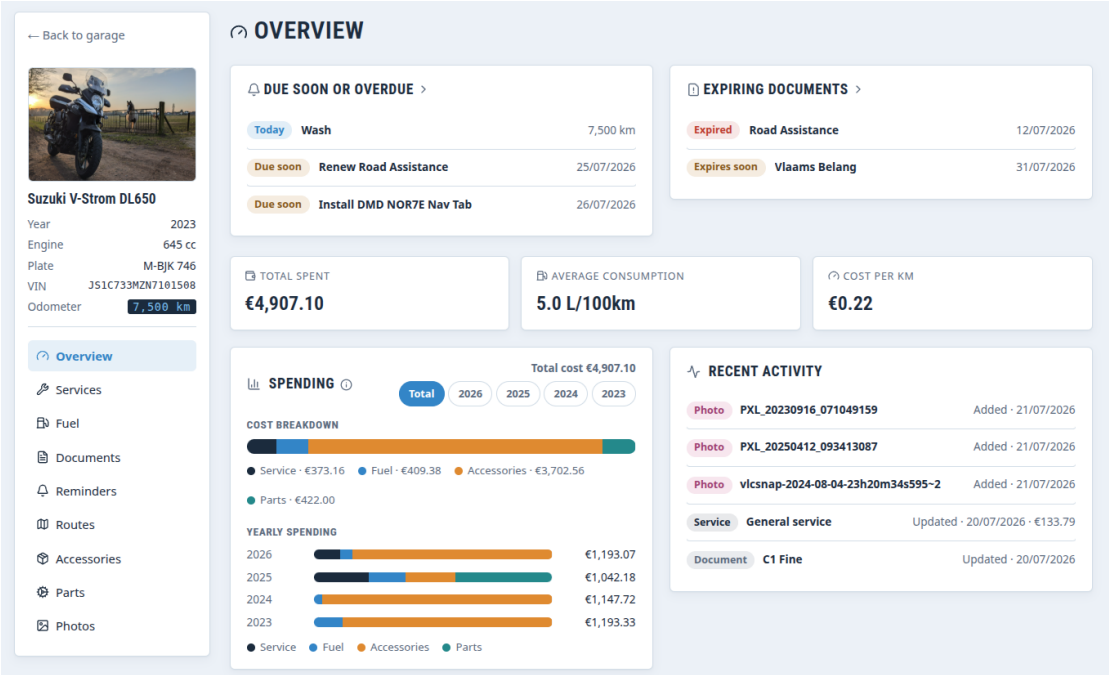
Η σελίδα επισκόπησης συγκεντρώνει:

- επερχόμενες ή εκπρόθεσμες υπενθυμίσεις
- έγγραφα που λήγουν
- συνολικές δαπάνες
- μέση κατανάλωση
- κόστος ανά χιλιόμετρο
- κατανομή κόστους
- πρόσφατη δραστηριότητα

Οι χρήσιμες πληροφορίες τοποθετούνται ψηλότερα, ενώ τα οικονομικά στοιχεία και η δραστηριότητα ακολουθούν σε ξεχωριστές κάρτες.

Η επιλογή μιας υπενθύμισης, ενός εγγράφου ή άλλου συνδεδεμένου στοιχείου μπορεί να μεταφέρει τον χρήστη στην αντίστοιχη αναλυτική ενότητα.

**Εικόνα 7.8**: Σελίδα επισκόπησης μοτοσυκλέτας




## 7.5 Ιστορικό συντήρησης

Η ενότητα συντήρησης συγκεντρώνει τις εργασίες που έχουν πραγματοποιηθεί για μια συγκεκριμένη μοτοσυκλέτα και τις παρουσιάζει μαζί με ημερομηνίες, χιλιόμετρα, κόστος και σχετικά αρχεία.

### 7.5.1 Λίστα, αναζήτηση και κατηγορίες

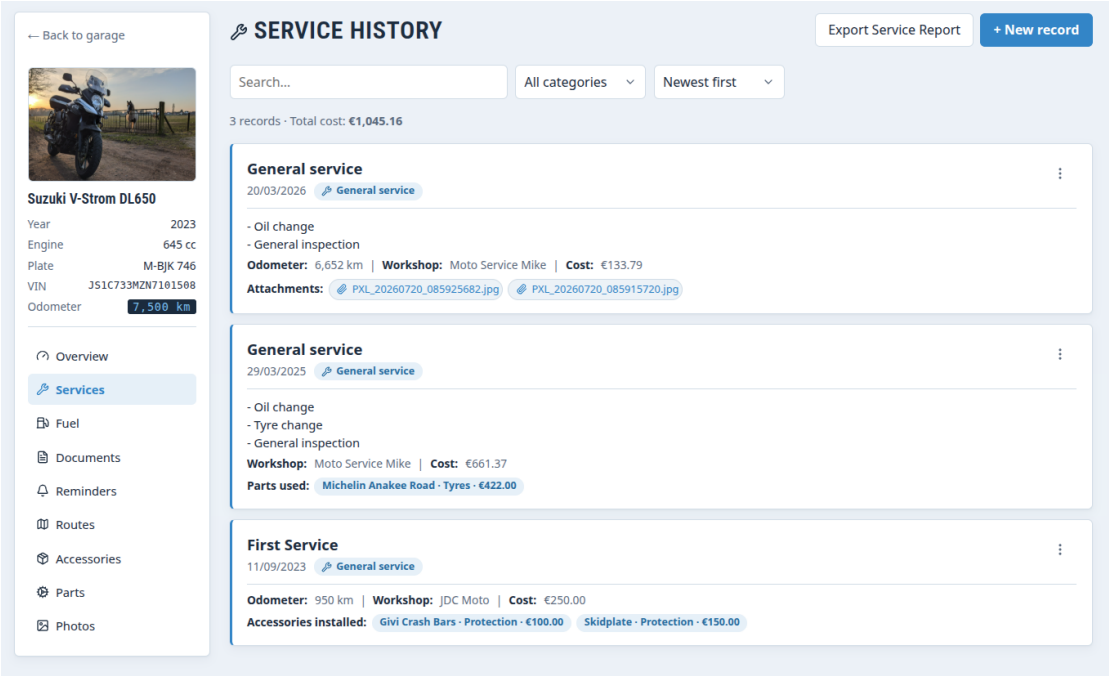
Οι εγγραφές εμφανίζονται σε μορφή καρτών. Κάθε κάρτα μπορεί να περιλαμβάνει:

- τίτλο
- ημερομηνία
- κατηγορία
- περιγραφή εργασιών
- ένδειξη χιλιομετρητή
- συνεργείο
- κόστος
- συνημμένα
- συνδεδεμένα ανταλλακτικά και αξεσουάρ

Στο επάνω μέρος εμφανίζονται ο συνολικός αριθμός εγγραφών και το συνολικό κόστος συντήρησης.

Η σελίδα παρέχει αναζήτηση, φίλτρο κατηγορίας και επιλογές ταξινόμησης. Οι κατηγορίες εμφανίζονται με συγκεκριμένες ετικέτες και βοηθούν στην οργάνωση μεγάλου ιστορικού.

**Εικόνα 7.9**: Σελίδα ιστορικού συντήρησης



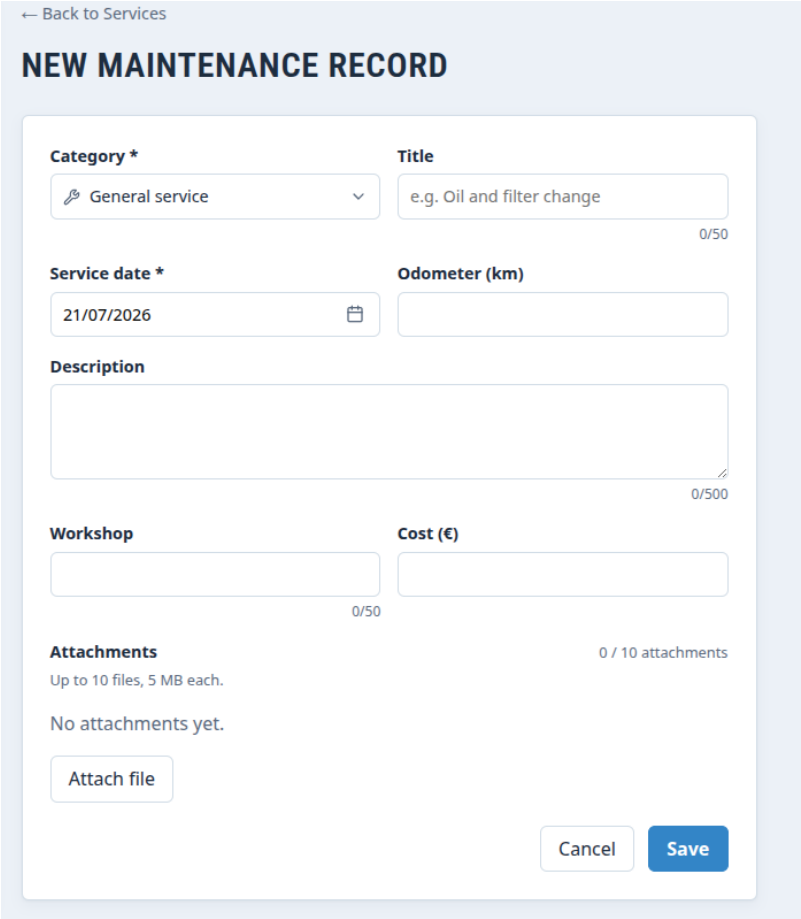

### 7.5.2 Δημιουργία εγγραφής

Η φόρμα νέου σέρβις περιλαμβάνει:

- κατηγορία,
- τίτλο
- ημερομηνία service
- χιλιομετρική ένδειξη
- περιγραφή
- συνεργείο
- κόστος
- συνημμένα αρχεία

Τα βασικά πεδία επισημαίνονται ως υποχρεωτικά, ενώ τα υπόλοιπα επιτρέπουν στον χρήστη να επιλέξει το επίπεδο λεπτομέρειας της καταχώρισης.

**Εικόνα 7.10**: Φόρμα δημιουργίας νέας εγγραφής συντήρησης




### 7.5.3 Συνημμένα και συνδεδεμένα στοιχεία

Κάθε εγγραφή μπορεί να συνοδεύεται από ορισμένο αριθμό αρχείων, με συγκεκριμένα όρια μεγέθους. Τα συνημμένα μπορούν να είναι αποδείξεις, τιμολόγια, φωτογραφίες εργασιών, διαγνωστικά έγγραφα ή δελτία συνεργείου.

Μια εργασία μπορεί επίσης να συνδέεται με ανταλλακτικά ή αξεσουάρ που χρησιμοποιήθηκαν ή τοποθετήθηκαν. Τα στοιχεία εμφανίζονται μέσα στην κάρτα ως σύνδεσμοι και οδηγούν στην αντίστοιχη ενότητα, όπου η σχετική εγγραφή μπορεί να εντοπιστεί και να επισημανθεί.

### 7.5.4 Χιλιόμετρα και αναφορά PDF

Η χιλιομετρική ένδειξη συνδέει την εργασία με την πραγματική χρήση του οχήματος. Όταν η νέα τιμή είναι μεγαλύτερη από την τρέχουσα ένδειξη της μοτοσυκλέτας, μπορεί να ενημερωθεί και ο χιλιομετρητής αυτόματα.

Από την ενότητα μπορεί επίσης να δημιουργηθεί αναφορά ιστορικού συντήρησης σε PDF. Η λειτουργία παρουσιάζεται αναλυτικά στην Ενότητα 7.11.

## 7.6 Καταγραφή καυσίμων και στατιστικά κατανάλωσης

Η ενότητα καυσίμων συγκεντρώνει το ιστορικό ανεφοδιασμών, προβάλλοντας παράλληλα συγκεντρωτικά στοιχεία για τη μέση κατανάλωση και το συνολικό κόστος.

### 7.6.1 Στατιστικά κατανάλωσης και κόστους

Στο επάνω μέρος της σελίδας εμφανίζονται:

- μέση κατανάλωση
- συνολικό κόστος καυσίμων
- κόστος ανά χιλιόμετρο
- συνολική ποσότητα καυσίμου

Παρέχεται επίσης ετήσια ανάλυση κόστους, ώστε ο χρήστης να μπορεί να συγκρίνει διαφορετικές περιόδους.

Οι τιμές εμφανίζονται μόνο όταν υπάρχουν επαρκή δεδομένα και προσαρμόζονται στις μονάδες και στο νόμισμα που έχει επιλέξει ο χρήστης.

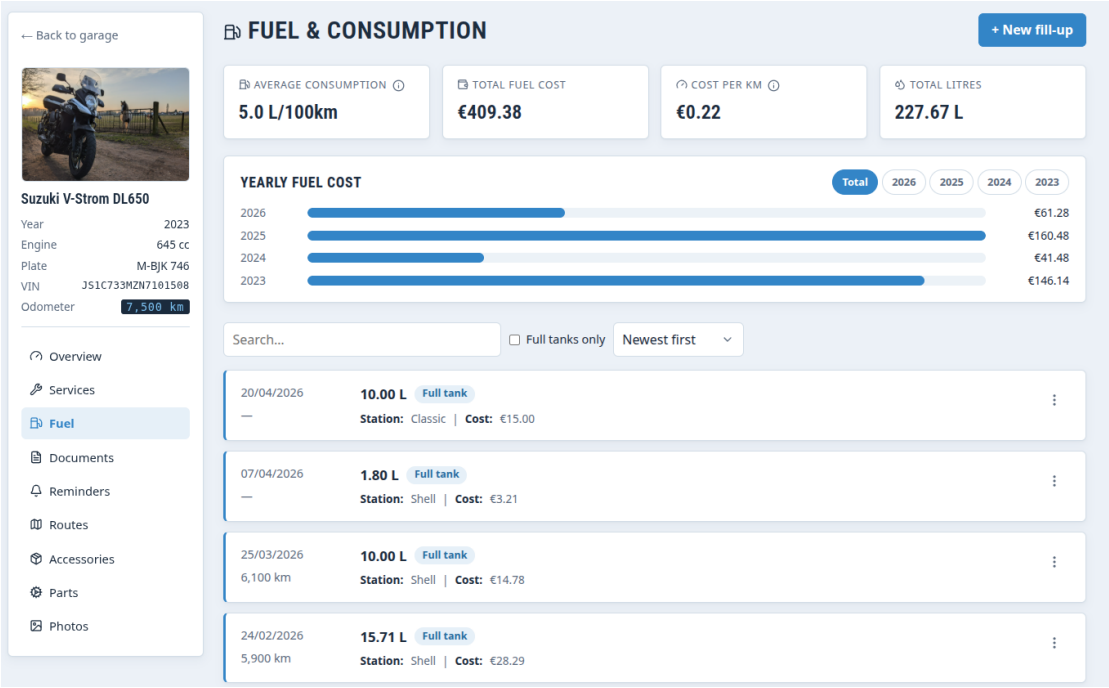
### 7.6.2 Ιστορικό ανεφοδιασμών

Κάθε ανεφοδιασμός παρουσιάζεται σε κάρτα και περιλαμβάνει:

- ημερομηνία
- χιλιομετρική ένδειξη
- ποσότητα καυσίμου
- ένδειξη πλήρους γεμίσματος
- πρατήριο
- συνολικό κόστος

Η λίστα υποστηρίζει αναζήτηση, χρονολογική ταξινόμηση και φίλτρο πλήρων γεμισμάτων.

**Εικόνα 7.11**: Σελίδα καυσίμων με συνοπτικά στατιστικά και ιστορικό ανεφοδιασμών

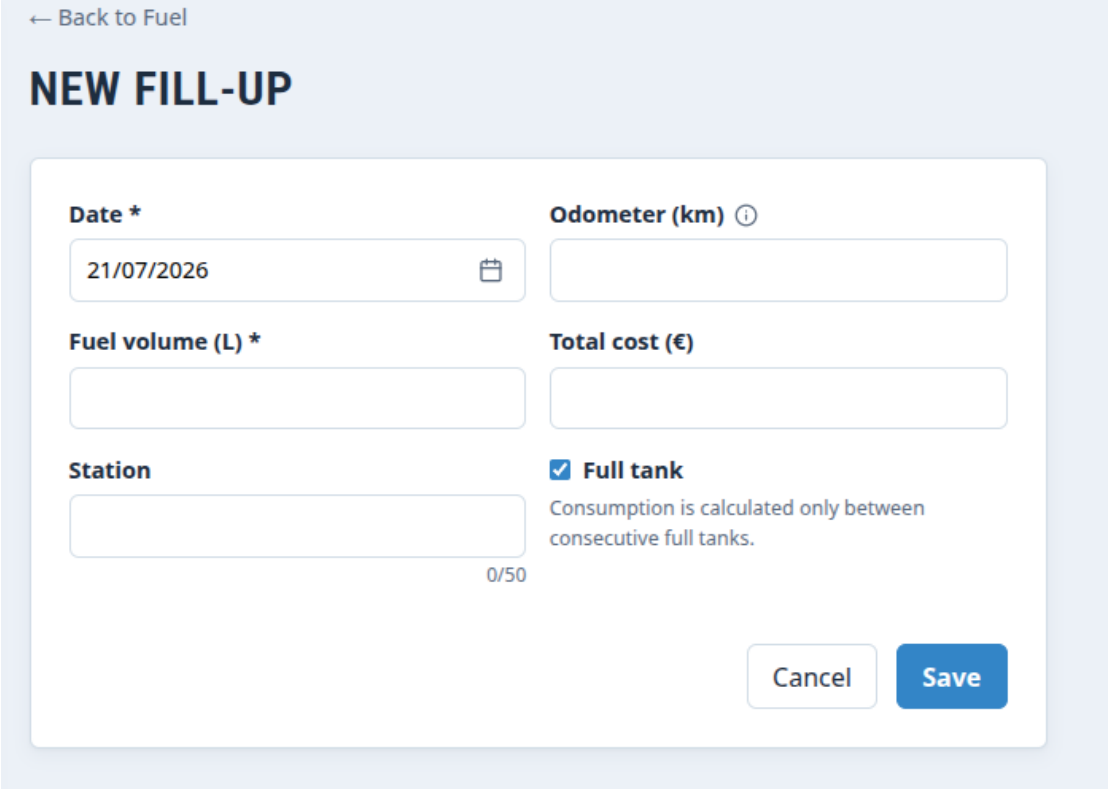


### 7.6.3 Δημιουργία ανεφοδιασμού

Η φόρμα καταχώρισης περιλαμβάνει ημερομηνία, χιλιόμετρα, ποσότητα, κόστος, πρατήριο και ένδειξη πλήρους γεμίσματος.

Ο περιορισμένος αριθμός πεδίων επιτρέπει γρήγορη χρήση και από κινητή συσκευή αμέσως μετά τον ανεφοδιασμό.

**Εικόνα 7.12**: Φόρμα δημιουργίας νέου ανεφοδιασμού




### 7.6.4 Πλήρη και μερικά γεμίσματα

Για τον ακριβή υπολογισμό της κατανάλωσης, η εφαρμογή βασίζεται στις εγγραφές πλήρους ανεφοδιασμού (full tank), συνυπολογίζοντας και τα ενδιάμεσα μερικά γεμίσματα στο ίδιο διάστημα.

Αν τα διαθέσιμα δεδομένα δεν επαρκούν για να βγει ασφαλές συμπέρασμα, η εφαρμογή αποφεύγει να προβάλει λανθασμένες ή παραπλανητικές τιμές.

Τέλος, τα έξοδα καυσίμου ενσωματώνονται τόσο στην επισκόπηση της κάθε μοτοσυκλέτας όσο και στη συνολική ανάλυση δαπανών του γκαράζ.

## 7.7 Αξεσουάρ, ανταλλακτικά και έγγραφα

Οι τρεις ενότητες ακολουθούν παρόμοια λογική πλοήγησης και διαχείρισης, αλλά καλύπτουν διαφορετικές κατηγορίες πληροφοριών.

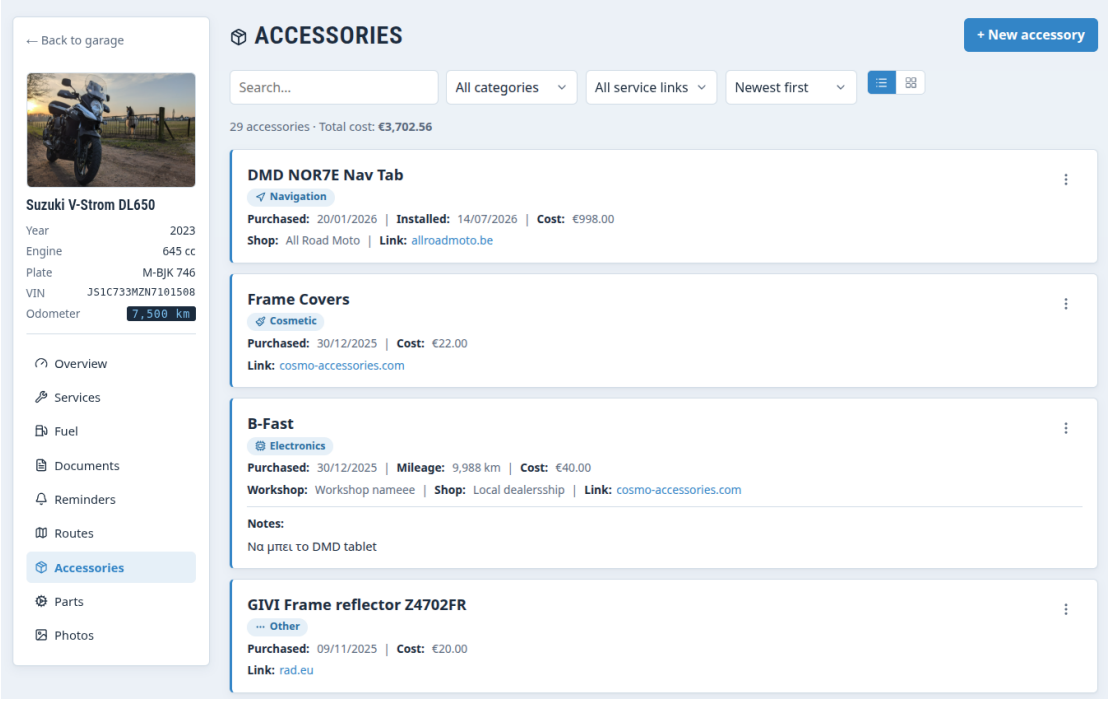
### 7.7.1 Αξεσουάρ

Η σελίδα αξεσουάρ παρουσιάζει τον αριθμό καταχωρίσεων και το συνολικό κόστος. Κάθε εγγραφή μπορεί να περιλαμβάνει:

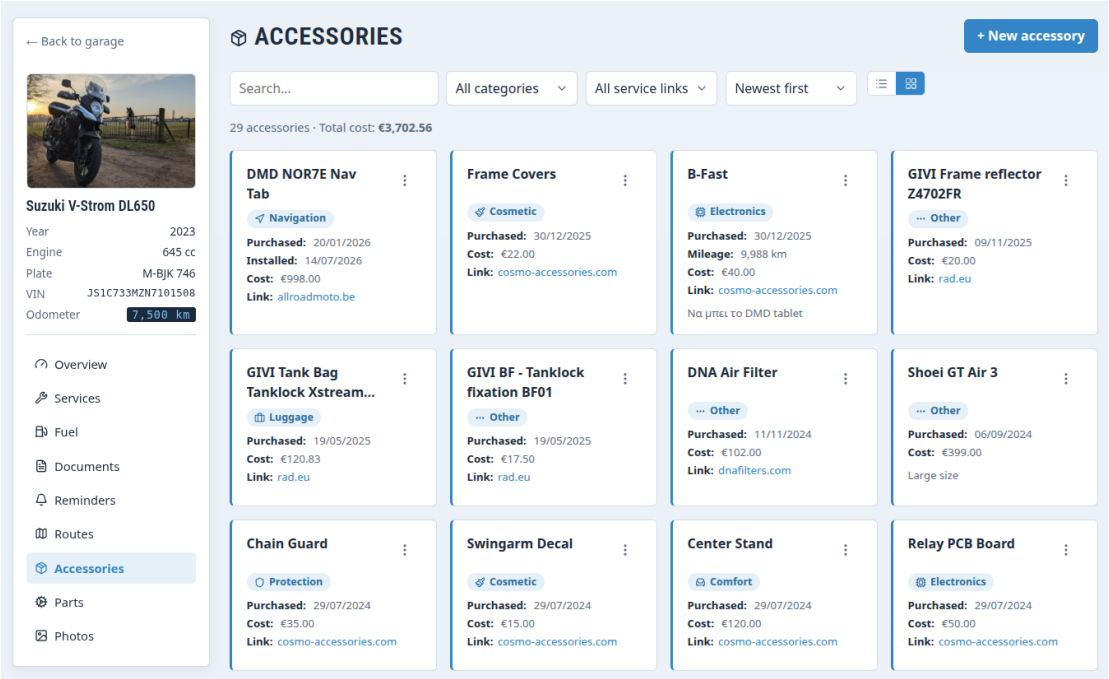
- μάρκα και όνομα
- κατηγορία
- ημερομηνία αγοράς και εγκατάστασης
- χιλιομετρική ένδειξη εγκατάστασης
- κόστος
- συνεργείο ή κατάστημα
- εξωτερικό σύνδεσμο
- σημειώσεις
- συνημμένα

Παρέχονται αναζήτηση, φίλτρα, ταξινόμηση και εναλλαγή ανάμεσα σε προβολή λίστας και πλέγματος.

**Εικόνα 7.13**: Λίστα αξεσουάρ σε μορφή λίστας

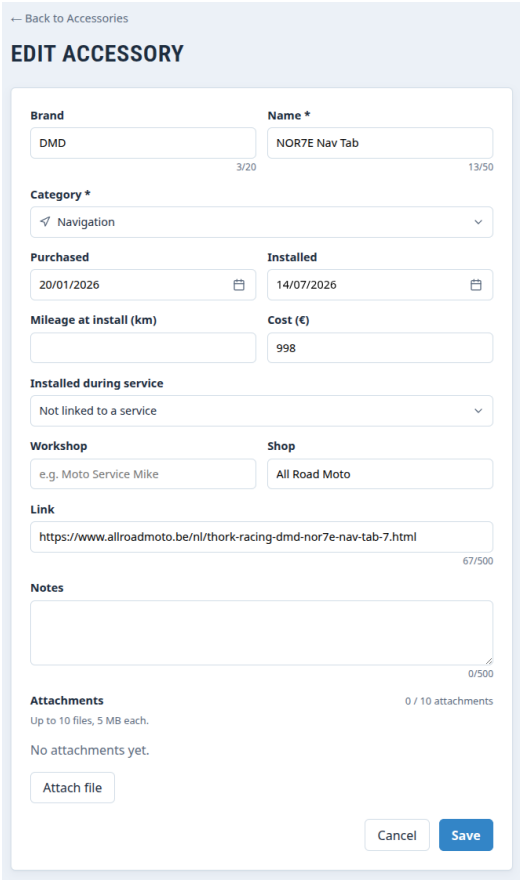


**Εικόνα 7.14**: Λίστα αξεσουάρ σε μορφή πλέγματος




Η φόρμα επεξεργασίας περιλαμβάνει τα βασικά και προαιρετικά πεδία, τα σχετικά αρχεία και δυνατότητα σύνδεσης με συγκεκριμένη εργασία συντήρησης.

**Εικόνα 7.15**: Φόρμα επεξεργασίας αξεσουάρ



### 7.7.2 Ανταλλακτικά

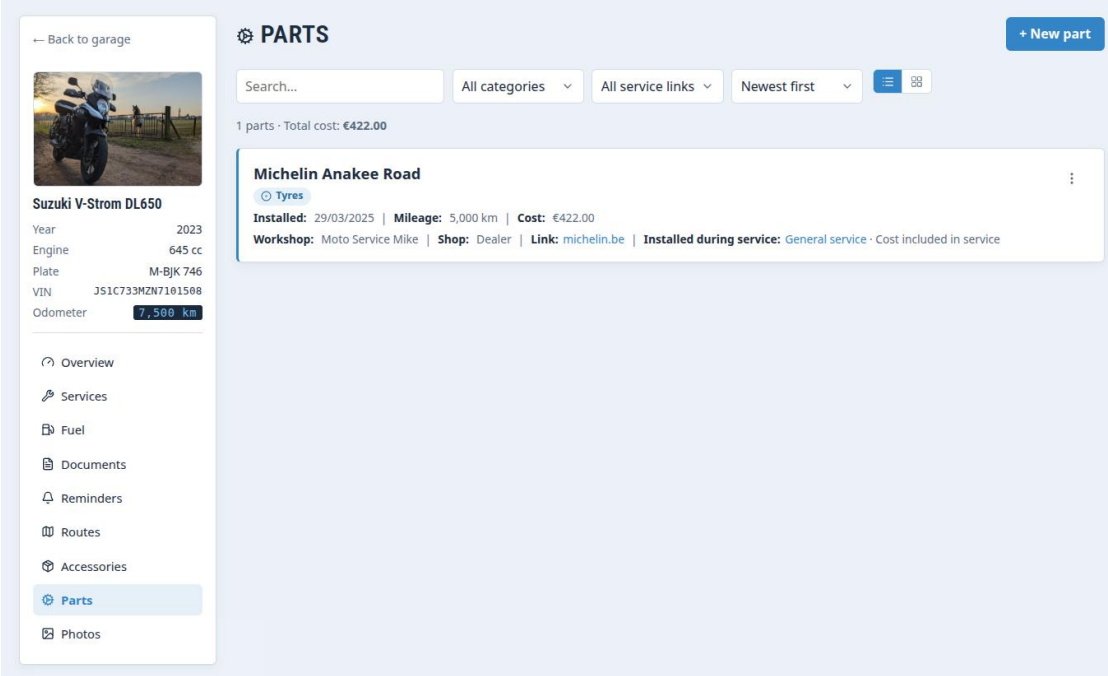
Η ενότητα ανταλλακτικών ακολουθεί παρόμοια δομή, αλλά εστιάζει σε εξαρτήματα και αναλώσιμα που σχετίζονται με την τεχνική κατάσταση της μοτοσυκλέτας.

Κάθε εγγραφή μπορεί να περιλαμβάνει:

- όνομα και κατηγορία
- ημερομηνία εγκατάστασης
- χιλιομετρική ένδειξη
- κόστος
- συνεργείο ή κατάστημα
- σύνδεσμο αγοράς
- συνημμένα
- σύνδεση με εργασία συντήρησης

Η σύνδεση με service επιτρέπει στον χρήστη να γνωρίζει σε ποια εργασία χρησιμοποιήθηκε κάθε ανταλλακτικό και να μετακινείται ανάμεσα στις δύο ενότητες.

**Εικόνα 7.16**: Λίστα ανταλλακτικών




### 7.7.3 Έγγραφα

Η ενότητα εγγράφων οργανώνει διοικητικά, ασφαλιστικά και τεχνικά αρχεία. Κάθε καταχώριση μπορεί να περιλαμβάνει:

- τίτλο
- τύπο
- ημερομηνία έκδοσης
- ημερομηνία λήξης
- σημειώσεις
- συνημμένο αρχείο

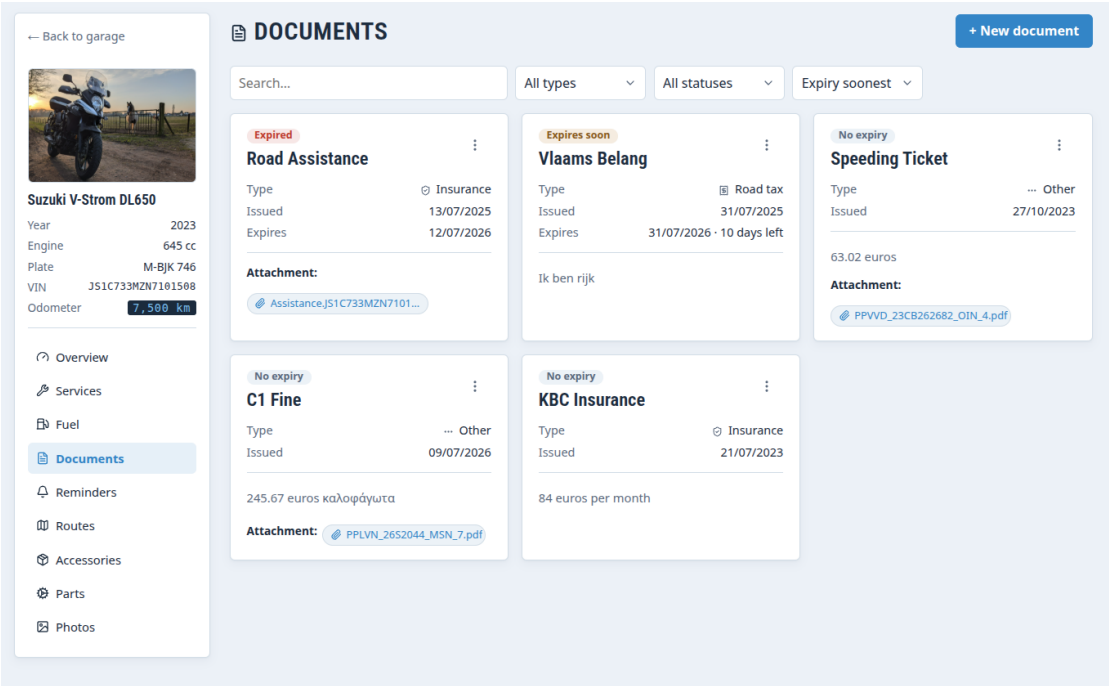
Η σελίδα υποστηρίζει αναζήτηση, φίλτρα τύπου και κατάστασης και ταξινόμηση.

Η κατάσταση ενός εγγράφου εμφανίζεται ως:

- χωρίς λήξη
- σε ισχύ
- λήγει σύντομα
- ληγμένο

Οι ενδείξεις αυτές βοηθούν να ξεχωρίζουν τα έγγραφα που απαιτούν προσοχή και εμφανίζονται επίσης στην επισκόπηση οχήματος.

**Εικόνα 7.17**: Λίστα εγγράφων




## 7.7.4 Κοινά χαρακτηριστικά

Και οι τρεις περιοχές παρέχουν:

- δημιουργία και επεξεργασία
- αναζήτηση και φίλτρα
- καταγραφή κόστους
- συνημμένα
- σύνδεση με συγκεκριμένη μοτοσυκλέτα
- προστατευμένη λήψη αρχείων

Η κοινή οργάνωση επιτρέπει στον χρήστη να μεταφέρεται από τη μία κατηγορία στην άλλη χωρίς να χρειάζεται να μάθει διαφορετικό τρόπο χρήσης.

## 7.8 Υπενθυμίσεις

Η ενότητα υπενθυμίσεων επιτρέπει τον προγραμματισμό μελλοντικών ενεργειών βάσει ημερομηνίας, χιλιομέτρων ή συνδυασμού των δύο.

### 7.8.1 Λίστα και καταστάσεις

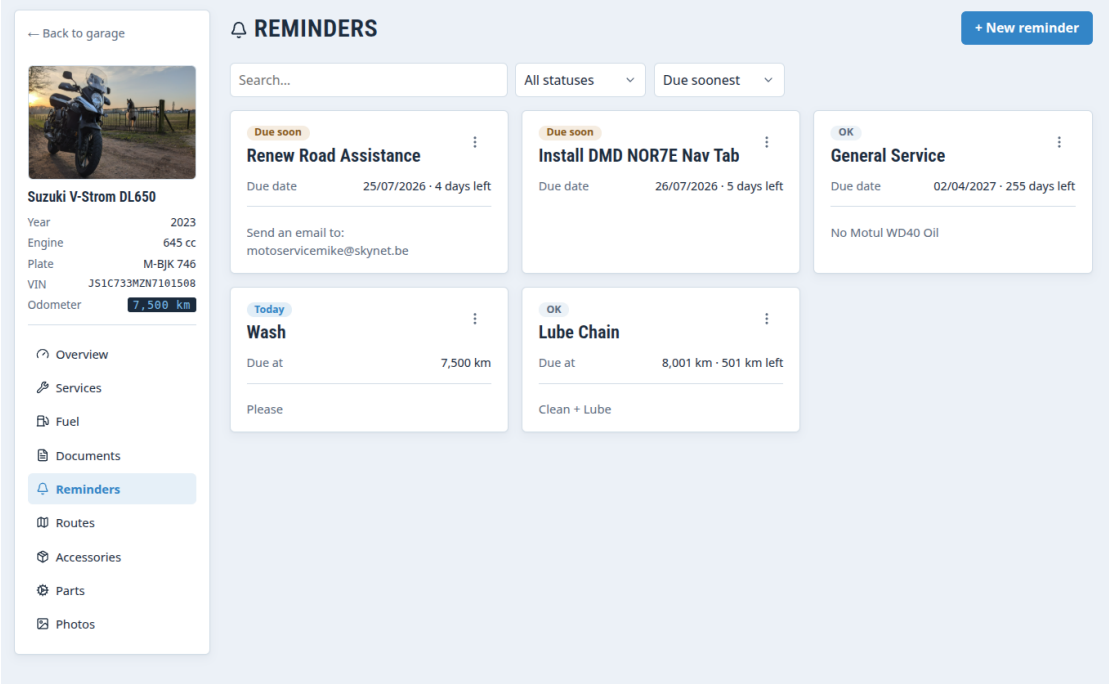
Οι υπενθυμίσεις εμφανίζονται σε κάρτες με:

- τίτλο
- ημερομηνία λήξης
- χιλιομετρικό όριο
- υπολειπόμενες ημέρες
- υπολειπόμενα χιλιόμετρα
- σημειώσεις
- κατάσταση

Τα labels ξεχωρίζουν τις υπενθυμίσεις που λήγουν σήμερα, πλησιάζουν το όριο ή παραμένουν εντός χρονικού και χιλιομετρικού ορίου.

Η σελίδα διαθέτει αναζήτηση, φίλτρο κατάστασης και ταξινόμηση με βάση την εγγύτερη προθεσμία.

**Σχήμα 7.18**: Σελίδα υπενθυμίσεων




### 7.8.2 Τύποι υπενθυμίσεων

Μια υπενθύμιση μπορεί να βασίζεται:

- σε ημερομηνία
- σε χιλιομετρική ένδειξη
- και στα δύο

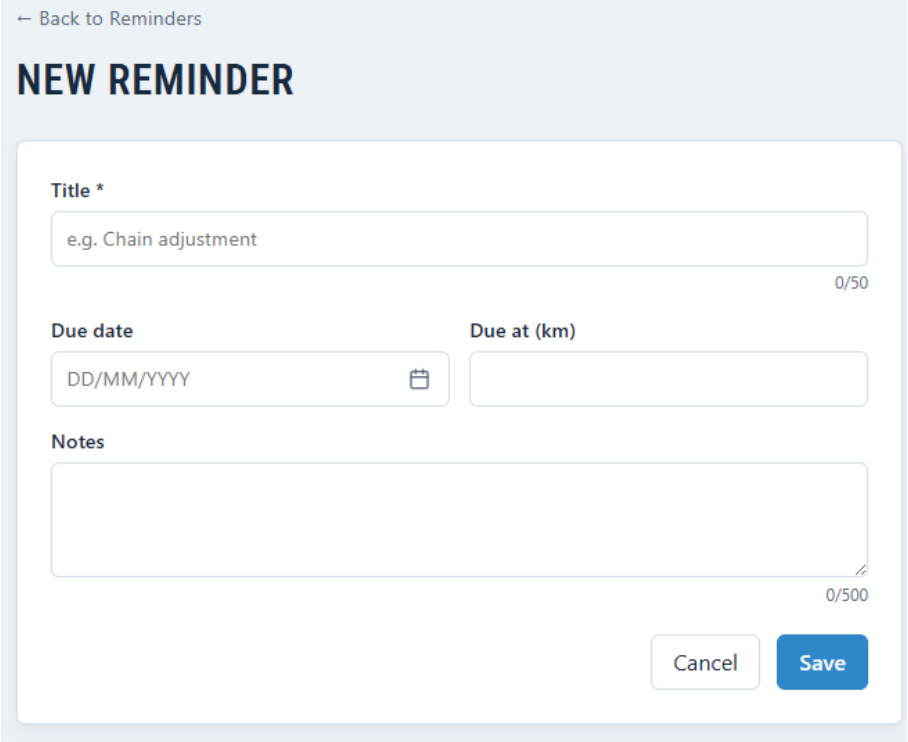
Παραδείγματα χρονικών υπενθυμίσεων είναι η ανανέωση ασφάλειας ή ο προγραμματισμός service. Παραδείγματα χιλιομετρικών υπενθυμίσεων είναι η αλλαγή λαδιών, η λίπανση αλυσίδας και ο έλεγχος ελαστικών.

Η εφαρμογή συγκρίνει τις τιμές με την τρέχουσα ημερομηνία και τον χιλιομετρητή του οχήματος.

### 7.8.3 Δημιουργία υπενθύμισης

Η φόρμα περιλαμβάνει τίτλο, ημερομηνία, χιλιομετρικό όριο και σημειώσεις. Ο χρήστης συμπληρώνει μόνο τα κριτήρια που ταιριάζουν στη συγκεκριμένη ενέργεια.

**Σχήμα 7.19**: Φόρμα δημιουργίας νέας υπενθύμισης




### 7.8.4 Σύνδεση με επισκόπηση και έγγραφα

Οι σημαντικότερες υπενθυμίσεις εμφανίζονται και στην επισκόπηση οχήματος. Με την επιλογή τους ο χρήστης μεταφέρεται στην αντίστοιχη κάρτα της αναλυτικής λίστας.

Οι υπενθυμίσεις μπορούν επίσης να χρησιμοποιούνται συμπληρωματικά με τα έγγραφα. Ένα έγγραφο αποθηκεύει την ημερομηνία λήξης του, ενώ ξεχωριστή υπενθύμιση μπορεί να αφορά την ενέργεια ανανέωσης ή πληρωμής.

Η τρέχουσα έκδοση δεν αποστέλλει email ειδοποιήσεις για υπενθυμίσεις. Η δυνατότητα αυτή αποτελεί μελλοντική επέκταση.

## 7.9 Διαδρομές GPX

Η ενότητα GPX επιτρέπει την αποθήκευση, οργάνωση και χαρτογραφική παρουσίαση διαδρομών που σχετίζονται με τη μοτοσυκλέτα.

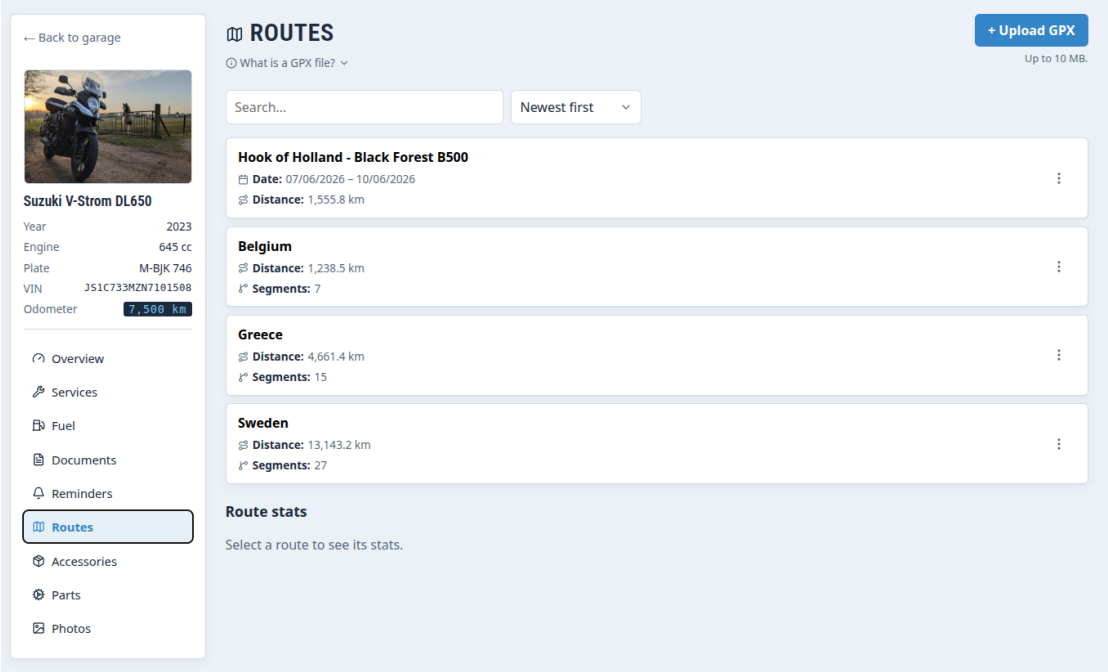
### 7.9.1 Λίστα, αναζήτηση και μεταφόρτωση

Η βασική σελίδα εμφανίζει τις αποθηκευμένες διαδρομές με:

- τίτλο
- ημερομηνία ή εύρος ημερομηνιών
- συνολική απόσταση
- αριθμό segments

Παρέχονται αναζήτηση, ταξινόμηση, ανέβασμα αρχείου και ένδειξη μέγιστου επιτρεπόμενου μεγέθους αρχείου.

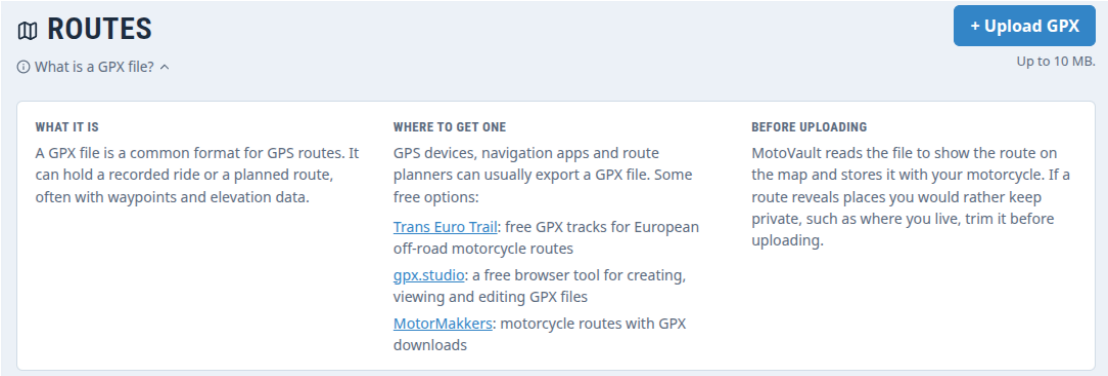
**Σχήμα 7.20**: Λίστα διαδρομών GPX




### 7.9.2 Ενημερωτικό πλαίσιο

Η σελίδα περιλαμβάνει ένα πτυσσόμενο μενού πληροφοριών (accordion) που εξηγεί τι είναι το αρχείο GPX, από πού μπορεί να εξαχθεί και τι χρειάζεται να ελέγξει ο χρήστης πριν το ανεβάσει. Διευκολύνονται έτσι οι χρήστες που δεν γνωρίζουν τη συγκεκριμένη μορφή αρχείου.

**Σχήμα 7.21**: Ενημερωτικό πλαίσιο για τη μορφή GPX




## 7.9.3 Επιλογή διαδρομής, χάρτης και στατιστικά

Όταν επιλεγεί μια διαδρομή, η εφαρμογή εμφανίζει τη γεωμετρία της σε χάρτη μέσω Leaflet και OpenStreetMap.

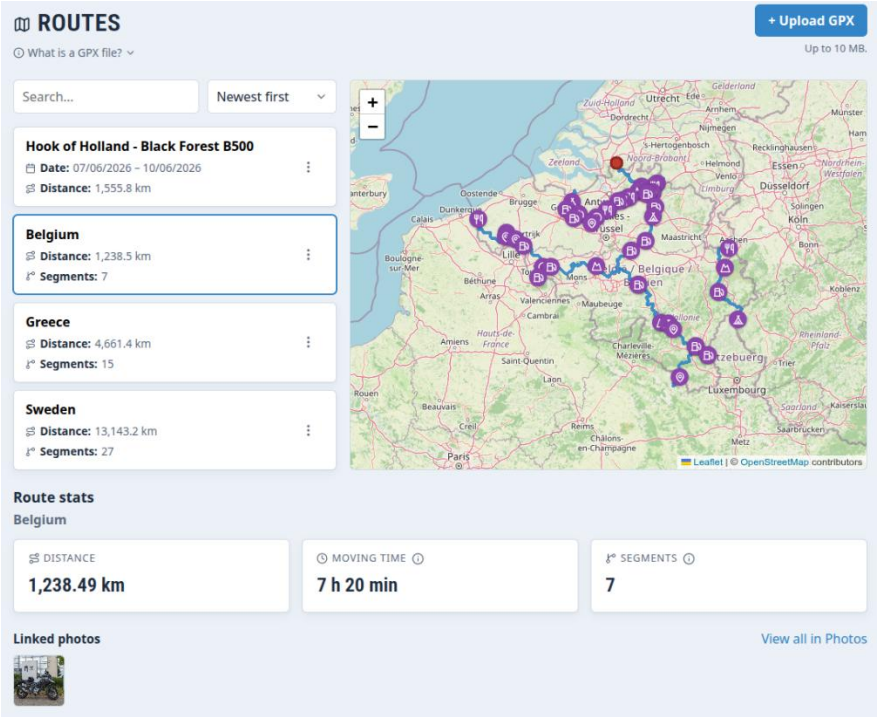
Η επιλεγμένη εγγραφή επισημαίνεται στη λίστα και ο χάρτης προσαρμόζεται στα όρια της διαδρομής. Ο χρήστης μπορεί να μετακινηθεί, να αλλάξει zoom και να εξετάσει την πορεία και τα σημεία ενδιαφέροντος (POIs).

Παρουσιάζονται επίσης βασικά στοιχεία, όπως:

- συνολική απόσταση
- χρόνος κίνησης
- αριθμός segments
- υψομετρική άνοδος και κάθοδος όπου υπάρχουν δεδομένα
- αριθμός σημείων ενδιαφέροντος

Η αλληλεπίδραση ανάμεσα στη λίστα και στον χάρτη επιτρέπει γρήγορη αλλαγή ενεργής διαδρομής χωρίς μετάβαση σε διαφορετική σελίδα.

**Σχήμα 7.22**: Προβολή επιλεγμένης διαδρομής στον χάρτη



### 7.9.4 Υψομετρικό προφίλ

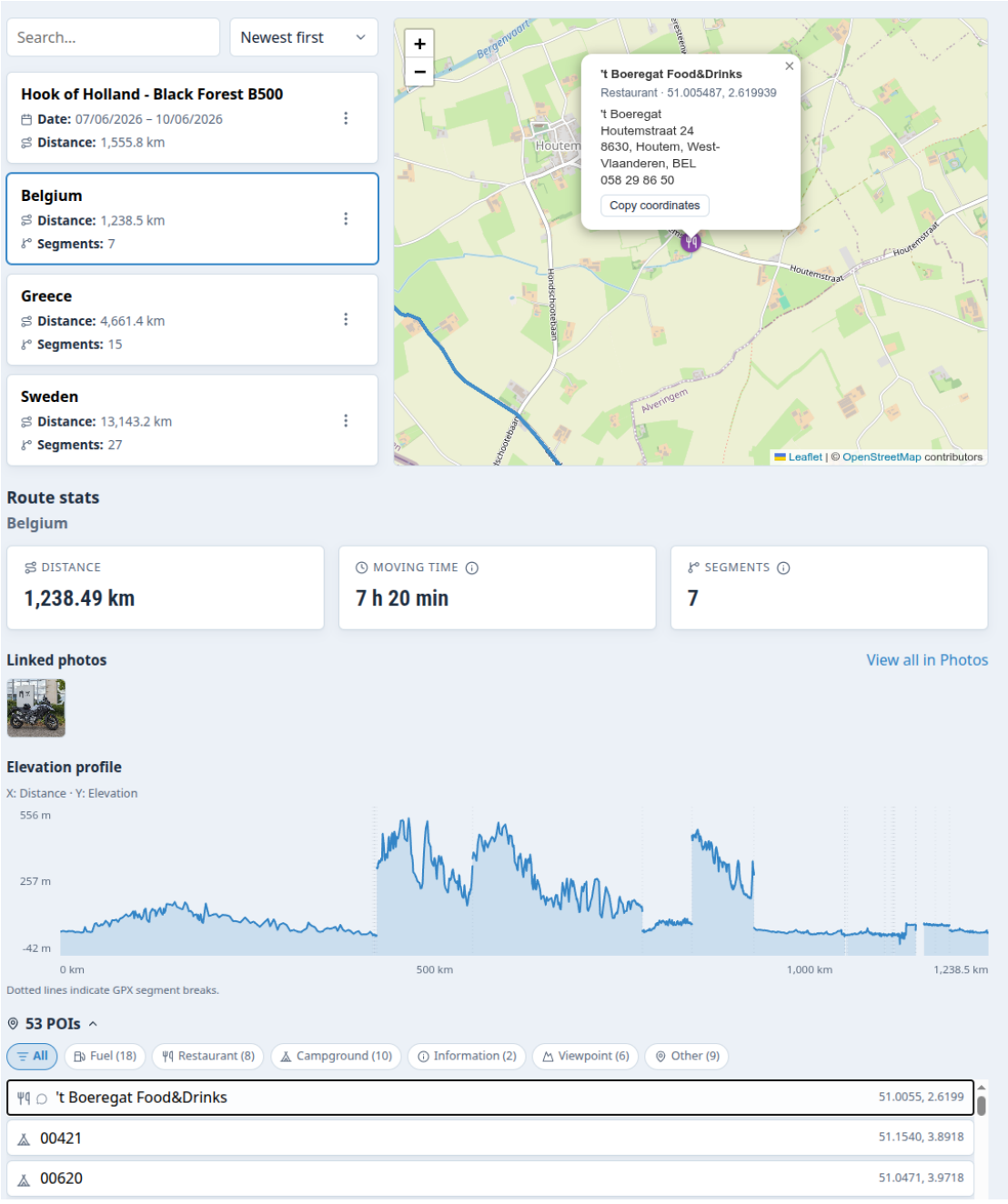
Η αναλυτική προβολή περιλαμβάνει γράφημα μεταβολής υψομέτρου σε σχέση με την απόσταση.

Το γράφημα επιτρέπει την αναγνώριση:

- επίπεδων τμημάτων
- αναβάσεων και καταβάσεων
- διαφορετικών segments
- σημείων με σημαντική υψομετρική μεταβολή

Οι διακεκομμένες κατακόρυφες γραμμές διαχωρίζουν τα segments της διαδρομής.

**Σχήμα 7.23**: Αναλυτική προβολή διαδρομής με υψομετρικό προφίλ



### 7.9.5 Σημεία ενδιαφέροντος

Τα waypoints του GPX παρουσιάζονται ως σημεία ενδιαφέροντος. Η λίστα μπορεί να περιλαμβάνει πρατήρια, εστιατόρια, camping, σημεία ενημέρωσης, θέας και άλλες τοποθεσίες.

Ο χρήστης μπορεί να εφαρμόσει φίλτρα κατηγορίας και να επιλέξει ένα σημείο. Η επιλογή εστιάζει τον χάρτη και εμφανίζει popup με τις διαθέσιμες πληροφορίες.

### 7.9.6 Συνδεδεμένες φωτογραφίες

Κάθε διαδρομή μπορεί να συνδεθεί με φωτογραφίες της μοτοσυκλέτας. Στην αναλυτική προβολή εμφανίζονται οι σχετικές μικρογραφίες (thumbnails), ενώ ο σύνδεσμος "View all in Photos" οδηγεί απευθείας στη συνολική συλλογή φωτογραφιών.

Έτσι, το GPS track, τα σημεία ενδιαφέροντος (waypoints) και το φωτογραφικό υλικό της διαδρομής ενοποιούνται.

## 7.10 Φωτογραφίες και media υλικό

Η ενότητα των φωτογραφιών λειτουργεί ως μια οργανωμένη συλλογή εικόνων για κάθε μοτοσυκλέτα.

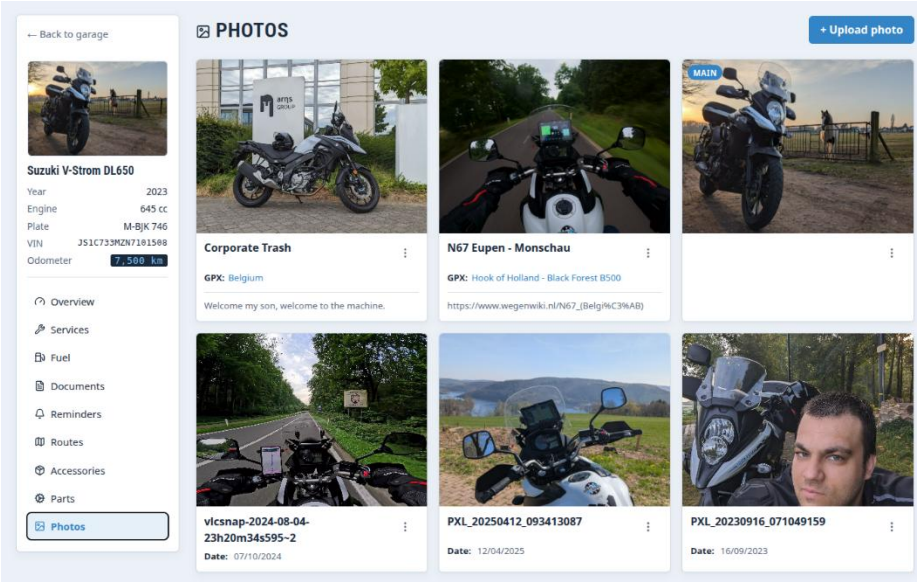
### 7.10.1 Προβολή και διαχείριση

Οι φωτογραφίες παρουσιάζονται σε πλέγμα καρτών. Κάθε κάρτα μπορεί να περιλαμβάνει:

- προεπισκόπηση
- τίτλο ή αρχικό filename
- ημερομηνία
- σημειώσεις
- σύνδεση με GPX διαδρομή
- διαθέσιμες ενέργειες

Το μενού ενεργειών επιτρέπει επεξεργασία metadata, ορισμό κύριας φωτογραφίας (στη sidebar), αλλαγή σύνδεσης με διαδρομή και διαγραφή.

**Σχήμα 7.24**: Σελίδα φωτογραφιών μοτοσυκλέτας



### 7.10.2 Μεταφόρτωση και metadata

Κατά τη μεταφόρτωση μπορούν να προστεθούν:

- τίτλος
- ημερομηνία λήψης
- σημειώσεις
- σύνδεση με διαδρομή

Τα metadata βοηθούν στη χρονολογική και θεματική οργάνωση του υλικού. Μπορούν να χρησιμοποιηθούν για αποθήκευση τοποθεσίας, τεχνικής παρατήρησης, περιγραφής ταξιδιού ή αλλαγής εξοπλισμού.

### 7.10.3 Κύρια φωτογραφία

Μία φωτογραφία μπορεί να οριστεί ως κύρια. Επισημαίνεται με την ένδειξη «MAIN» και επαναχρησιμοποιείται:
- στην κάρτα του οχήματος στο γκαράζ,
- στην πλευρική κάρτα,
- στην επισκόπηση μοτοσυκλέτας.

Αν η κύρια φωτογραφία διαγραφεί, η εφαρμογή ενημερώνει την αντίστοιχη αναφορά του οχήματος.

### 7.10.4 Σύνδεση με GPX διαδρομές

Μια φωτογραφία μπορεί να συνδέεται με συγκεκριμένη διαδρομή. Ο σύνδεσμος εμφανίζεται στην κάρτα και οδηγεί στη σχετική GPX εγγραφή.

Αντίστροφα, η αναλυτική προβολή μιας διαδρομής εμφανίζει τις συνδεδεμένες φωτογραφίες. Κατά τη μετάβαση η διεπαφή μπορεί να εστιάζει στο σχετικό στοιχείο, ώστε να αποφεύγεται η χειροκίνητη αναζήτηση.

### 7.10.5 Ιδιωτικότητα

Οι φωτογραφίες συνδέονται με το όχημα και τον λογαριασμό του χρήστη και δεν εκτίθενται ως δημόσια αρχεία. Η προβολή και η λήψη τους πραγματοποιούνται μέσα από την εφαρμογή μετά από έλεγχο πρόσβασης.

## 7.11 Εισαγωγή, εξαγωγή και αναφορές

Οι λειτουργίες εισαγωγής, εξαγωγής και αναφορών επιτρέπουν τη μεταφορά δεδομένων, τη δημιουργία αντιγράφων ασφαλείας και την παραγωγή ανθρώπινα αναγνώσιμου ιστορικού.

### 7.11.1 Εξαγωγή οχήματος

Ο χρήστης μπορεί να εξαγάγει μία μοτοσυκλέτα ως ZIP archive που περιλαμβάνει:

- στοιχεία οχήματος
- συντηρήσεις
- ανεφοδιασμούς
- αξεσουάρ
- ανταλλακτικά
- έγγραφα
- υπενθυμίσεις

- διαδρομές
- φωτογραφίες
- συνημμένα

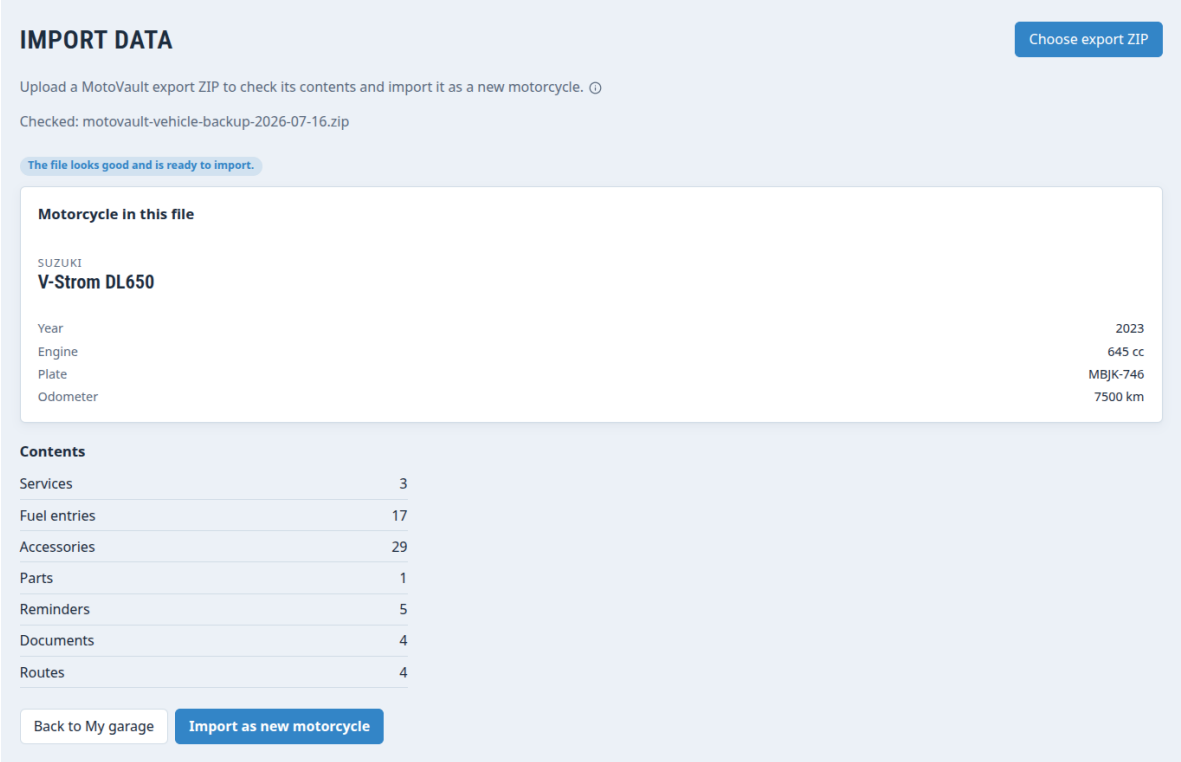
Η λειτουργία αυτή δημιουργεί ένα πλήρες backup του οχήματος, το οποίο μπορεί να εισαχθεί ξανά στο σύστημα ανά πάσα στιγμή.

### 7.11.2 Προεπισκόπηση και εισαγωγή

Πριν από την εισαγωγή, η εφαρμογή παρουσιάζει:
- στοιχεία του οχήματος,
- αριθμό εγγραφών ανά κατηγορία,
- κατάσταση εγκυρότητας του archive,
- επιλογή επιβεβαίωσης.

Η προεπισκόπηση δεν δημιουργεί δεδομένα. Η εισαγωγή εκτελείται μόνο μετά από επιβεβαίωση και δημιουργεί νέα μοτοσυκλέτα, χωρίς να αντικαθιστά ή να συγχωνεύει υπάρχον όχημα.

**Σχήμα 7.25**: Προεπισκόπηση περιεχομένου archive πριν από την εισαγωγή



### 7.11.3 Backup και restore λογαριασμού

Το πλήρες backup λογαριασμού περιλαμβάνει:
- όλες τις μοτοσυκλέτες,
- τις σχετικές εγγραφές,
- φωτογραφίες,
- συνημμένα,
- αρχικά αρχεία GPX,
- βασικές προτιμήσεις.

Δεν περιλαμβάνει κωδικούς, ενεργές συνεδρίες ή συνδεδεμένες εξωτερικές ταυτότητες.

Η επαναφορά επιτρέπεται όταν ο λογαριασμός δεν περιέχει ήδη μοτοσυκλέτες, ώστε να αποφεύγονται συγκρούσεις. Οι σχετικές ενέργειες εμφανίζονται στη σελίδα λογαριασμού του Σχήματος 7.4.

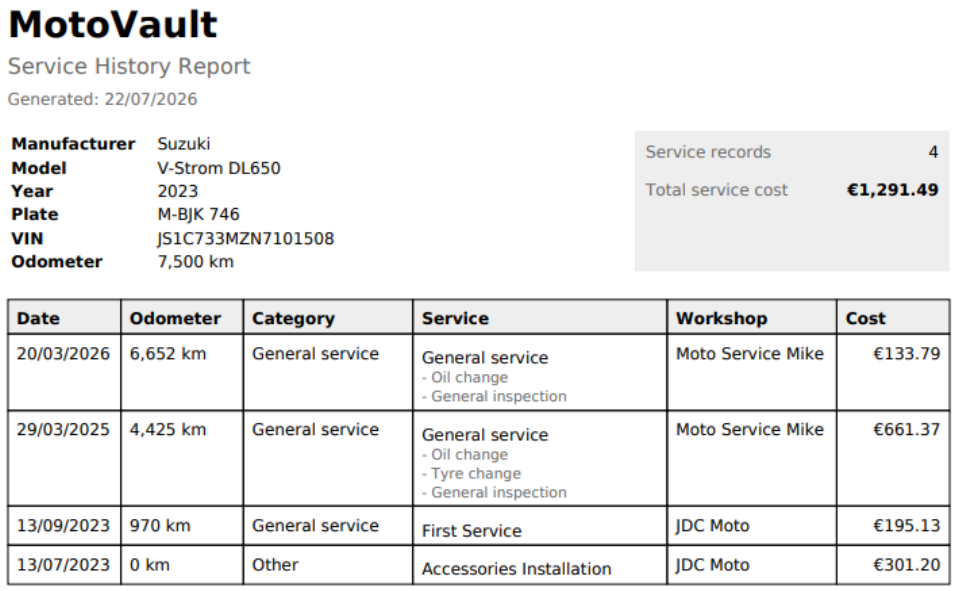
### 7.11.4 Αναφορά ιστορικού συντήρησης

Η εφαρμογή δημιουργεί PDF με:

- βασικά στοιχεία οχήματος
- ημερομηνία δημιουργίας
- αριθμό εγγραφών
- συνολικό κόστος
- αναλυτικό πίνακα εργασιών
- ημερομηνίες
- χιλιόμετρα
- κατηγορίες
- συνεργείο και κόστος

Η αναφορά μπορεί να χρησιμοποιηθεί για προσωπική αρχειοθέτηση, παρουσίαση σε συνεργείο, αποθήκευση εκτός εφαρμογής ή τεκμηρίωση κατά την πώληση. Παραμένει προσωπική αναφορά και όχι επίσημη πιστοποίηση.

**Σχήμα 7.26**: PDF αναφορά ιστορικού συντήρησης



**MotoVault**

Service History Report

Generated: 22/07/2026

| | |
|---|---|
| **Manufacturer** | Suzuki |
| **Model** | V-Strom DL650 |
| **Year** | 2023 |
| **Plate** | M-BJK 746 |
| **VIN** | JS1C733MZN7101508 |
| **Odometer** | 7,500 km |

| | |
|---|---|
| Service records | 4 |
| Total service cost | **€1,291.49** |

| Date | Odometer | Category | Service | Workshop | Cost |
|---|---|---|---|---|---|
| 20/03/2026 | 6,652 km | General service | General service<br>- Oil change<br>- General inspection | Moto Service Mike | €133.79 |
| 29/03/2025 | 4,425 km | General service | General service<br>- Oil change<br>- Tyre change<br>- General inspection | Moto Service Mike | €661.37 |
| 13/09/2023 | 970 km | General service | First Service | JDC Moto | €195.13 |
| 13/07/2023 | 0 km | Other | Accessories Installation | JDC Moto | €301.20 |

### 7.11.5 Διαφορά backup και PDF

Το backup προορίζεται για επαναφορά και μεταφορά δομημένων δεδομένων μαζί με τα αρχεία τους.

Το PDF προορίζεται για ανάγνωση, εκτύπωση και παρουσίαση του ιστορικού συντήρησης. Δεν μπορεί να χρησιμοποιηθεί για επανεισαγωγή δεδομένων.

## 7.12 Προσαρμοστικότητα και πολυγλωσσική υποστήριξη

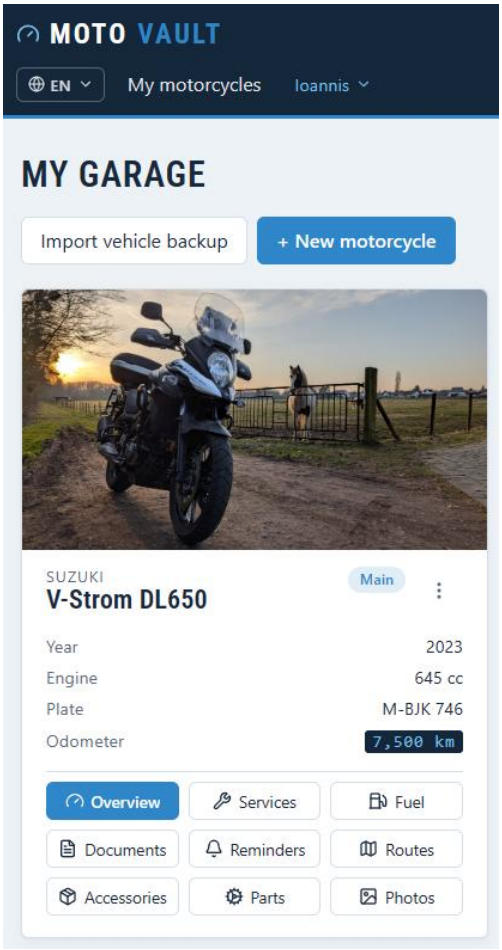
Το UI είναι αυτομάτως προσαρμόσιμο (responsive) σε διαφορετικά μεγέθη οθόνης και προσφέρει πλήρη υποστήριξη για ελληνικά και αγγλικά.

### 7.12.1 Προβολή γκαράζ σε κινητή συσκευή

Στη mobile προβολή οι κάρτες των οχημάτων τοποθετούνται σε μία κατακόρυφη στήλη. Η φωτογραφία καταλαμβάνει το διαθέσιμο πλάτος και οι λειτουργικές συντομεύσεις οργανώνονται σε συμπαγές πλέγμα.

Οι βασικές ενέργειες παραμένουν διαθέσιμες χωρίς οριζόντια κύλιση.

**Σχήμα 7.27**: Προβολή του ψηφιακού γκαράζ σε κινητή συσκευή

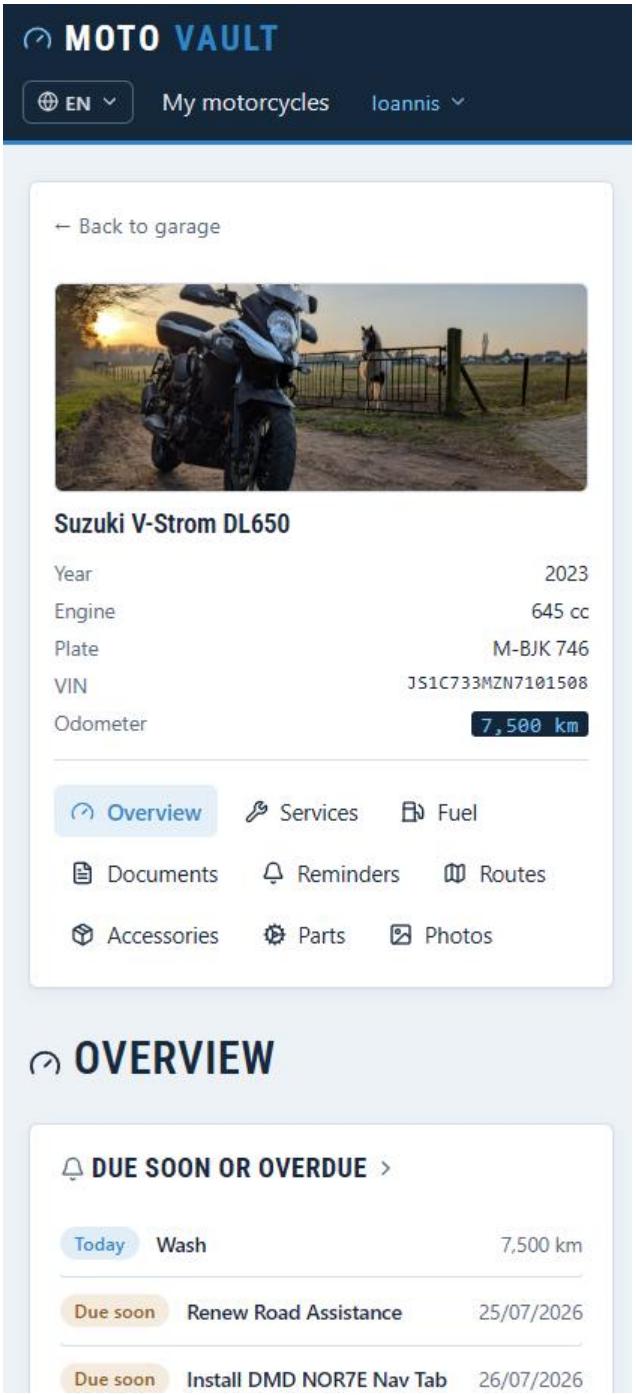


### 7.12.2 Προβολή οχήματος σε μικρή οθόνη

Στην αναλυτική προβολή η πλευρική κάρτα μετατρέπεται σε πλήρους πλάτους τμήμα πάνω από το κύριο περιεχόμενο.

Το μενού πλοήγησης αναδιπλώνεται σε περισσότερες γραμμές και οι κάρτες υπενθυμίσεων, εγγράφων και στατιστικών εμφανίζονται κατακόρυφα.

**Σχήμα 7.28**: Προσαρμοστική προβολή της επισκόπησης μοτοσυκλέτας



### 7.12.3 Διατήρηση λειτουργικότητας

Η mobile προβολή δεν αποτελεί ξεχωριστή περιορισμένη έκδοση. Παραμένουν διαθέσιμες:
- όλες οι λειτουργικές περιοχές,
- η δημιουργία και επεξεργασία εγγραφών,
- οι φόρμες,
- οι υπενθυμίσεις και τα έγγραφα,
- τα στατιστικά,
- οι φωτογραφίες και οι διαδρομές.

Η κύρια αλλαγή αφορά την αναδιάταξη των στοιχείων, των πινάκων, των καρτών και των κουμπιών.

### 7.12.4 Γλώσσες

Ο χρήστης μπορεί να αλλάξει τη γλώσσα από τον επιλογέα της επάνω μπάρας χωρίς να αποσυνδεθεί.

Η τρέχουσα έκδοση υποστηρίζει τη πλειοψηφία των ευρωπαικών γλωσσών.

Οι τίτλοι, τα κουμπιά, τα πεδία, τα μηνύματα, οι καταστάσεις και τα validation errors μεταφράζονται. Η προσθήκη και άλλων γλωσσών είναι τεχνικά δυνατή και εύκολη με χρήση AI εργαλείων.

### 7.12.5 Μονάδες και τοπική μορφοποίηση

Οι προτιμήσεις λογαριασμού επιτρέπουν επιλογή μονάδων απόστασης, όγκου και κατανάλωσης, νομίσματος και μορφής ημερομηνίας.

Οι επιλογές εφαρμόζονται στις σχετικές οθόνες χωρίς να αλλάζουν τα βασικά αποθηκευμένα δεδομένα. Η γλώσσα, το νόμισμα και οι μονάδες παραμένουν ανεξάρτητες ρυθμίσεις.

## 7.13 Συνολική αποτίμηση της διεπαφής

Η παρουσίαση του Moto-Vault ανέδειξε τον τρόπο με τον οποίο οι επιμέρους λειτουργίες του συστήματος συνδέονται σε ολοκληρωμένα σενάρια χρήσης (user flows) με επίκεντρο τη μοτοσυκλέτα.

### 7.13.1 Συνοχή και οργάνωση

Οι κύριες σελίδες χρησιμοποιούν κοινά μοτίβα:
- κάρτες,
- λίστες,
- φίλτρα,
- φόρμες,
- συνοπτικούς δείκτες,
- γραφήματα,
- μενού ενεργειών.

Η μπάρα στα αριστερά διατηρεί πάντα ορατό το επιλεγμένο όχημα, δίνοντας άμεση πρόσβαση στις βασικές ενότητες. Οι φόρμες διατηρούν μια ενιαία δομή με ξεκάθαρες ετικέτες, υποχρεωτικά πεδία, μηνύματα επικύρωσης (validation) και κουμπιά αποθήκευσης ή ακύρωσης.

Οι συχνές ενέργειες είναι άμεσα διαθέσιμες, ενώ οι λιγότερο συχνές ομαδοποιούνται σε μενού. Τέλος, οι επικίνδυνες/κρίσιμες ενέργειες (όπως η διαγραφή) απαιτούν επιβεβαίωση και προβάλλονται σε διακριτό Danger Zone.

### 7.13.2 Διασύνδεση λειτουργιών

Οι λειτουργικές περιοχές δεν λειτουργούν ως απομονωμένες λίστες.

Ενδεικτικά:
- οι εργασίες συντήρησης συνδέονται με αξεσουάρ και ανταλλακτικά,
- οι φωτογραφίες συνδέονται με GPX διαδρομές,
- οι υπενθυμίσεις και τα έγγραφα εμφανίζονται στην επισκόπηση,
- τα κόστη συντήρησης, καυσίμων και εξοπλισμού συγκεντρώνονται στα dashboards,
- η κύρια φωτογραφία επαναχρησιμοποιείται στο γκαράζ και στην επισκόπηση.

Η αμφίδρομη πλοήγηση (cross-referencing) και η άμεση μετάβαση στις σχετικές εγγραφές σχεδιάστηκαν ώστε ο χρήστης να βρίσκει αμέσως αυτό που ψάχνει, χωρίς χειροκίνητη αναζήτηση.

### 7.13.3 Προσαρμοστικότητα και σημεία προς αξιολόγηση

Η διεπαφή(UI/UX) προσαρμόζεται σε desktop και mobile browser, ενώ υποστηρίζει ελληνικά, αγγλικά και διαφορετικές μονάδες ή μορφές παρουσίασης.

Η παρουσίαση του κεφαλαίου επιβεβαιώνει ότι οι λειτουργίες είναι διαθέσιμες και οργανωμένες σε σταθερό περιβάλλον, αλλά σίγουρα δεν αποδεικνύει όμως από μόνη της ότι η εφαρμογή είναι εύχρηστη για όλους τους χρήστες.

Ζητήματα όπως:
- η ταχύτητα εξοικείωσης,
- η σαφήνεια της πλοήγησης,
- η ευκολία συμπλήρωσης των φορμών,
- η αντιλαμβανόμενη πολυπλοκότητα,
- η συνολική ικανοποίηση,

εξετάζονται μέσα από την αξιολόγηση χρηστικότητας.

### 7.13.4 Μετάβαση στην αξιολόγηση

Το Moto-Vault καλύπτει όλες τις βασικές ροές των απαιτήσεων, συγκεντρώνοντάς τες σε ένα ενοποιημένο UI.

Στο επόμενο κεφάλαιο εξετάζεται η τεχνική αξιολόγηση και η αντίληψη των χρηστών για τη χρηστικότητα του συστήματος.

# 8. Αξιολόγηση και Συζήτηση

## 8.1 Εισαγωγή

Μετά την ολοκλήρωση της σχεδίασης, της υλοποίησης και της παρουσίασης του Moto-Vault, ακολουθεί η αξιολόγηση του τελικού συστήματος από χρήστες.

Στην αξιολόγηση θέλουμε να δούμε πώς αντιλαμβάνονται οι χρήστες τη χρηστικότητα της εφαρμογής και κατά πόσο θεωρούν ότι μπορεί να υποστηρίξει με κατανοητό τρόπο τη διαχείριση πληροφοριών μοτοσυκλέτας.

Για την ποσοτική αποτίμηση χρησιμοποιήθηκε το System Usability Scale. Επιπλέον προστέθηκαν δύο συμπληρωματικές ερωτήσεις σχετικά με την πρόθεση μελλοντικής χρήσης και την πρόθεση σύστασης της εφαρμογής, οι οποίες αναλύονται ξεχωριστά και δεν συμμετέχουν στον υπολογισμό της βαθμολογίας SUS.

## 8.2 Σχεδιασμός της αξιολόγησης και συμμετέχοντες

Η αξιολόγηση του Moto-Vault πραγματοποιήθηκε με τη συμμετοχή δώδεκα μοτοσικλετιστών, δηλαδή ατόμων που ανήκουν στο βασικό στόχο της εφαρμογής. Οι δέκα ήταν γνωστοί στον ερευνητή, ενώ οι δύο συμμετείχαν έπειτα από πρόταση να δοκιμάσουν την εφαρμογή.

Οι συμμετέχοντες είχαν τη δυνατότητα ελεύθερης εξερεύνησης του συστήματος. Δεν ακολούθησαν κοινό σύνολο προκαθορισμένων εργασιών και δεν υπήρχε συγκεκριμένο χρονικό όριο. Κάθε άτομο χρησιμοποίησε την εφαρμογή για όσο χρόνο επιθυμούσε και περιηγήθηκε στις λειτουργίες που θεωρούσε περισσότερο σχετικές με τις ανάγκες του.

Δεν ζητήθηκε καθοδήγηση για την πλοήγηση, τις φόρμες ή τις βασικές λειτουργίες. Η μόνη βοήθεια που ζητήθηκε αφορούσε το πού μπορούσαν να βρεθούν αρχεία GPX για τη δοκιμή της σχετικής λειτουργίας.

Μετά την εξερεύνηση, οι συμμετέχοντες συμπλήρωσαν ηλεκτρονικό ερωτηματολόγιο μέσω Google Forms. Όλοι απάντησαν σε όλες τις ερωτήσεις, επομένως το τελικό σύνολο δεδομένων δεν περιείχε κενές τιμές.

Η μέθοδος μπορεί να χαρακτηριστεί ως μη ελεγχόμενη αξιολόγηση χρηστικότητας (unmoderated usability testing) με δείγμα ευκολίας και ελεύθερη εξερεύνηση. Ήταν κατάλληλη για μια αρχική αποτίμηση του τεχνουργήματος, αλλά δεν επιτρέπει τη μέτρηση χρόνου ολοκλήρωσης, ποσοστού επιτυχίας ή αριθμού σφαλμάτων σε κοινές εργασίες.

## 8.3 Εργαλείο και διαδικασία αξιολόγησης

Για την ποσοτική αποτίμηση της χρηστικότητας χρησιμοποιήθηκε το System Usability Scale, γνωστό ως SUS (Brooke, 1996). Το SUS αποτελείται από δέκα προτάσεις και παράγει συνολική βαθμολογία από 0 έως 100.

Οι συμμετέχοντες απάντησαν σε πενταβάθμια κλίμακα:

1. Διαφωνώ απόλυτα
2. Διαφωνώ
3. Ούτε συμφωνώ ούτε διαφωνώ
4. Συμφωνώ
5. Συμφωνώ απόλυτα

Οι δέκα προτάσεις ήταν:

1. Πιστεύω ότι θα ήθελα να χρησιμοποιώ το Moto-Vault συχνά.
2. Βρήκα το σύστημα άσκοπα περίπλοκο.
3. Θεωρώ ότι το σύστημα ήταν εύκολο στη χρήση.
4. Πιστεύω ότι θα χρειαζόμουν υποστήριξη για να μπορέσω να χρησιμοποιήσω το σύστημα.
5. Βρήκα ότι οι διάφορες λειτουργίες του συστήματος ήταν καλά ενσωματωμένες και συνδεδεμένες μεταξύ τους.
6. Θεωρώ ότι υπήρχαν πολλές ασυνέπειες στο Moto-Vault.
7. Φαντάζομαι ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν το σύστημα πολύ γρήγορα.
8. Βρήκα το σύστημα πολύ δύσχρηστο ή βαρύ στη χρήση.
9. Ένιωσα μεγάλη αυτοπεποίθηση και σιγουριά χρησιμοποιώντας το σύστημα.
10. Χρειάστηκε να μάθω πολλά πράγματα προτού μπορέσω να ξεκινήσω και να εξοικειωθώ με το σύστημα.

Για τις μονές θετικά διατυπωμένες προτάσεις, η συνεισφορά υπολογίζεται απάντηση μείον 1. Για τις ζυγές αρνητικά διατυπωμένες προτάσεις, η συνεισφορά υπολογίζεται 5 μείον την απάντηση. Το άθροισμα των δέκα συνεισφορών πολλαπλασιάζεται με 2,5.

SUS Score = Άθροισμα συνεισφορών × 2,5

Στο ερωτηματολόγιο προστέθηκαν επίσης δύο συμπληρωματικές ερωτήσεις:

1. Θα συνεχίσω να χρησιμοποιώ την εφαρμογή στο μέλλον.
2. Θα προτείνω την εφαρμογή σε φίλους “μηχανόβιους”.

Οι δύο αυτές ερωτήσεις δεν αποτελούν μέρος του SUS και δεν συμμετείχαν στον υπολογισμό της συνολικής βαθμολογίας. Αναλύθηκαν ξεχωριστά ως ενδείξεις πρόθεσης μελλοντικής χρήσης και σύστασης.

## 8.4 Αποτελέσματα αξιολόγησης

### 8.4.1 Αποτελέσματα SUS

**Πίνακας 8.1:** Ατομικές βαθμολογίες SUS

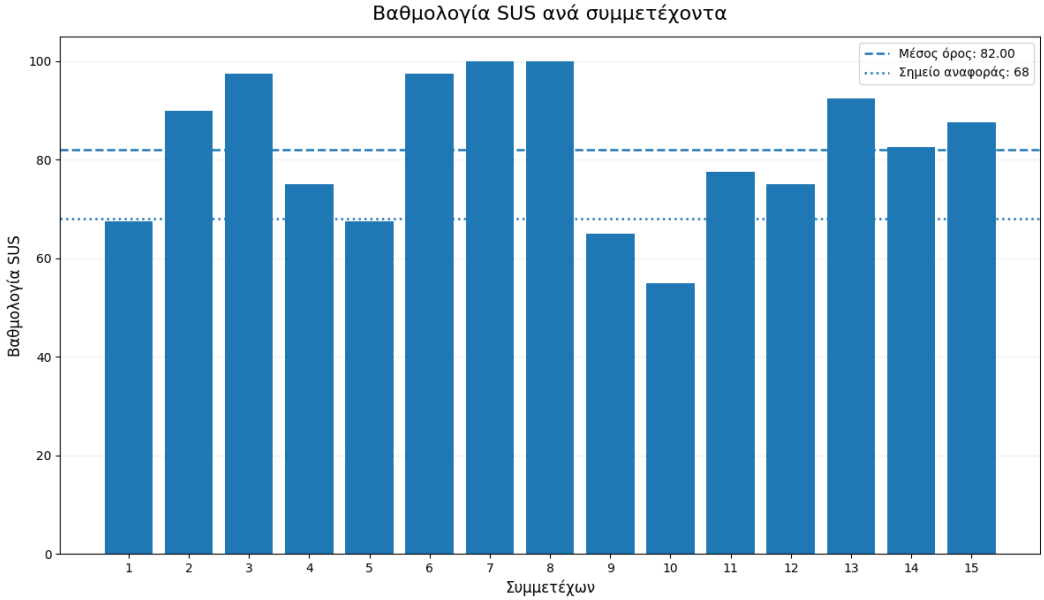
| **Συμμετέχων** | **Βαθμολογία SUS** |
|---|---|
| 1 | 67,5 |
| 2 | 90,0 |
| 3 | 97,5 |
| 4 | 75,0 |
| 5 | 67,5 |
| 6 | 97,5 |
| 7 | 100,0 |
| 8 | 100,0 |
| 9 | 65,0 |
| 10 | 55,0 |
| 11 | 77,5 |
| 12 | 75,0 |
| 13 | 92,5 |
| 14 | 82,5 |
| 15 | 87,5 |

**Πίνακας 8.2:** Βασικά περιγραφικά στατιστικά

| Δείκτης | Τιμή |
|---|---|
| Συμμετέχοντες | 15 |
| Μέσος όρος | 82 |
| Διάμεσος | 82,50 |
| Ελάχιστη τιμή | 55,00 |
| Μέγιστη τιμή | 100,00 |
| Τυπική απόκλιση | 14,37 |

Ο μέσος όρος 82 είναι θετική ένδειξη υψηλής αντιλαμβανόμενης χρηστικότητας και βρίσκεται πάνω από το ευρέως χρησιμοποιούμενο σημείο αναφοράς των 68 μονάδων (Sauro, 2011). Το εύρος από 55 έως 100 δείχνει ότι η εμπειρία δεν ήταν απολύτως ομοιόμορφη για όλους.

**Σχήμα 8.1**: Βαθμολογία SUS ανά συμμετέχοντα




## 8.4.2 Πρόσθετες ερωτήσεις

Στην ερώτηση «**Θα συνεχίσω να χρησιμοποιώ την εφαρμογή στο μέλλον**» ο μέσος όρος ήταν 3,75.

Από τους 15 συμμετέχοντες:

- 7 επέλεξαν 4 ή 5,
- 8 επέλεξαν την ουδέτερη τιμή 3,
- κανένας δεν επέλεξε 1 ή 2.

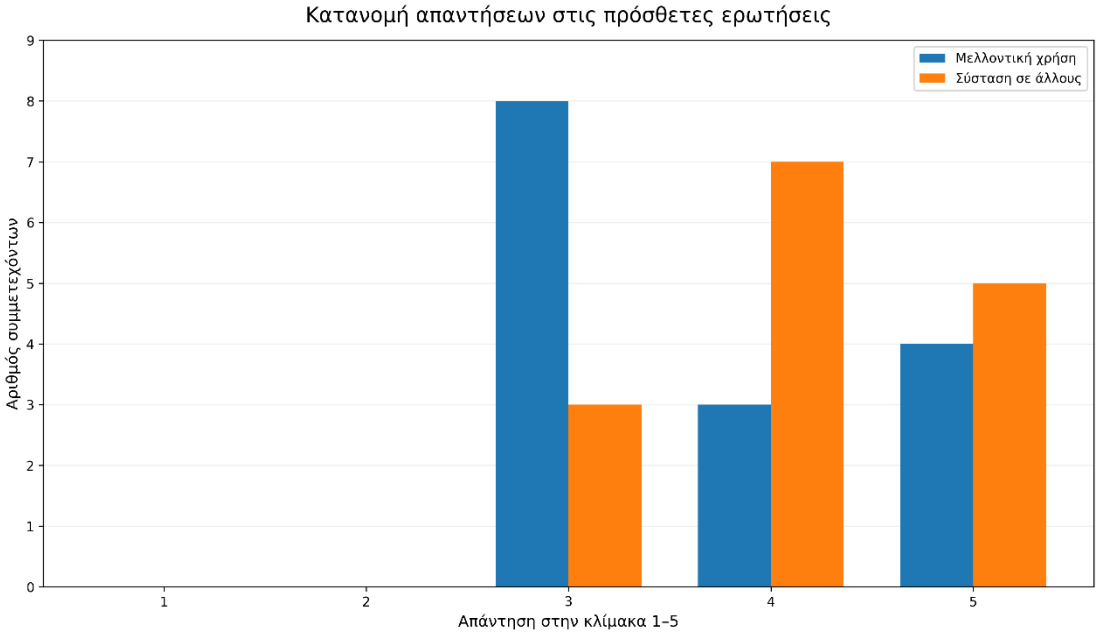
Στην ερώτηση «**Θα προτείνω την εφαρμογή σε φίλους "μηχανόβιους"**» ο μέσος όρος ήταν 4,13.

Από τους 15 συμμετέχοντες:
- 12 επέλεξαν 4 ή 5,
- 3 επέλεξαν την ουδέτερη τιμή 3,
- κανένας δεν επέλεξε 1 ή 2.

**Πίνακας 8.3**: Αποτελέσματα αξιολόγησης προθέσεων μελλοντικής χρήσης και σύστασης

| Ερώτηση | Μέσος όρος | Θετικές απαντήσεις | Ουδέτερες | Αρνητικές |
|---|---|---|---|---|
| Μελλοντική χρήση | 3,73 | 7 | 8 | 0 |
| Σύσταση σε άλλους | 4,13 | 12 | 3 | 0 |

**Σχήμα 8.2**: Κατανομή απαντήσεων στις πρόσθετες ερωτήσεις




Οι χρήστες δήλωσαν πρόθυμοι να συστήσουν την εφαρμογή σε άλλους αναβάτες (μέσος όρος 4,13), αν και εμφανίστηκαν πιο επιφυλακτικοί όσον αφορά τη δική τους καθημερινή χρήση (3,73).

## 8.5 Συζήτηση και προτάσεις βελτίωσης

Η μέση βαθμολογία SUS 82 είναι καλή ένδειξη ότι οι συμμετέχοντες αντιλήφθηκαν το Moto-Vault ως εύχρηστο και σχετικά εύκολο στην εκμάθηση. Η ένδειξη αυτή ενισχύεται από το γεγονός ότι δεν καταγράφηκε ανάγκη καθοδήγησης για τις βασικές λειτουργίες.

Η υψηλή τάση των χρηστών να συστήσουν την εφαρμογή δείχνει ότι αναγνώρισαν την πρακτική της αξία. Αντίθετα, η πιο συγκρατημένη πρόθεση για συχνή χρήση φανερώνει ότι η καλή χρηστικότητα από μόνη της δεν αρκεί για να καθιερώσει την εφαρμογή εργαλείο στην καθημερινότητά τους.

Η μακροχρόνια χρήση μπορεί να επηρεάζεται από παράγοντες όπως:
- η συχνότητα χειροκίνητης καταχώρισης,
- η ευκολία πρόσβασης από κινητή συσκευή,
- η σύνδεση με εξωτερικές υπηρεσίες,
- η πραγματική ανάγκη κάθε χρήστη.

Ορισμένοι συμμετέχοντες διατύπωσαν τρεις βασικές προτάσεις βελτίωσης.

### 8.5.1 Αυτοματοποίηση καταχωρίσεων
Προτάθηκε η αυτόματη δημιουργία εγγραφών ανεφοδιασμού από φωτογραφίες απόδειξης και χιλιομετρητή. Μια τέτοια λειτουργία θα μπορούσε να χρησιμοποιεί OCR ή τεχνητή νοημοσύνη για να προσυμπληρώνει ημερομηνία, ποσότητα καυσίμου, κόστος, πρατήριο και χιλιομετρική ένδειξη.

### 8.5.2 Εφαρμογή για κινητές συσκευές
Προτάθηκε η δημιουργία εφαρμογής για κινητές συσκευές, ώστε η καταχώριση δεδομένων να γίνεται ευκολότερα κατά τη χρήση της μοτοσυκλέτας, στο πρατήριο ή στο συνεργείο.

### 8.5.3 Διασύνδεση με συνεργεία
Προτάθηκε η σύνδεση των εγγραφών συντήρησης με πραγματικά συνεργεία, ώστε το ιστορικό να συνδέεται με τον επαγγελματία που πραγματοποίησε την εργασία.

## 8.6 Περιορισμοί και συνολική αποτίμηση
Η αξιολόγηση παρείχε θετικές ενδείξεις, αλλά παρουσιάζει συγκεκριμένα θέματα.

Πρώτον, το δείγμα ήταν μικρό και αποτελούνταν από 12 άτομα. Δεύτερον, επρόκειτο για δείγμα ευκολίας, καθώς οι 10 από τους 12 συμμετέχοντες ήταν γνωστοί στον ερευνητή. Η προσωπική σχέση μπορεί να επηρέασε τη διάθεση διατύπωσης αυστηρής κριτικής.

Οι συμμετέχοντες είχαν ελεύθερη εξερεύνηση χωρίς κοινές προκαθορισμένες εργασίες και χωρίς καταγεγραμμένο χρόνο χρήσης. Επομένως, δεν είναι γνωστό αν όλοι δοκίμασαν τις ίδιες λειτουργίες ή με το ίδιο βάθος.

Δεν συλλέχθηκαν μετρήσεις, όπως:
- χρόνος ολοκλήρωσης,
- αριθμός σφαλμάτων(bugs),
- ποσοστό επιτυχίας,
- αριθμός ενεργειών,
- σημεία εγκατάλειψης.

Δεν καταγράφηκαν δημογραφικά στοιχεία ή η προηγούμενη εμπειρία των δοκιμαστών (πχ. εξοικείωση με GPX, έτη οδήγησης). Αυτό κατέστησε αδύνατη τη διαφοροποιημένη ανάλυση των αποτελεσμάτων ανάλογα με το προφίλ του κάθε χρήστη.

Η αξιολόγηση εστιάζει στην αρχική αλληλεπίδραση και όχι στη μακροχρόνια υιοθέτηση του συστήματος. Η εκφρασμένη πρόθεση μελλοντικής χρήσης δεν διασφαλίζει ότι οι συμμετέχοντες θα διατηρούσαν τη συστηματική καταγραφή δεδομένων σε ορίζοντα εβδομάδων ή μηνών.

Ακόμα και έτσι, τα αποτελέσματα παρέχουν θετικές αρχικές ενδείξεις για την αντιλαμβανόμενη χρηστικότητα του Moto-Vault. Η συνολική αποτίμηση της εργασίας και οι κατευθύνσεις μελλοντικής εξέλιξης παρουσιάζονται στο Κεφάλαιο 9.

# 9. Συμπεράσματα και Προοπτικές

## 9.1 Εισαγωγή

Η εργασία εδώ τελείωσε με τον σχεδιασμό, την ανάπτυξη και την αξιολόγηση του Moto-Vault, πετυχαίνοντας τον αρχικό στόχο που είχε τεθεί.

Καλύφθηκε όλος ο κύκλος. Από την αποτύπωση των απαιτήσεων και τη σύγκριση του ανταγωνισμού, μέχρι τον αρχιτεκτονικό σχεδιασμό REST API, τη μοντελοποίηση της βάσης δεδομένων, την υλοποίηση του frontend/backend και τους ελέγχους ασφαλείας και δοκιμών χρηστικότητας. Τα αποτελέσματα της αξιολόγησης επιβεβαίωσαν ότι το σύστημα καλύπτει τις ανάγκες του σύγχρονου μοτοσικλετιστή.

Η ανάπτυξη ακολούθησε τη Design Science Research Methodology, με το Moto-Vault να αποτελεί το κατασκεύασμα της έρευνας.

Εδώ φαίνονται τα βασικά συμπεράσματα της εργασίας, εξετάζεται ο βαθμός επίτευξης των στόχων και παρουσιάζονται οι σημαντικότερες συνεισφορές, οι περιορισμοί και οι δυνατότητες μελλοντικής εξέλιξης του Moto-Vault.

## 9.2 Επίτευξη στόχων και απάντηση στα ερευνητικά ερωτήματα

Η ολοκλήρωση του Moto-Vault επιτρέπει τη συγκεντρωτική αποτίμηση των στόχων και των ερευνητικών ερωτημάτων που τέθηκαν στο Κεφάλαιο 3.

### 9.2.1 Κύριο ερευνητικό ερώτημα

Το κύριο ερευνητικό ερώτημα ήταν:

**Πώς μπορεί να σχεδιαστεί, να αναπτυχθεί και να αξιολογηθεί μια διαδικτυακή πλατφόρμα που συγκεντρώνει και οργανώνει τις βασικές πληροφορίες μιας μοτοσυκλέτας σε ένα ενιαίο και εύχρηστο περιβάλλον;**

Η εργασία απάντησε στο ερώτημα μέσω της ανάπτυξης μιας πλήρους διαδικτυακής εφαρμογής με διακριτό frontend, backend, σχεσιακή βάση δεδομένων και υποσύστημα αποθήκευσης αρχείων.

Το Moto-Vault οργανώνει τις πληροφορίες γύρω από κάθε μοτοσυκλέτα και συγκεντρώνει σε ενιαίο περιβάλλον στοιχεία οχήματος, συντήρηση, καύσιμα, κόστη, αξεσουάρ, ανταλλακτικά, έγγραφα, φωτογραφίες, υπενθυμίσεις και διαδρομές GPX.

### 9.2.2 Λειτουργικές και μη λειτουργικές απαιτήσεις

Το πρώτο επιμέρους ερευνητικό ερώτημα αφορούσε τις **απαιτήσεις που είναι αναγκαίες για την προσωπική ψηφιακή διαχείριση πληροφοριών μοτοσυκλέτας**.

Η ανάλυση έδειξε ότι η λύση δεν αρκεί να περιορίζεται μόνο στη συντήρηση ή στα καύσιμα. Χρειάζεται να υποστηρίζει πολλαπλές κατηγορίες δεδομένων, σαφή σύνδεση με συγκεκριμένο όχημα, ιδιωτικά αρχεία, αναζητήσεις, φίλτρα, στατιστικά, υπενθυμίσεις και δυνατότητες μεταφοράς δεδομένων.

Οι σημαντικότερες μη λειτουργικές απαιτήσεις αφορούσαν:
- ασφάλεια και έλεγχο πρόσβασης,
- ιδιωτικότητα/ασφάλεια δεδομένων και αρχείων,

- αξιοπιστία,
- χρηστικότητα,
- συντηρησιμότητα,
- προσαρμογή σε διαφορετικά μεγέθη οθόνης,
- δυνατότητα αυτοματοποιημένων δοκιμών.

Οι απαιτήσεις αυτές ενσωματώθηκαν στον σχεδιασμό και υλοποιήθηκαν σε μεγάλο βαθμό στην τελική εφαρμογή.

### 9.2.3 Αρχιτεκτονική και τεχνικές επιλογές

Το δεύτερο ερευνητικό ερώτημα εξέταζε **ποια αρχιτεκτονική και ποιες τεχνικές επιλογές μπορούν να υποστηρίξουν τις απαιτήσεις με ασφαλή και συντηρήσιμο τρόπο**.

Η λύση βασίστηκε σε Angular, Spring Boot, PostgreSQL, Flyway, JWT, αρθρωτό μονόλιθο στο backend, κοινή διεπαφή αποθήκευσης αρχείων και αυτοματοποιημένες δοκιμές backend και frontend.

Ο διαχωρισμός αυτός επέτρεψε σαφή όρια ανάμεσα στη διεπαφή, την επιχειρησιακή λογική, τη σχεσιακή βάση και το storage, χωρίς την πρόσθετη πολυπλοκότητα μιας κατανεμημένης αρχιτεκτονικής.

### 9.2.4 Ενοποίηση λειτουργιών

Το τρίτο ερευνητικό ερώτημα αφορούσε **τον βαθμό στον οποίο το Moto-Vault μπορεί να συνδυάσει λειτουργίες που συχνά εμφανίζονται κατανεμημένες σε διαφορετικά εργαλεία**.

Το Moto-Vault συνδυάζει στο ίδιο περιβάλλον τη διαχείριση οχημάτων, συντήρησης, καυσίμων, εξόδων, εξαρτημάτων, εγγράφων, φωτογραφιών, υπενθυμίσεων και διαδρομών GPX. Η βασική διαφοροποίηση δεν είναι μόνο η συνύπαρξη των λειτουργιών, αλλά η κοινή οργάνωσή τους γύρω από κάθε μοτοσυκλέτα.

### 9.2.5 Αντιλαμβανόμενη χρηστικότητα

Το τέταρτο ερευνητικό ερώτημα εξέταζε **πώς αντιλαμβάνονται οι χρήστες τη χρηστικότητα του Moto-Vault**.

Η μέση βαθμολογία SUS ήταν 82, γεγονός που αποτελεί θετική ένδειξη υψηλής αντιλαμβανόμενης χρηστικότητας για το συγκεκριμένο δείγμα. Η αναλυτική παρουσίαση και ερμηνεία των αποτελεσμάτων δόθηκε στο Κεφάλαιο 8.

### 9.2.6 Περιορισμοί και δυνατότητες βελτίωσης

Το πέμπτο ερευνητικό ερώτημα αφορούσε τους **περιορισμούς και τις δυνατότητες εξέλιξης**.

Οι βασικοί περιορισμοί αφορούν το εύρος της εφαρμογής, τη χειροκίνητη εισαγωγή δεδομένων, την απουσία εξωτερικών διασυνδέσεων(OBD, κλπ.), και τους περιορισμούς της αξιολόγησης. Οι περιορισμοί και οι αντίστοιχες κατευθύνσεις βελτίωσης παρουσιάζονται αναλυτικά στις Ενότητες 9.4 και 9.5.

### 9.2.7 Συνολική επίτευξη στόχων
Το project παρέμεινε πιστό στον αρχικό σχεδιασμό, επιτυγχάνοντας στο σύνολό τους τους στόχους που είχαν τεθεί.

## 9.3 Συνεισφορά της εργασίας
Όπως έδειξε η ανάλυση των άλλων εφαρμογών, η αξία της διπλωματικής εστιάζει στην ενοποίηση λειτουργιών. Το MotoVault καταφέρνει να συγκεντρώσει σε ένα ενιαίο web περιβάλλον δυνατότητες που μέχρι σήμερα ήταν κατακερματισμένες ή διαθέσιμες αποκλειστικά σε mobile εφαρμογές

### 9.3.1 Σχεδιαστική συνεισφορά
Σε επίπεδο σχεδιασμού, η εργασία πρότεινε ένα ενιαίο μοντέλο οργάνωσης δεδομένων με κεντρική οντότητα τη μοτοσυκλέτα.

Αντί οι πληροφορίες να αντιμετωπίζονται ως ανεξάρτητες εγγραφές, συσχετίζονται με συγκεκριμένο όχημα και μπαίνουν σε κοινό περιβάλλον. Έχουμε συγκεντρωτική παρουσίαση:
- τεχνικών στοιχείων
- ιστορικού συντήρησης
- ανεφοδιασμών
- εξόδων
- ανταλλακτικών
- αξεσουάρ
- εγγράφων
- υπενθυμίσεων
- φωτογραφιών
- διαδρομών

Περιορίζεται έτσι η ανάγκη χρήσης πολλών διαφορετικών εργαλείων.

### 9.3.2 Τεχνική συνεισφορά
Σε τεχνικό επίπεδο, αναπτύχθηκε μια πλήρης fullstack εφαρμογή με σαφή διαχωρισμό frontend, backend, βάσης δεδομένων και αποθήκευσης αρχείων. Η επιλογή αρθρωτού μονόλιθου επέτρεψε τον διαχωρισμό των λειτουργικών περιοχών χωρίς την πρόσθετη πολυπλοκότητα μιας κατανεμημένης αρχιτεκτονικής.

### 9.3.3 Συνεισφορά ως προς την ασφάλεια και αξιοπιστία
Η ασφάλεια και η αξιοπιστία ενσωματώθηκαν στον σχεδιασμό μέσω ελέγχου πρόσβασης, ιδιωτικής διαχείρισης αρχείων, επικύρωσης εισόδων, ασφαλούς διαχείρισης λογαριασμών και μηχανισμών ανάκτησης από αποτυχίες.

### 9.3.4 Η πρακτική συνεισφορά
Το Moto-Vault μπορεί να χρησιμοποιηθεί ως προσωπικό εργαλείο οργάνωσης πληροφοριών μοτοσυκλέτας.

Η πρακτική του αξία βρίσκεται στη δυνατότητα του χρήστη να:
- διατηρεί πλήρες ιστορικό
- οργανώνει αρχεία και φωτογραφίες
- παρακολουθεί κόστη και κατανάλωση
- δημιουργεί υπενθυμίσεις

- αποθηκεύει διαδρομές
- εξάγει ή επαναφέρει δεδομένα
- δημιουργεί αναφορά συντήρησης

Η συγκέντρωση των λειτουργιών αυτών σε ένα σύστημα μειώνει τον κατακερματισμό και διευκολύνει τη διατήρηση ψηφιακού αρχείου.

### 9.3.5 Μεθοδολογική συνεισφορά

Η εφαρμογή της Design Science Research Methodology παρείχε μια οργανωμένη διαδικασία σύνδεσης του προβλήματος με το τελικό προϊόν.

Η εργασία περιλάμβανε:

1. αναγνώριση του προβλήματος
2. καθορισμό στόχων
3. ανάλυση απαιτήσεων
4. σχεδιασμό
5. ανάπτυξη
6. παρουσίαση
7. αξιολόγηση

Η διαδικασία αυτή επιτρέπει η τεχνική υλοποίηση να εξετάζεται ως απάντηση σε συγκεκριμένες ανάγκες και όχι μόνο ως άσκηση προγραμματισμού.

### 9.3.6 Η εμπειρική συνεισφορά

Η αξιολόγηση με το SUS παρείχε αρχικά εμπειρικά δεδομένα για την χρηστικότητα του Moto-Vault.

Ο μέσος όρος 82 δείχνει ότι οι χρήστες έμειναν ικανοποιημένοι από την εφαρμογή. Οι επιπλέον ερωτήσεις και τα σχόλιά τους έδειξαν ότι άλλο πράγμα είναι η ευχρηστία και άλλο η πρόθεσή τους να τη χρησιμοποιούν συστηματικά στο μέλλον.

Αυτό το εύρημα είναι σημαντικό, καθώς δείχνει ότι η αποδοχή ενός app δεν κρίνεται μόνο από το πόσο εύχρηστο είναι το UI/UX, αλλά από το κατά πόσο καταφέρνει να γίνει μέρος της καθημερινής ρουτίνας του χρήστη.

## 9.4 Οι Περιορισμοί της εργασίας

Η παρούσα εργασία έχει ορισμένους περιορισμούς που αφορούν το λειτουργικό της εύρος, τον βαθμό αυτοματοποίησης, τις εξωτερικές διασυνδέσεις, τη δυνατότητα κλιμάκωσης (scalability) και τη μεθοδολογία αξιολόγησης.

### 9.4.1 Περιορισμένο λειτουργικό εύρος

Η εφαρμογή επικεντρώνεται στην προσωπική διαχείριση πληροφοριών μοτοσυκλέτας από ιδιώτες χρήστες.

Δεν υποστηρίζει ρόλους ή ροές για:
- συνεργεία
- αντιπροσωπείες
- διαχείριση στόλων
- ασφαλιστικές εταιρείες
- εμπορικές συναλλαγές

Η επιλογή αυτή βοήθησε να εστιάσουμε στο βασικό πρόβλημα, αλλά περιορίζει την εφαρμογή μόνο σε προσωπική χρήση.

### 9.4.2 Περιορισμένη αυτοματοποίηση

Η εισαγωγή των περισσότερων δεδομένων πραγματοποιείται χειροκίνητα.

Ο χρήστης πρέπει να καταχωρίζει πληροφορίες για ανεφοδιασμούς, εργασίες συντήρησης, χιλιόμετρα, εξαρτήματα, έγγραφα και κόστη.

Δεν υλοποιήθηκαν μηχανισμοί αυτόματης εξαγωγής από αποδείξεις, φωτογραφίες, OBD, telematics ή εξωτερικές υπηρεσίες.

### 9.4.3 Απουσία mobile εφαρμογής

Το frontend προσαρμόζεται σε κινητές συσκευές, αλλά δεν υπάρχει ξεχωριστή εφαρμογή Android ή iOS.

Για τον λόγο αυτό δεν υποστηρίζονται πλήρως δυνατότητες όπως:
- push notifications
- offline χρήση
- background συγχρονισμός
- βαθύτερη ενσωμάτωση με κάμερα και λειτουργίες συσκευής

### 9.4.4 Περιορισμένες εξωτερικές διασυνδέσεις

Το Moto-Vault λειτουργεί αυτόνομα, καθώς δεν υπάρχει διασύνδεση με συνεργεία, βάσεις δεδομένων κατασκευαστών, ασφαλιστικές υπηρεσίες, δεδομένα τεχνικού ελέγχου, συστήματα OBD, telematics ή υπηρεσίες VIN.

Αυτό περιορίζει τον βαθμό αυτοματοποίησης, επιβεβαίωσης και εμπλουτισμού των καταχωρίσεων.

### 9.4.5 Περιορισμοί μοντέλου δεδομένων και δικαιωμάτων

Το μοντέλο καλύπτει τις ανάγκες προσωπικής χρήσης, αλλά δεν υποστηρίζει πλήρως:
- συνιδιοκτησία
- κοινόχρηστα οχήματα
- ιστορικό μεταβίβασης
- επαγγελματικές εγκρίσεις
- σύνθετους ρόλους
- πλήρες audit trail

Η επέκταση προς συνεργατικά ή επαγγελματικά σενάρια θα απαιτούσε αλλαγές στο μοντέλο δικαιωμάτων και στις επιχειρησιακές ροές.

### 9.4.6 Κλιμάκωση και λειτουργία

Η εργασία δεν περιλάμβανε:
- load tests
- stress tests
- μεγάλο αριθμό ταυτόχρονων χρηστών
- πολύ μεγάλο όγκο αρχείων
- μακροχρόνια παρακολούθηση παραγωγικής λειτουργίας

Επομένως, δεν μπορεί να εξαχθεί οριστικό συμπέρασμα για τη συμπεριφορά του συστήματος σε μεγάλη κλίμακα. Η υποδομή single VPS δεν προσφέρει από μόνη της high availability.

### 9.4.7 Περιορισμοί αξιολόγησης

Οι περιορισμοί της αξιολόγησης παρουσιάστηκαν αναλυτικά στην Ενότητα 8.6. Αφορούν το μικρό δείγμα ευκολίας, την απουσία κοινών εργασιών και αντικειμενικών μετρήσεων και τη μη πραγματοποίηση μακροχρόνιας αξιολόγησης.

### 9.4.8 Απουσία συγκριτικής αξιολόγησης

Δεν πραγματοποιήθηκε άμεση πειραματική σύγκριση με άλλη εφαρμογή του ίδιου πεδίου. Για τον λόγο αυτό, η εργασία δεν υποστηρίζει ότι το Moto-Vault υπερέχει ποσοτικά έναντι συγκεκριμένου ανταγωνιστικού συστήματος.

### 9.4.9 Συντήρηση και εξέλιξη

Η εφαρμογή βασίζεται σε σύγχρονα frameworks, tech stack και βιβλιοθήκες, τα οποία απαιτούν συνεχή ενημέρωση.

Η μακροχρόνια λειτουργία προϋποθέτει:
- αναβαθμίσεις dependencies
- παρακολούθηση ευπαθειών
- έλεγχο συμβατότητας
- συντήρηση migrations
- ενημέρωση tests
- δοκιμές backup και restore

Η εργασία δεν καλύπτει πλήρη κύκλο πολυετούς λειτουργίας και συντήρησης.

## 9.5 Μελλοντικές επεκτάσεις

Η εξέλιξη του MotoVault μπορεί να κινηθεί σε διάφορες κατευθύνσεις. Με βάση τα σχόλια των χρηστών και τις τεχνικές ελλείψεις που εντοπίστηκαν κατά το development, οι μελλοντικές προσθήκες χωρίζονται σε άμεσες βελτιώσεις και σε βαθύτερες αναδιαρθρώσεις της αρχιτεκτονικής.

### 9.5.1 Αυτοματοποίηση καταχωρίσεων

Μία από τις σημαντικότερες βελτιώσεις είναι η μείωση της χειροκίνητης εισαγωγής δεδομένων.
Στην παρούσα έκδοση, ο χρήστης βάζει μόνος του τις πληροφορίες για ανεφοδιασμούς, εργασίες συντήρησης, χιλιομετρικές ενδείξεις, έγγραφα και έξοδα. Παρότι η διαδικασία αυτή επιτρέπει πλήρη έλεγχο των δεδομένων, μπορεί να λειτουργήσει αποτρεπτικά στη μακροχρόνια χρήση της εφαρμογής.

Σε επόμενο βήμα θα μπορούσαν να υποστηριχθούν:
- αναγνώριση αποδείξεων/αντλίας καυσίμων μέσω OCR
- εξαγωγή ημερομηνίας, ποσότητας καυσίμου και συνολικού κόστους
- αναγνώριση χιλιομετρικής ένδειξης από φωτογραφία
- προσυμπλήρωση εγγραφών ανεφοδιασμού
- εξαγωγή στοιχείων από έγγραφα και αποδείξεις συνεργείων
- προτάσεις κατηγοριοποίησης εξόδων

Τα δεδομένα που εξάγονται αυτόματα δεν θα πρέπει να αποθηκεύονται χωρίς έλεγχο. Η εφαρμογή θα πρέπει να προσυμπληρώνει τις αντίστοιχες φόρμες και να ζητά επιβεβαίωση ή διόρθωση από τον χρήστη και αυτό γιατί η αυτόματη αναγνώριση μπορεί να δημιουργήσει λανθασμένα δεδομένα.

### 9.5.2 Εφαρμογή για κινητές συσκευές

Παρότι η διεπαφή του Moto-Vault προσαρμόζεται σε διαφορετικά μεγέθη οθόνης, δεν υπάρχει ξεχωριστή εφαρμογή για Android ή iOS.

Μια mobile έκδοση θα μπορούσε να διευκολύνει την καθημερινή καταχώριση πληροφοριών στο πρατήριο, στο συνεργείο ή κατά τη διάρκεια ταξιδιού. Θα μπορούσε επίσης να προσφέρει:

- push notifications
- καλύτερη αξιοποίηση της κάμερας
- γρήγορη μεταφόρτωση φωτογραφιών και εγγράφων
- offline πρόσβαση σε επιλεγμένα δεδομένα
- συγχρονισμό στο παρασκήνιο
- ευκολότερη καταγραφή ανεφοδιασμών
- άμεση πρόσβαση στις υπενθυμίσεις

Μια λύση θα ήταν να βασιστεί σε Progressive Web App ή σε τεχνολογία όπως το Capacitor, αξιοποιώντας το υπάρχον Angular frontend. Με αυτόν τον τρόπο θα μπορούσε να διατηρηθεί μεγάλο μέρος του υπάρχοντος κώδικα χωρίς να απαιτηθεί εξαρχής ξεχωριστή εφαρμογή.

### 9.5.3 Δεδομένα οχήματος και εξωτερικές διασυνδέσεις

Μια σημαντική επέκταση θα ήταν η δημιουργία οργανωμένης βάσης δεδομένων με κατασκευαστές, μοντέλα και τεχνικά χαρακτηριστικά μοτοσυκλετών.

Κατά την προσθήκη ενός οχήματος θα μπορούσαν να προσυμπληρώνονται στοιχεία όπως:

- κατασκευαστής
- μοντέλο
- έτος παραγωγής
- κυβισμός
- χωρητικότητα ρεζερβουάρ
- διαστάσεις και πιέσεις ελαστικών
- βασικά διαστήματα συντήρησης
- συμβατά αναλώσιμα και ανταλλακτικά

Η χρήση προκαθορισμένων δεδομένων θα μείωνε τα λάθη και θα βελτίωνε τη συνέπεια των καταχωρίσεων.

Το σύστημα θα μπορούσε να διασυνδεθεί με εξωτερικές υπηρεσίες, όπου υπάρχουν διαθέσιμα και νόμιμα προσβάσιμα APIs. Ενδεικτικές διασυνδέσεις είναι:

- υπηρεσίες αποκωδικοποίησης VIN (επί πληρωμής άνα χρήση)
- βάσεις ανακλήσεων κατασκευαστών
- επίσημα προγράμματα συντήρησης
- ασφαλιστικές υπηρεσίες
- δεδομένα τεχνικού ελέγχου
- υπηρεσίες τιμών καυσίμων
- πρόσθετες υπηρεσίες χαρτών και σημείων ενδιαφέροντος

Η δυνατότητα αυτή εξαρτάται από τη διαθεσιμότητα, το κόστος και τους όρους χρήσης κάθε εξωτερικής υπηρεσίας. Επομένως, δεν μπορεί να θεωρηθεί εύκολο ότι όλες οι παραπάνω πηγές μπορούν να ενσωματωθούν.

Σε πιο προχωρημένο επίπεδο, το MotoVault θα μπορούσε να διασυνδεθεί απευθείας με το όχημα μέσω OBD2 adapters, συσκευών GPS ή αισθητήρων smartphone. Αυτό θα επέτρεπε την αυτόματη ενημέρωση των χιλιομετρικών αποστάσεων, των καταγραφών διαδρομής, καθώς και την αυτόματη λήψη κωδικών βλάβης (fault codes).

Η υλοποίηση τέτοιων λειτουργιών θα απαιτούσε ιδιαίτερη προσοχή στην ασφάλεια, στην ιδιωτικότητα και στη συμβατότητα μεταξύ διαφορετικών κατασκευαστών και μοντέλων.

### 9.5.4 Συνεργεία, κοινή χρήση και πολλαπλοί ρόλοι

Η δημιουργία επαγγελματικών λογαριασμών θα μπορούσε να μετατρέψει το Moto-Vault από προσωπικό εργαλείο σε πλατφόρμα συνεργασίας μεταξύ ιδιοκτητών και συνεργείων.

Ένα συνεργείο θα μπορούσε, μετά από έγκριση του ιδιοκτήτη:

- να συνδέεται με συγκεκριμένο όχημα
- να δημιουργεί ή να επιβεβαιώνει εγγραφές συντήρησης
- να ανεβάζει παραστατικά και φωτογραφίες
- να ενημερώνει τις εργασίες και τα ανταλλακτικά που χρησιμοποιήθηκαν
- να εμφανίζεται σε κατάλογο ή χάρτη
- να δηλώνει αν αποτελεί επίσημη αντιπροσωπεία

Ο ιδιοκτήτης θα πρέπει να διατηρεί πλήρη έλεγχο των δεδομένων που κοινοποιούνται, της διάρκειας πρόσβασης και της δυνατότητας ανάκλησης.

Στην ίδια κατεύθυνση θα μπορούσαν να προστεθούν διαφορετικοί ρόλοι, όπως:

- ιδιοκτήτης
- συνιδιοκτήτης
- προσωρινός χρήστης
- μηχανικός
- διαχειριστής στόλου

Η προσθήκη πολλών ρόλων θα απαιτούσε πιο σύνθετο μοντέλο εξουσιοδότησης, λεπτομερή δικαιώματα ανά λειτουργία και πλήρες ιστορικό αλλαγών.

Μια ακόμη χρήσιμη δυνατότητα θα ήταν η ελεγχόμενη κοινή χρήση του ιστορικού της μοτοσυκλέτας. Ο χρήστης θα μπορούσε να δημιουργεί:

- προσωρινό σύνδεσμο μόνο για ανάγνωση
- αναφορά για συνεργείο
- αναφορά για υποψήφιο αγοραστή
- πακέτο δεδομένων κατά τη μεταβίβαση του οχήματος
- επιβεβαιωμένο ιστορικό εργασιών

Η λειτουργία αυτή θα μπορούσε να αυξήσει τη διαφάνεια και την πρακτική αξία του οργανωμένου ιστορικού.

### 9.5.5 Προηγμένες ειδοποιήσεις

Το υπάρχον σύστημα υπενθυμίσεων μπορεί να επεκταθεί με περισσότερους τρόπους ειδοποίησης και πιο σύνθετους κανόνες.

Σε επόμενη έκδοση θα μπορούσαν να προστεθούν:

- αποστολή email
- push notifications
- ειδοποιήσεις λήξης εγγράφων
- υπενθυμίσεις βάσει ημερομηνίας
- υπενθυμίσεις βάσει χιλιομέτρων
- συνδυασμός ημερομηνίας και απόστασης
- πρόβλεψη επόμενης συντήρησης
- κανόνες βάσει του μοτίβου χρήσης
- προσαρμοζόμενα χρονικά όρια ειδοποίησης

Η τρέχουσα έκδοση εμφανίζει τις υπενθυμίσεις μέσα στο UI, αλλά δεν αποστέλλει email ή push notifications.

Για να αποφευχθεί η υπερβολική αποστολή ειδοποιήσεων, ο χρήστης θα πρέπει να μπορεί να επιλέγει τον τρόπο, τη συχνότητα και το χρονικό σημείο ειδοποίησης.

### 9.5.6 Προηγμένη ανάλυση και αναφορές

Το Moto-Vault διαθέτει ήδη τα βασικά οικονομικά στατιστικά, αλλά καλό θα ήταν να προστεθούν πιο σύνθετες αναλύσεις όπως:

- πρόβλεψη μελλοντικών εξόδων
- εντοπισμός τάσεων κατανάλωσης
- σύγκριση μεγαλύτερων χρονικών περιόδων
- εκτίμηση επόμενης συντήρησης
- προσαρμοσμένες αναφορές
- σύγκριση διαφορετικών μοτοσυκλετών
- αναλυτικό ετήσιο κόστος ιδιοκτησίας
- νέες κατηγορίες εξόδων, όπως διόδια, ασφάλιση, τέλη και στάθμευση

Αυτές θα μπορούσαν να ενισχύσουν τη χρησιμότητα του dashboard και να βοηθήσουν τον χρήστη να κατανοεί καλύτερα το συνολικό κόστος και τη χρήση κάθε οχήματος.

### 9.5.7 Προσβασιμότητα, γλώσσες και απόδοση πολυμέσων

Η εφαρμογή μπορεί να βελτιωθεί περαιτέρω ως προς την προσβασιμότητα.

Πιθανές επεκτάσεις είναι:

- πληρέστερη χρήση μόνο με πληκτρολόγιο
- καλύτερη συμβατότητα με screen readers
- αυξημένη αντίθεση
- προσαρμογή μεγέθους κειμένου
- σωστά μηνύματα σφάλματος
- συστηματικός έλεγχος σύμφωνα με τις οδηγίες WCAG
- υποστήριξη περισσότερων μη ευρωπαικών γλωσσών

Εργαλεία AI μπορούν να μεταφράσουν εύκολα, αλλά η τελική επιμέλεια πρέπει να πραγματοποιείται από άνθρωπο, ώστε να διασφαλίζεται η ορθότητα της ορολογίας.

Η διαχείριση πολυμέσων μπορεί επίσης να βελτιωθεί μέσω:

- δημιουργίας thumbnails
- συμπίεσης εικόνων
- lazy loading

- responsive image sizes
- caching
- χρήσης CDN όταν προκύψει πραγματική ανάγκη

Οι αλλαγές αυτές θα βελτίωναν ιδιαίτερα την εμπειρία χρήσης από κινητές συσκευές και συνδέσεις χαμηλότερης ταχύτητας.

### 9.5.8 Ασφάλεια, ποιότητα και τεχνική κλιμάκωση

Πριν από ευρεία δημόσια διάθεση θα ήταν χρήσιμο να πραγματοποιηθούν πρόσθετοι έλεγχοι ασφάλειας και ποιότητας, όπως:

- ανεξάρτητο penetration test
- static code analysis
- επαναλαμβανόμενοι έλεγχοι OWASP Top 10
- έλεγχος security headers
- περιοδικός έλεγχος δικαιωμάτων αρχείων
- δοκιμές rate limiting
- έλεγχοι production configuration
- δοκιμές ανάκτησης από backup

Οι έλεγχοι αυτοί θα συμπλήρωναν τους υπάρχοντες μηχανισμούς CI/CD, dependency scanning και αυτοματοποιημένων δοκιμών.

Σε περίπτωση σημαντικής αύξησης του αριθμού χρηστών ή του όγκου δεδομένων, θα μπορούσαν να εξεταστούν:

- caching
- CDN για στατικά αρχεία και πολυμέσα
- οριζόντια κλιμάκωση
- read replicas της βάσης δεδομένων
- βελτιστοποίηση queries και indexes
- προηγμένο monitoring
- disaster recovery
- αυτοματοποιημένα backups
- τακτικές δοκιμές επαναφοράς

Οι αλλαγές αυτές δεν πρέπει να υλοποιηθούν προληπτικά χωρίς πραγματική ανάγκη. Η τεχνική κλιμάκωση πρέπει να βασίζεται σε μετρήσεις φόρτου, χρόνων απόκρισης και χρήσης πόρων.

### 9.5.9 Ενίσχυση της αξιολόγησης

Η αξιολόγηση του Moto-Vault μπορεί να επεκταθεί σε επόμενη ερευνητική φάση.

Μια πληρέστερη μελέτη θα μπορούσε να περιλαμβάνει:

- μεγαλύτερο και πιο ποικιλόμορφο δείγμα
- συμμετέχοντες με διαφορετικό επίπεδο τεχνικής εξοικείωσης
- προκαθορισμένες εργασίες
- μέτρηση χρόνου ολοκλήρωσης
- καταγραφή λαθών και ποσοστού επιτυχίας
- συνεντεύξεις ή ανοιχτές ερωτήσεις
- μακροχρόνια χρήση για εβδομάδες ή μήνες
- σύγκριση με άλλη εφαρμογή
- επαναληπτική αξιολόγηση μετά από βελτιώσεις

Με αυτόν τον τρόπο θα μπορούσαν να αξιολογηθούν όχι μόνο η αντιλαμβανόμενη χρηστικότητα, αλλά και η αντικειμενική απόδοση, η ευκολία εκμάθησης και η διατήρηση της χρήσης σε μεγαλύτερο χρονικό διάστημα.

### 9.5.10 Προτεραιότητες επόμενων βημάτων

Οι επεκτάσεις διαφέρουν σημαντικά ως προς το κόστος και την πολυπλοκότητα. Μια ρεαλιστική σειρά προτεραιότητας θα μπορούσε να είναι:

- βελτιστοποίηση της mobile εμπειρίας και της φόρτωσης πολυμέσων,
- αποστολή email και push notifications,
- αυτοματοποίηση ανεφοδιασμών μέσω φωτογραφιών
- βάση δεδομένων μοντέλων και τεχνικών χαρακτηριστικών
- εμπλουτισμός των αναφορών και των στατιστικών
- σύνδεση με OBD2 και τράβηγμα δεδομένων από εγκέφαλο οχήματος
- κατάλογος και λογαριασμοί συνεργείων
- σύνδεση VIN και εξωτερικών υπηρεσιών
- ανεξάρτητος έλεγχος ασφάλειας
- τεχνική κλιμάκωση όταν προκύψει πραγματικό φορτίο

Η τελική κατανομή προτεραιοτήτων πρέπει να βασιστεί σε πραγματικά δεδομένα χρήσης, στις ανάγκες των χρηστών, στο feedback και στο τεχνικό κόστος κάθε αλλαγής.

## 9.6 Τελικά συμπεράσματα

Το κύριο πρόβλημα της εργασίας ήταν η κατακερματισμένη διαχείριση πληροφοριών μοτοσυκλέτας μέσω του σχεδιασμού και της ανάπτυξης του Moto-Vault.

Η εφαρμογή μαζεύει τις βασικές κατηγορίες δεδομένων σε ένα ενιαίο περιβάλλον και υποστηρίζεται από αρχιτεκτονική που δίνει έμφαση στην ασφάλεια, την ιδιωτικότητα και τη συντηρησιμότητα.

Οι αρχικές δοκιμές έδειξαν ότι η εφαρμογής ήταν εύχρηστη, παρόλο που τα συμπεράσματα περιορίζονται στο συγκεκριμένο δείγμα συμμετεχόντων.

Η υλοποίηση πέτυχε τους στόχους της εντός του καθορισμένου scope, αφήνοντας το MotoVault έτοιμο για το μέλλον.

# 10. Βιβλιογραφία

## Επιστημονικές πηγές

Brooke, J. (1996). SUS: A “quick and dirty” usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis. https://doi.org/10.1201/9781498710411-35

Chen, F., Shang, D., Zhou, G., Ye, K., & Ren, F. (2024a). Mileage-aware for vehicle maintenance demand prediction. *Applied Sciences, 14*(16), Article 7341. https://doi.org/10.3390/app14167341

Chen, F., Shang, D., Zhou, G., Ye, K., & Wu, G. (2024b). Multi-source data fusion for vehicle maintenance project prediction. *Future Internet, 16*(10), Article 371. https://doi.org/10.3390/fi16100371

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28* (1), 75–105. https://doi.org/10.2307/25148625

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Sauro, J. (2011). *A practical guide to the System Usability Scale: Background, benchmarks & best practices*. Measuring Usability LLC.

## Πρότυπα και τεχνικές προδιαγραφές

International Organization for Standardization. (2016). *ISO 14224:2016: Petroleum, petrochemical and natural gas industries. Collection and exchange of reliability and maintenance data for equipment*. https://www.iso.org/standard/64076.html

Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT)* (RFC 7519). Internet Engineering Task Force. https://doi.org/10.17487/RFC7519

Nottingham, M., Wilde, E., & Dalal, S. (2016). *Problem details for HTTP APIs* (RFC 7807). Internet Engineering Task Force. https://doi.org/10.17487/RFC7807

Topografix. (2004). *GPX 1.1 schema documentation.* https://www.topografix.com/gpx/1/1/

## Θεσμικές πηγές

European Commission. (2025a, April 24). *Updated rules for safer roads, less air pollution and digital vehicle documents*. https://transport.ec.europa.eu/news-events/news/updated-rules-safer-roads-less-air-pollution-and-digital-vehicle-documents-2025-04-24_en

European Commission. (2025b, April 23). *Questions and answers on the revision of the roadworthiness package*. https://ec.europa.eu/commission/presscorner/detail/en/qanda_25_1084

## Υφιστάμενες εφαρμογές και υπηρεσίες

Drivvo. (χ.χ.*). Vehicle tracker for fuel, expenses and maintenance*. https://www.drivvo.com/en-US/personal-use/

Fuel Log. (χ.χ.). *Fuel Log: Gas Expenses Tracker*. https://play.google.com/store/apps/details?id=com.andrzejsalwin.motorcyclefuellogfree

Motorfolio. (χ.χ.). *Πρόγραμμα διαχείρισης συνεργείου και ψηφιακό βιβλίο service*. https://motorfolio.gr/

MotorManage. (χ.χ.). *Motorcycle maintenance app.* https://motormanage.app/app

Motohub. (χ.χ.). *Motohub Moto companion*, https://play.google.com/store/apps/details?id=ru.gilgamesh.abon.motot

MotoLogger. (χ.χ.). *MotoLogger: Motorcycle maintenance, AI chat and document vault*. https://motologger.ie/features

MotoVault Motorcycle Garage. (χ.χ.*). Motorcycle trip planning, maintenance tracking and AI diagnostics*. https://motovault.app/

RideLog. (χ.χ.). *Motorcycles tracker for iPhone and Android*. https://ridelog.it/en/

Rydful. (χ.χ.). *Ride tracker, fuel logbook and maintenance app*. https://rydful.com/

Στρίφτεροι. (χ.χ.). *Ελληνική εφαρμογή για αναβάτες μοτοσυκλέτας.* https://strifteroi.gr/