



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΠΡΟΠΤΥΧΙΑΚΟ
ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΚΕΝΤΡΟ ΕΡΕΥΝΑΣ ΓΙΑ
ΤΗΝ ΗΛΕΚΤΡΟΝΙΚΗ
ΔΙΑΚΥΒΕΡΝΗΣΗ



Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου

Λιάρης Θωμάς

321/2015107, icsd15107@icsd.aegean.gr

Ντρούτσης Κωνσταντίνος

321/2014141, icsd14141@icsd.aegean.gr

Επιβλέπων: Καθηγητής, Χαραλαμπίδης Ιωάννης

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση
Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου

Η Διπλωματική Εργασία παρουσιάστηκε ενώπιον
του Διδακτικού Προσωπικού του Πανεπιστημίου Αιγαίου
Σε Μερική Εκπλήρωση των απαιτήσεων για την απόκτηση του διπλώματος του
Μηχανικού Πληροφοριακών και Επικοινωνιακών Συστημάτων

Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ
ΕΠΙΚΥΡΩΝΕΙ ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΩΝ
ΛΙΑΡΟΣ ΘΩΜΑΣ
ΝΤΡΟΥΤΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Χαραλαμπίδης Ιωάννης, Επιβλέπων

Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

Λουκής Ευριπίδης, Μέλος

Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

Κοκολάκης Σπύρος, Μέλος

Καθηγητής

Τμήμα Μηχανικών Πληροφοριακών
και Επικοινωνιακών Συστημάτων

© 2021

Με επιφύλαξη παντός δικαιώματος.

ΛΙΑΡΟΣ ΘΩΜΑΣ – ΝΤΡΟΥΤΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Πανεπιστήμιο Αιγαίου

Πρόλογος

Η πτυχιακή εργασία που μας ανατέθηκε δεν ήταν απλά μια εργασία για τα πλαίσια της σχολής αλλά μας έδωσε μια ιδέα πως θα είναι το μέλλον μας, δηλαδή η συνεργασία με άλλους συνάδελφους και η αφοσίωση σε ένα σημαντικό project μηνών. Παρά τις δυσκολίες της εποχής που περνάμε καταφέραμε να συνεργαστούμε με άτομα από διαφορετικές γωνιές της Ελλάδας και να την φέρουμε εις πέρας τον στόχο μας σε έναν ικανοποιητικό βαθμό.

Θα θέλαμε σε αυτό το σημείο να ευχαριστήσουμε τον κύριο Ιωάννη Χαραλαμπίδη για το ενδιαφέρον του να μας εγκλιματίσει στα πλαίσια της εφαρμογής και να μας βοηθήσει να αρχίσουμε αυτό το σημαντικό έργο. Επίσης θα θέλαμε να ευχαριστήσουμε την κυρία Ζωή Λαχανά για την προσφορά της για βοήθεια σε ότι ζήτημα αντιμετωπίσαμε και την καθοδήγηση μας προς την τελική παρουσίαση.

Περίληψη

Η διπλωματική μας προσφέρει μια εναλλακτική λύση στις πρώτες βοήθειες που παρέχει το κράτος. Σε αρκετές περιοχές της Ελλάδας υπάρχουν ελλιπή μέσα πρώτης βοήθειας για περιστατικά σε εξωτερικό χώρο, όπως έλλειψη προσωπικού ή ασθενοφόρων. Επίσης υπάρχουν δυσπρόσιτες περιοχές και απομακρυσμένα σπίτια ή συνοικισμοί που δεν διαθέτουν το δικό τους κέντρο υγείας με αποτέλεσμα μια μεγάλη έκταση να έχει ένα κεντρικό σημείο πρώτων βοηθειών. Έτσι σε περίπτωση κάποιου προβλήματος υγείας ή ενός τραυματισμού, ο κίνδυνος να αυξηθεί λόγω αναμονής ή μη έγκαιρης αντιμετώπισης.

Πλέον έχει παρθεί η πρωτοβουλία από διάφορους δημότες να κάνουν δωρεές απινιδωτών που τοποθετούνται σε διάφορα σημεία δημόσια. Παράλληλα υπάρχουν εθελοντές διασώστες που θεωρητικά αλλά και πρακτικά εκπαιδεύονται στην χρήση αυτών των απινιδωτών. Κάπως έτσι γεννήθηκε η ανάγκη για τον συντονισμό των γεγονότων, την ειδοποίηση των διασωστών και κατ' επέκταση της χρήσης αυτών των απινιδωτών όπου είναι δυνατόν. Η εφαρμογή μας είναι η δικιά μας προσέγγιση στην λύση αυτού το προβλήματος.

Έχουμε δημιουργήσει μια πλατφόρμα όπου ο καθένας μπορεί να αναφέρει κάποιο γεγονός που συνέβη είτε στον ίδιο είτε σε κάποιον συνάνθρωπο του. Ο χρήστης έπειτα από μια σύντομη συμπλήρωση φόρμας αναφέρει το συμβάν, που με την σειρά του διαμοιράζεται στους ενεργούς διασώστες. Οι τελευταίοι έχουν την δυνατότητα να αναλάβουν το συμβάν. Έπειτα η εφαρμογή τους ενημερώνει μέσω ενός χάρτη που βρίσκεται το γεγονός και αναζητεί τους πιο κοντινούς απινιδωτές από αυτό. Έτσι εν μέσω μιας κατάστασης που ο χρόνος είναι πολύτιμος ο διασώστης μπορεί να ξέρει μέσα σε λίγα δευτερόλεπτα που πρέπει να κατευθυνθεί ώστε να βοηθήσει.

Λέξεις - Κλειδιά: *συμβάν, απινιδωτής, διασώστης, γεγονός, χάρτης, πρώτες βοήθειες, εφαρμογή*

Abstract

Design and Development of an Original Application in Mobile Devices for the Management of Defibrillators, with application in Karlovasi Samos

Our thesis offers an alternative solution to first aid provided by the state. In several areas of Greece there is a lack of first aid for incidents outside, such as lack of staff or ambulances. There are also inaccessible areas and remote houses or settlements that do not have their own health center resulting in a large area having a central first aid point. Thus, in case of a health problem or an injury, the risk increases due to waiting or untimely treatment.

Now the initiative has been taken by various citizens to make defibrillator donations that are placed in various places in public. At the same time, there are volunteer rescuers who are theoretically and practically trained in the use of these defibrillators. Somehow the need arose for the coordination of events, the notification of rescuers and consequently the use of these defibrillators where possible. Our application is our approach to solving this problem.

We have developed a platform where anyone can report an event that happened to either themselves or a fellow human being. The user after a short completion of a form reports the event, which in turn is shared with the active rescuers. The latter have the opportunity to take over the event. The application then informs them through a map where the event is located and searches for the nearest defibrillators from it. So, in the midst of a situation where time is precious the rescuer within seconds will be informed where he must be directed to help.

Keywords: *health problem, first aid, defibrillators, rescuers, event, platform, map*

Πίνακας Περιεχομένων

1.	Εισαγωγή	8
1.1	Υγεία, δημόσιοι απινιδωτές και διαχείριση τους	8
1.2	Εφαρμογή για έκτακτα συμβάντα.....	8
1.3	Στόχος και αντικείμενο διπλωματικής	9
1.4	Δομή της διπλωματικής	10
2.	Ανάλυση πεδίου	11
2.1	Εφαρμογές για απινιδωτές στον κόσμο.....	11
2.2	Εφαρμογές για απινιδωτές στην Ελλάδα.....	15
2.3	Συμπεράσματα επί των εφαρμογών	19
3.	Μεθοδολογία	21
3.1	Τα μεθοδολογικά βήματα και βοηθήματα	21
4.	Απαιτήσεις εφαρμογής	23
4.1	Λειτουργικές απαιτήσεις.....	23
4.1.1	Ανάλυση απαιτήσεων	23
4.1.2	Απαιτήσεις για μια πλατφόρμα	23
4.1.3	Οι απινιδωτές	23
4.1.4	Συμβάν.....	24
4.1.5	Πληροφορίες και προφίλ του χρήστη	25
4.1.6	Ρόλοι του χρήστη στην εφαρμογή	26
4.2	Μη λειτουργικές απαιτήσεις.....	34
4.2.1	Απόδοση	34
4.2.2	Χρησιμότητα.....	34
4.2.3	Ασφάλεια.....	34
4.2.4	Ιδιωτικότητα	35
4.2.5	Ευελιξία - Δυνατότητα Συντήρησης	35
4.2.6	Επαληθευσιμότητα	35
4.2.7	Δυνατότητα επαναχρησιμοποίησης	35
4.2.8	Φορητότητα.....	36
4.2.9	Σημαντικότητα σωστής ανάλυσης απαιτήσεων	38
5.	Αρχιτεκτονική συστήματος και τεχνολογίες προγραμματισμού	39
5.1	Back-end	39
5.2	Front-end	43
5.2.1	Απαραίτητα plugins για την λειτουργικότητα της εφαρμογής	44
5.2.2	Μεταφορά και εμφάνιση σε κινητή συσκευή – PWA.....	45
5.3	Βάση αποθήκευσης δεδομένων	45

5.4 Οι υπηρεσίες ως containers	47
5.5 Flowcharts της εφαρμογής SamosAlive	49
6. Ανάλυση πλατφόρμας.....	59
6.1 Η πρώτη επαφή με την εφαρμογή.....	59
6.2 Η αναφορά ενός συμβάντος	60
6.3 Η ανάθεση ενός γεγονότος σε διασώστη	62
6.4 Το μενού της εφαρμογής	64
6.4.1 Υγεία	66
6.4.2 Προβλήματα	68
6.4.3 Ανακοινώσεις	69
6.4.4 Σελίδα διαχείρισης	71
6.4.5 Γενικές χρήσιμες σελίδες	78
6.4.6 Δευτερεύον μενού για τον χρήστη.....	79
7. Συμπεράσματα και Προοπτικές (στην περίπτωση της Μελετητικής είναι το 6 ^ο Κεφάλαιο).....	83
7.1. Συμπεράσματα επί των αποτελεσμάτων	83
7.2 Συμπεράσματα επί της μεθοδολογίας.....	83
7.3 Συμπεράσματα επί των εργαλείων	83
7.4 Προοπτικές/ Επόμενα Βήματα	84
8. Αναφορές	85
9. Παράρτημα.....	86

1. Εισαγωγή

1.1 Υγεία, δημόσιοι απινιδωτές και διαχείριση τους

Στις μέρες μας αρκετοί άνθρωποι επιλέγουν να αθληθούν σε εξωτερικό χώρο. Αυτό πέρα από τα θετικά που σίγουρα έχει, κρύβει και κινδύνους όπως τραυματισμούς ή υπερβολική κόπωση. Αντίστοιχα θέματα υγείας μπορεί να παρουσιάσει ο καθένας σε μια απλή έξοδο ή δουλειά σε συνάρτηση και με το ιατρικό ιστορικό του. Η αντιμετώπιση όμως τέτοιων συμβάντων δεν είναι πάντα εφικτή σε σύντομο χρονικό διάστημα λόγω έλλειψης προσωπικού ή ασθενοφόρων. Επιπλέον αρκετά σπίτια ή συνοικισμοί είναι απομονωμένα χωρίς ιατρείο με αποτέλεσμα μια αρκετά μεγάλη έκταση και κατά συνέπεια πληθυσμός να υποστηρίζεται από ένα συγκεκριμένο κέντρο υγείας. Το πρόβλημα που δημιουργείται είναι ότι ένας μικρός τραυματισμός ή αδιαθεσία μπορεί λόγω μεγάλης αναμονής να σοβαρέψει και να γίνει κίνδυνος για την ζωή.

Λόγω κάποιων δυσάρεστων γεγονότων στον κόσμο τον τελευταίο καιρό, η κοινωνία σκέφτηκε ότι η χρήση απινιδωτή θα μπορούσε να βοηθήσει κάποιες καταστάσεις και να προλάβει τα χειρότερα. Στην Σάμο συγκεκριμένα αρκετοί δημότες έλαβαν την πρωτοβουλία να συνεισφέρουν οικονομικά για να γίνει αγορά απινιδωτών που τοποθετήθηκαν σε δημόσια σημεία σε όλο το νησί. Παράλληλα υπάρχει μια εν δράση ομάδα διασωστών στο νησί που έχει εκπαιδευτεί πλέον να χρησιμοποιεί αυτούς τους απινιδωτές.

Όμως ακόμα και με αυτές τις ενέργειες εκκρεμούν ακόμα κάποια προβλήματα υπό επίλυση όπως το γεγονός ότι οι πολίτες δεν μπορούν να γνωρίζουν ποιος συμπολίτης του έχει πρόβλημα, τον ακριβή τόπο και ώρα. Ακόμα και αν έχουν όμως γνώση του περιστατικού δεν υπάρχει εύκολη πληροφόρηση στους διασώστες που βρίσκονται ακριβώς όλοι οι απινιδωτές, αν λάβουμε υπόψη ότι προστίθενται συνέχεια περισσότεροι.

1.2 Εφαρμογή για έκτακτα συμβάντα

Η δική μας διπλωματική θα προσπαθήσει να επιλύσει αυτό το θέμα συντονισμού ώστε να εξελίσσεται μια ολοκληρωμένη διαδικασία στην περίπτωση κάποιου συμβάντος. Θέλουμε να συγκεντρώσουμε πληροφορίες για όλους τους τοπικούς απινιδωτές ώστε να υπάρχει άμεση ενημέρωση σε κάποιον που θα τους χρειαστεί. Αυτό γίνεται στατικά την πρώτη φορά που θα ανοίξει η εφαρμογή και από εκεί και πέρα ο διαχειριστής της διπλωματικής ή ακόμα και οι διασώστες θα έχουν την δυνατότητα να προσθέσουν απινιδωτές στο σύστημα μόλις ενημερώνονται από τον Δήμο.

Επιπλέον θέλουμε να επιλύσουμε το πρόβλημα ότι οι διασώστες δεν γνωρίζουν κάθε περιστατικό που συμβαίνει στην πόλη σε ζωντανό χρόνο. Παραδείγματος χάρι μπορεί η τοπική δημόσια υπηρεσία υγείας να έχει ειδοποιηθεί μέσω τηλεφώνου για να ένα συμβάν και να μην είναι σε θέση να ανταπεξέλθει έγκαιρα, όμως πολλοί διασώστες να είναι διαθέσιμοι για προσφορά βοήθειας χωρίς να γνωρίζουν την ύπαρξη του

προβλήματος. Αυτό είναι ένα σημαντικό μέρος της δουλειάς μας, ίσως το πιο σημαντικό καθώς ένας χάρτης με απινιδωτές από μόνος του είναι χρήσιμη πληροφορία χωρίς όμως να μπορεί να συγκριθεί με την βοήθεια που προσφέρει ένας εκπαιδευμένος διασώστης.

Εδώ έρχεται η δική μας πρακτική πλατφόρμα που έχει σκοπό να λύσει όλα αυτά τα ζητήματα. Αρχικά συλλέγοντας πληροφορίες για τους απινιδωτές όπως το μοντέλο τους, την τοποθεσία τους αλλά και το αν είναι διαθέσιμοι για χρήση, μπορούν να έχουμε ένα ολοκληρωμένο προφίλ και κατάσταση για τον καθένα από αυτούς. Έπειτα βάζοντας τους απλούς χρήστες και τους διασώστες μέσα στην εφαρμογή τους διαχωρίζουμε σε ομάδες και αντίστοιχα ο καθένας έχει άλλες λειτουργικότητες. Ο απλός χρήστης να μπορεί να αναφέρει ένα συμβάν με το πάτημα ενός κουμπιού και συμπληρώνοντας μια σύντομη φόρμα για τις απαραίτητες πληροφορίες που χρειάζεται να γνωρίζει κάποιος διασώστης, όπως η διεύθυνση και ο τύπος του περιστατικού. Τέλος θα πρέπει να υπάρχει αντίστοιχη αναφορά σε όλους τους διασώστες που βρίσκονται κοντά στο συμβάν με ειδοποίηση. Όποιος επιλέξει να το αναλάβει τότε ενημερώνεται για τους πιο κοντινούς διαθέσιμους απινιδωτές, σε σχέση με το συμβάν, ώστε να αποφεύγεται άσκοπο χάσιμο χρόνο που είναι πολύτιμος. Με προτεινόμενες οδηγίες για την διεκπεραίωση του περιστατικού πλέον ο συγκεκριμένος διασώστης είναι υπεύθυνος για τον ασθενή και τον απινιδωτή που θα χρησιμοποιήσει.

1.3 Στόχος και αντικείμενο διπλωματικής

Στόχος της διπλωματικής μας: Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μίας εφαρμογής που να λειτουργεί σε υπολογιστές και κινητές συσκευές και να επιτρέπει την σωστή διαχείριση των απινιδωτών που είναι εγκατεστημένοι στο Καρλόβασι Σάμου. Επιθυμητές λειτουργίες είναι εκτός από διαχείριση απινιδωτών η δημιουργία και οργάνωση συμβάντων, η ενημέρωση των διασωστών και οι αναφορές για διάφορα προβλήματα.

Αντικείμενο: Στα πλαίσια της διπλωματικής εργασίας θα σχεδιαστεί και θα αναπτυχθεί πρωτότυπη εφαρμογή που θα έχει τα εξής βασικά χαρακτηριστικά:

- Θα επιτρέπει την εισαγωγή και απεικόνιση σε χάρτη των σημείων όπου είναι εγκατεστημένοι απινιδωτές στο Καρλόβασι Σάμου (με δυνατότητα προσαρμογής σε κάθε πόλη της Ελλάδας)
- Θα επιτρέπει την αλλαγή ρόλου, μετά από κατάλληλη έγκριση, των εκπαιδευμένων χειριστών των απινιδωτών στον ρόλο του «Διασώστη»
- Θα επιτρέπει την χρήση από οποιονδήποτε πολίτη, σε ρόλο άμεσης ενημέρωσης, για εμφάνιση κοντινών απινιδωτών ή δημιουργία αναφοράς συμβάντος σε περίπτωση κάποιου γεγονότος που θα χρειάζεται άμεση επέμβαση
- Σε περίπτωση γεγονότος, θα μπορεί να υποδεικνύει τον πλησιέστερο απινιδωτή και να ειδοποιεί τους εκπαιδευμένους χρήστες, ώστε να σπεύσουν προς βοήθεια

- Σε περίπτωση προβλήματος κάποιου απινιδωτή να μην εμφανίζεται στον χάρτη, για να αποφεύγονται εμπόδια στην άμεση βοήθεια και να αναφέρεται προς επίλυση σε τεχνικούς
- Η εφαρμογή πρέπει να λειτουργεί στα ελληνικά και στα αγγλικά

1.4 Δομή της διπλωματικής

Στο κεφάλαιο 2 θα αναφέρουμε τι υπάρχει ήδη στο χώρο όσο αφορά εφαρμογές για απινιδωτές σε παγκόσμια αλλά και ελληνικά δεδομένα καθώς επίσης τι διαφορετικό προσφέρει η δικιά μας πλατφόρμα. Το κεφάλαιο 3 αναλύει την μεθοδολογία που ακολουθήσαμε για να χτίσουμε την εφαρμογή μας. Το επόμενο στην σειρά κεφάλαιο, 4, αναλύει τις απαιτήσεις που έχει η δημιουργία μιας τέτοια εφαρμογής προτού δημιουργηθεί λαμβάνοντας υπόψη τα υπάρχοντα δεδομένα. Στο Κεφάλαιο 5 αναπτύσσουμε περαιτέρω το πως δουλεύει η εφαρμογή κυρίως σε back-end κώδικα αλλά και σε front-end, δηλαδή το πως συνδέονται και λειτουργούν τα διάφορα μέρη του κώδικα ώστε να λάβουμε αυτό το αποτέλεσμα. Σε συνδυασμό με την αναφορά σε τεχνολογίες που χρησιμοποιήσαμε θα μπορούσαμε να το ονομάσουμε ένα είδος Documentation. Το Κεφάλαιο 6 έχει ως κύριο θέμα να αναλύσει το μενού της εφαρμογής κάνοντας μια «ξενάγηση» στις δυνατότητες και τα χαρακτηριστικά που θα συναντήσει ο χρήστης όταν κατεβάσει το Samos Alive. Εν κατακλείδι αναφέρουμε την σχετική βιβλιογραφία αλλά και κάποια παραρτήματα κώδικα για κάποιες σημαντικές λειτουργίες.

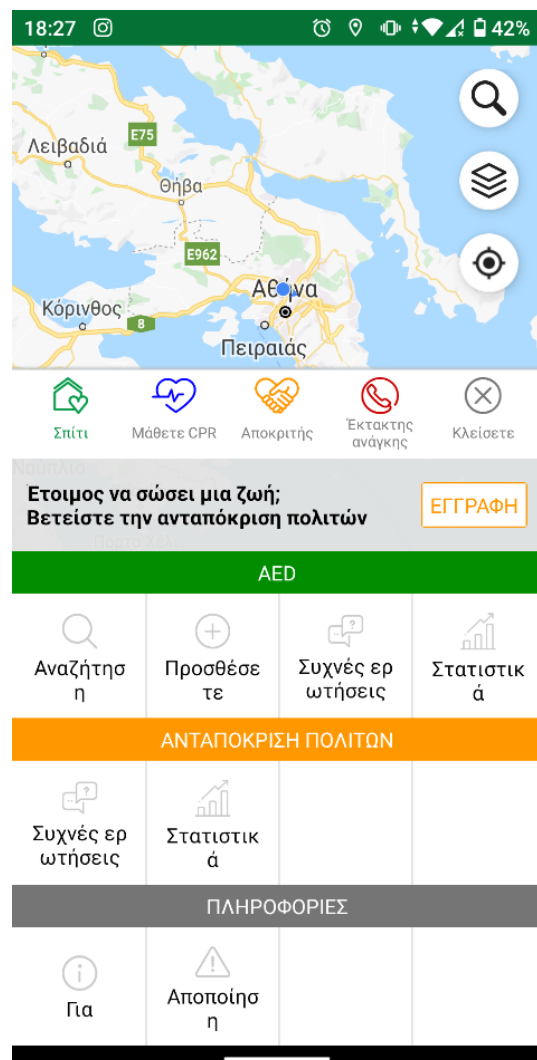
2. Ανάλυση πεδίου

2.1 Εφαρμογές για απινιδωτές στον κόσμο

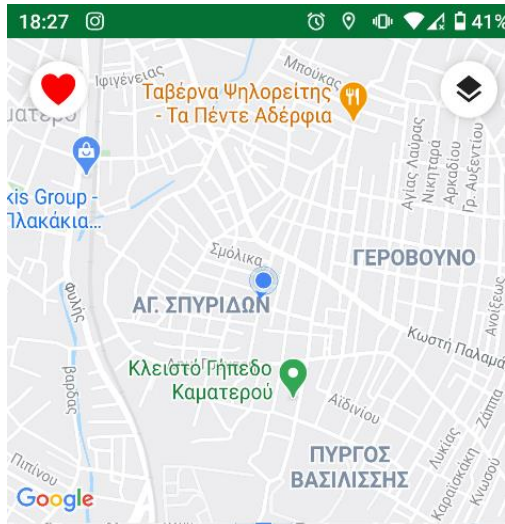
Η εφαρμογή μας στο γενικό πλαίσιο που ανήκει, δηλαδή της υγείας και συγκεκριμένα των απινιδωτών δεν είναι κάτι πρωτοπόρο. Ευτυχώς υπάρχουν ήδη στον χώρο εφαρμογές που κρατούν δεδομένα για απινιδωτές και μπορεί να γίνει αναζήτηση τους στον χάρτη ή προσθήκη συσκευών από απλούς πολίτες. Οι περισσότερες εφαρμογές βέβαια παρέχουν μόνο χάρτες με απινιδωτές. Επίσης υπάρχουν αντίστοιχες εφαρμογές που παρέχουν μόνο/και οδηγό βοήθειας στην χρήση ενός απινιδωτή ή και γενικότερα της αντιμετώπισης τέτοιων συμβάντων με την τεχνική CRP.

Μια πιο εξελιγμένη έκδοση αυτών των εκδόσεων είναι ίσως η πιο διαδεδομένη εφαρμογή για απινιδωτές, η Staying Alive. Η τελευταία προσφέρει εκτός της αναζήτησης απινιδωτή την συνεισφορά των διασωστών. Πιο συγκεκριμένα κάθε πολίτης μπορεί να κάνει αίτηση για να γίνει διασώστης. Όσοι είναι εγκεκριμένοι σε περίπτωση κάποιου κοντινού περιστατικού θα λάβουν ειδοποίηση, από τις αρμόδιες αρχές διασωστών ή πρώτων βοηθειών, για να πάνε στο συμβάν να ξεκινήσουν CRP μέχρι να έρθουν οι πρώτες βοήθειες. Περιέχει γεωγραφικές πληροφορίες για απινιδωτές από όλον τον κόσμο και διατίθεται μετάφραση σε δεκαοχτώ γλώσσες, όπου μέσα σε αυτές συμπεριλαμβάνονται και τα ελληνικά.

Επιπλέον δίνει την δυνατότητα σε κάποιον που θέλει να εξερευνήσει επιπλέον τις δυνατότητες εφαρμογής να έχει πρόσβαση σε συχνές ερωτήσεις για τυχόν απορίες που δημιουργήθηκαν. Ένα ακόμα θετικό είναι ότι παρέχει στατιστικά και για τις συσκευές των απινιδωτών αλλά και για τους πολίτες/διασώστες. Για εκείνους που θέλουν να μάθουν πληροφορίες για την εφαρμογή αλλά και για αποποίηση ευθυνών μπορεί κάποιος να διαβάσει συγκεκριμένα τι ισχύει για να είναι σίγουρος.



Εικόνα 2.1: Εφαρμογή Staying Alive

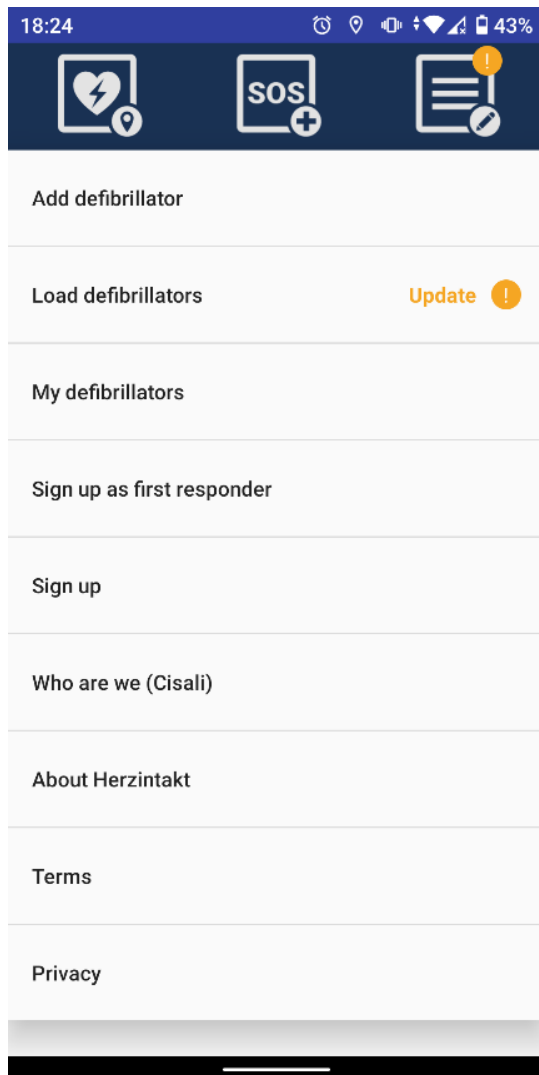


Εικόνα 2.1: Κοντινότερος απινιδωτής από την τοποθεσία μου στο *Staying Alive*

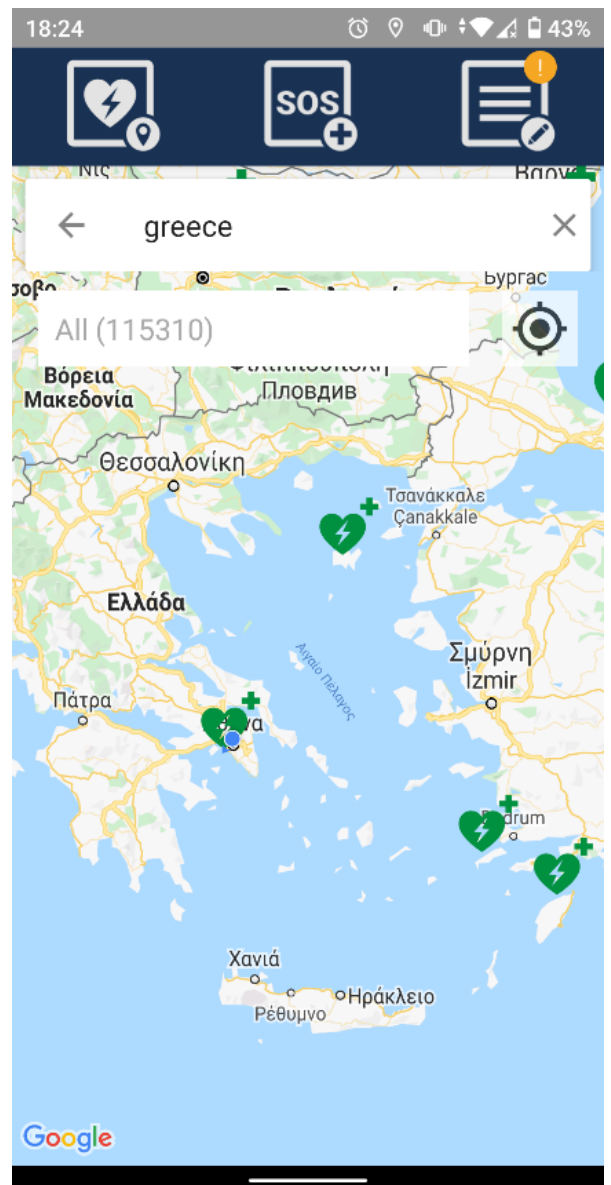
Μια ακόμα εφαρμογή που περιλαμβάνει και τον ρόλο των διασωστών είναι η “Cisali Defibrillator Firstresponder EMC worldwide”. Η “Cisali” (Citizens Save Lives) παρέχει παρόμοιες υπηρεσίες με την προηγούμενη εφαρμογή που αναλύσαμε, πάλι ο χρήστης μπορεί να εγγραφεί ως διασώστης, να προσθέσει απινιδωτή στον χάρτη και να αναζητήσει τον κοντινότερο από την περιοχή του. Επίσης υπάρχει επιλογή για φόρτωση καινούργιων απινιδωτών για να ανανεωθεί ο χάρτης με τα τελευταία δεδομένα, καθώς και έλεγχο πάνω στους απινιδωτές που έχει ανεβάσει ο κάθε χρήστης. Πάλι υπάρχουν βασικοί όροι χρήσεις και προϋποθέσεις. Αυτό βλέπουμε να το υιοθετούν πολλές εφαρμογές γιατί το αντικείμενο των συγκεκριμένων είναι η υγεία, η οποία είναι ένα λεπτό ζήτημα και πρέπει να υπάρχει πλήρης ενημέρωση και συμμόρφωση των πολιτών με τους κανόνες που θέτει κάθε εφαρμογή αλλά και οι νόμοι.

Έχει δύο κουμπιά έκτακτης ανάγκης, το ένα για να καλέσει τις τοπικές πρώτες βοήθειες όπου μέσω της τοποθεσίας του κάθε χρήστη επιλέγει αυτόματα τον κατάλληλο αριθμό, για παράδειγμα στην Ελλάδα το 112. Το δεύτερο ειδοποιεί τους κοντινούς διασώστες. Αυτή η εφαρμογή αριθμεί πάνω από εκατό δεκαπέντε χιλιάδες (115.000) απινιδωτές σε παγκόσμια κλίμακα.

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση
Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου



Εικόνα 2.2: Εφαρμογή Cisali

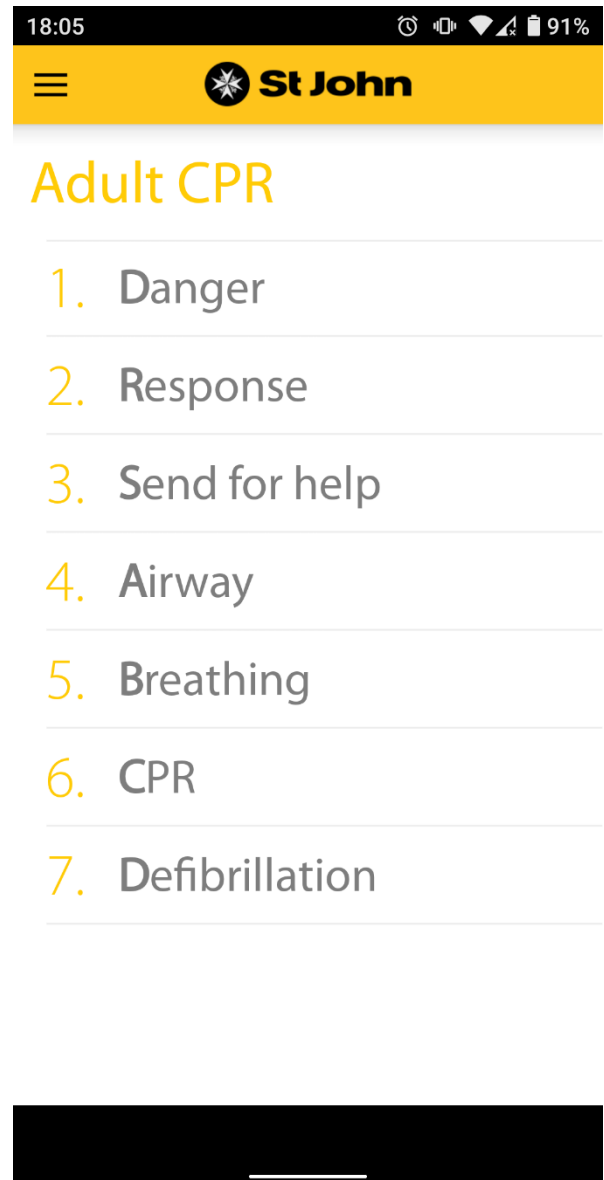


Εικόνα 2.3: Απινιδωτές στην Ελλάδα στην Cisali

Στον κλάδο της υγείας και σχετική με απινιδωτές είναι και η εφαρμογή “St John NZ CPR & AEDs”. Η τελευταία μας προσφέρει χρήσιμες πληροφορίες για καταστάσεις έκτακτης ανάγκης. Συγκεκριμένα παρέχει οδηγίες για την μέθοδο CRP και για την χρήση απινιδωτή με εικόνες, κείμενο αλλά και βίντεο. Δεν περιέχει πληροφορίες για την τοποθεσία κοντινών απινιδωτών αλλά παρόλα αυτά είναι χρήσιμη λύση σε κάποιον που δεν γνωρίζει από πρώτες βοήθειες καθώς περιγράφει βήμα-βήμα την κάθε διαδικασία. Μπορείς να θέσεις συγκεκριμένο τηλέφωνο έκτακτης ανάγκης με βάση την περιοχή σου και υπάρχει link για την αντίστοιχη ιστοσελίδα όπου βρίσκονται στοιχεία για την εφαρμογή και τους προγραμματιστές της.

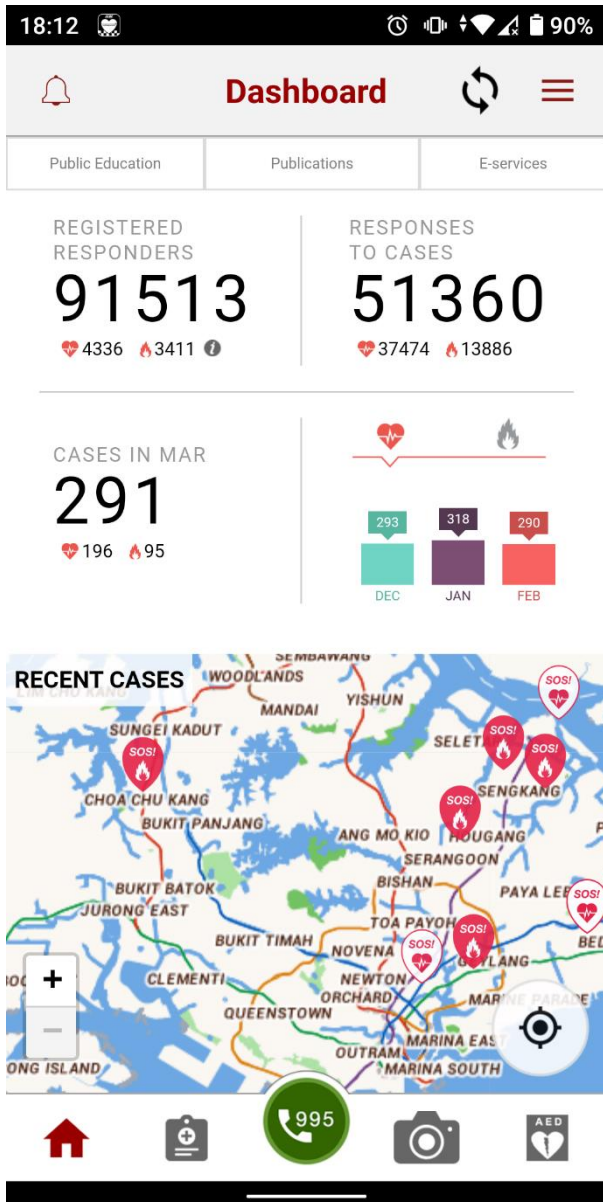


Εικόνα 2.4: St John, αρχική σελίδα

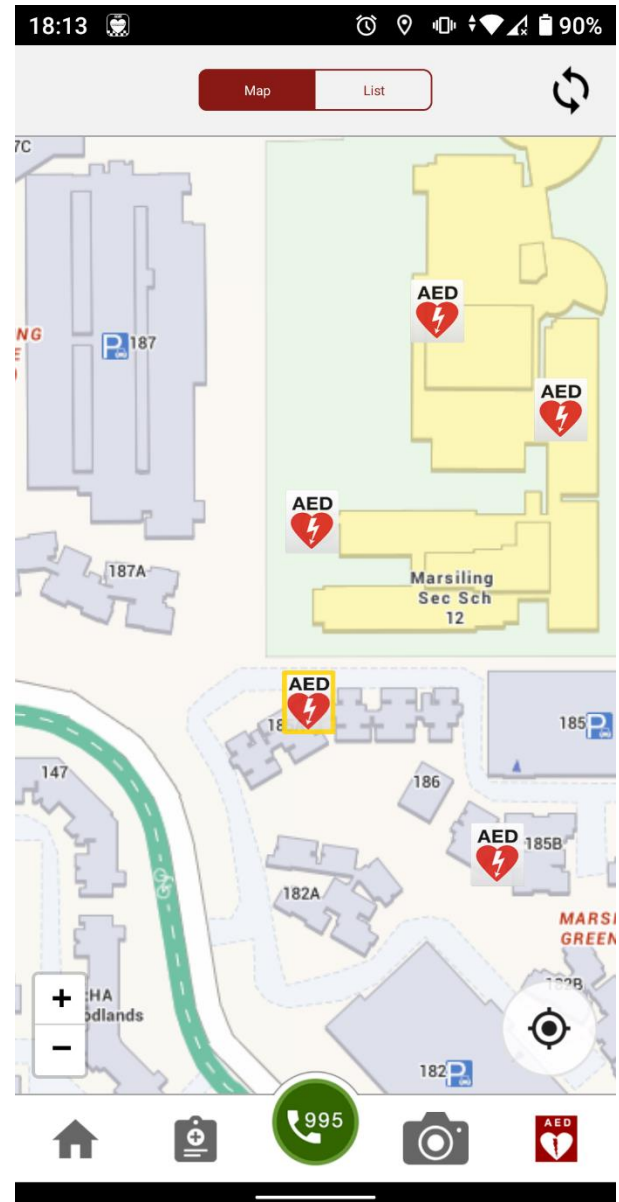


Εικόνα 2.5: St John, Οδηγίες για CPR

Τέλος άλλη μια αξιοσημείωτη εφαρμογή είναι η “myResponder”. Είναι ακόμα υπό ανάπτυξη η εφαρμογή διότι δεν παρέχει παγκόσμιο χάρτη αλλά εστιάζει στην Σιγκαπούρη και γύρω από αυτή. Αυτοί που κατέχουν την εφαρμογή μπορούν να μάθουν μέσω οδηγιών να διαχειρίζονται περιστατικά υγείας αλλά και περιστατικά που απαιτούν την επέμβαση της πυροσβεστικής όπως για παράδειγμα μια φωτιά. Αντίστοιχα κάποιος μπορεί να δημιουργήσει έναν τέτοιο συμβάν και οι εγγεγραμμένοι διασώστες ενημερώνονται για τον τύπο και την τοποθεσία του περιστατικού. Ανάλογα με την εκπαίδευσή τους και τις ικανότητές τους μπορούν να επιλέξουν αν θα πάνε να βοηθήσουν.



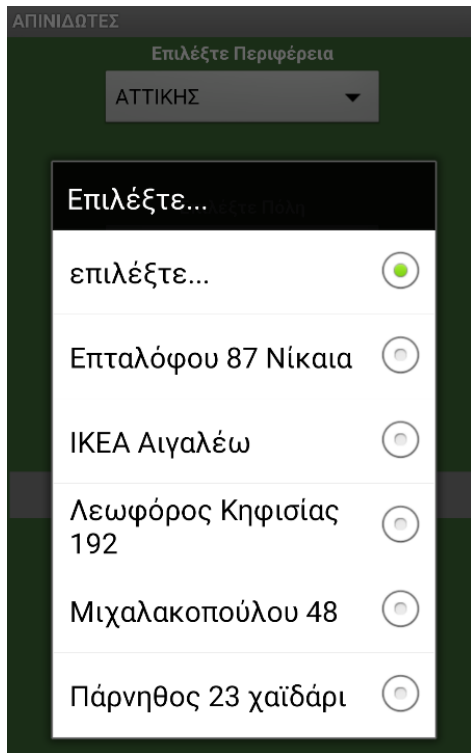
Εικόνα 2.6: Αρχική σελίδα του myResponder με στατιστικά



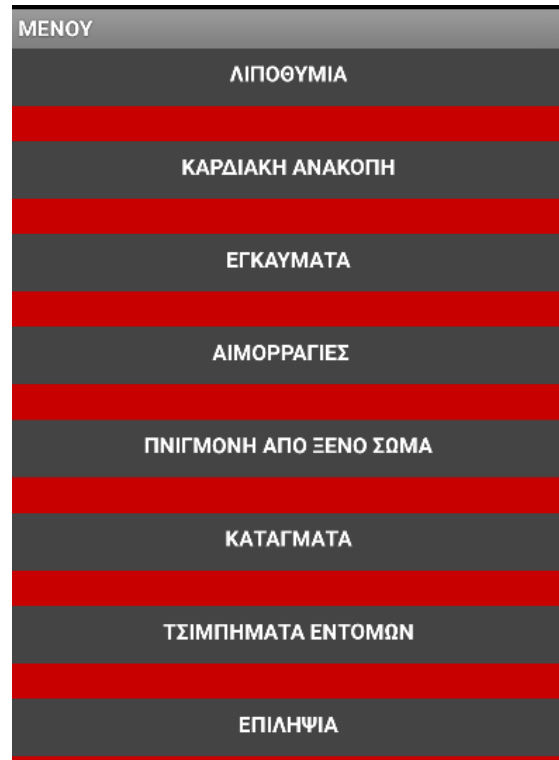
Εικόνα 2.7: myResponder, αναζήτηση απινιδωτή στον χάρτη

2.2 Εφαρμογές για απινιδωτές στην Ελλάδα

Συγκεκριμένα για την Ελλάδα υπάρχουν δύο εφαρμογές για αναζήτηση απινιδωτή. Η πρώτη από αυτές είναι η “Easy to help” την οποία δημιούργησε μια μαθήτρια από την Λάρισα. Το easy for help προσφέρει την δυνατότητα εύρεσης απινιδωτή επιλέγοντας νομό και πόλη ώστε να παρουσιάσει τις διευθύνσεις των απινιδωτών και με ένα πάτημα να οδηγήσει τον χρήστη στο χάρτες της google με εντοπισμό της θέσης του απινιδωτή. Παράλληλα όμως περιέχει και έναν οδηγό πρώτων βοηθειών για διάφορα περιστατικά όπως λιποθυμία και εγκαύματα. Για καθένα από αυτά παρέχει λεπτομερείς οδηγίες με βήματα για τυχόν έκτακτη ανάγκη.

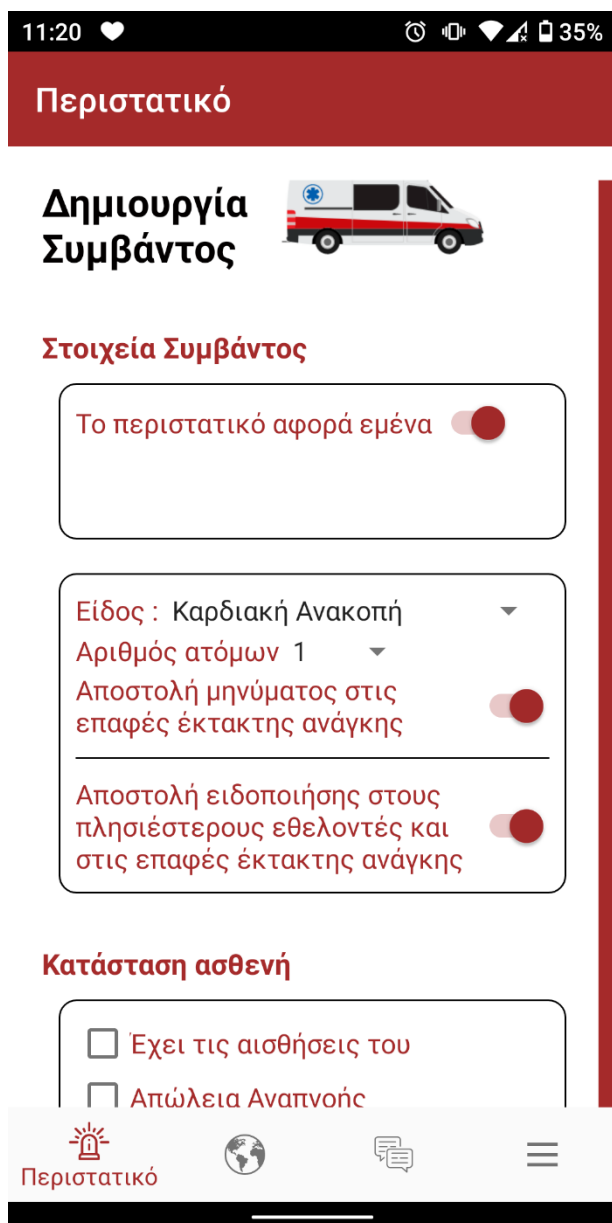


Εικόνα 2.9: Εμφάνιση απινιδωτών σε επιλεγμένη περιοχή "Easy to help"

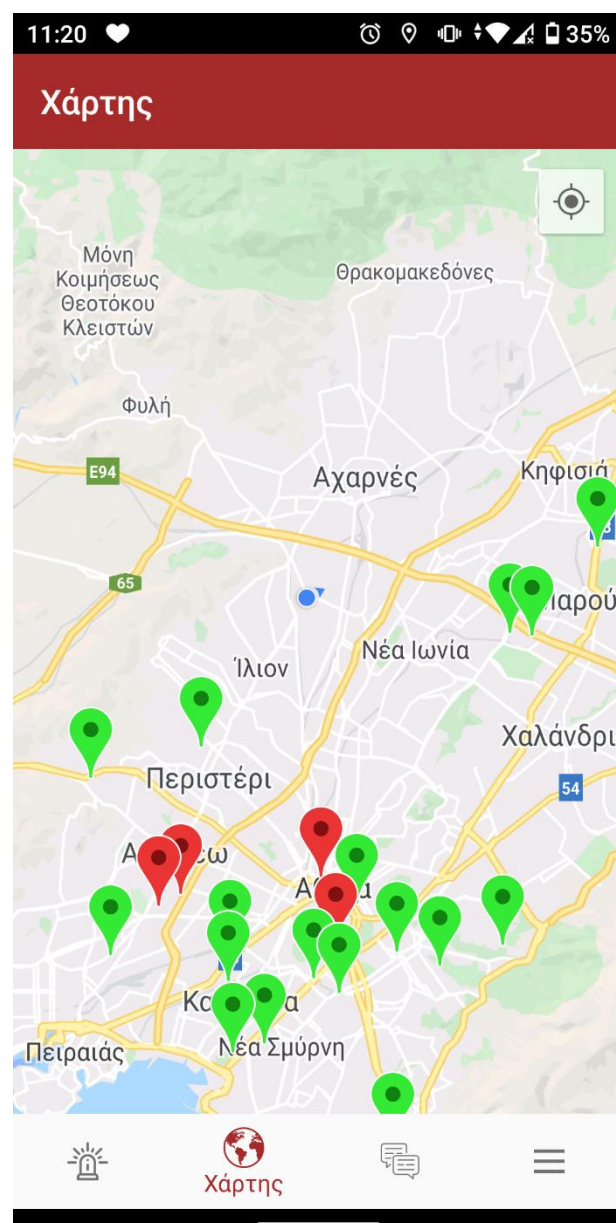


Εικόνα 2.8: Επιπλέον μενού με πρώτες βοήθειες "Easy to help"

Η δεύτερη και πιο ολοκληρωμένη εφαρμογή είναι η “Lifetime HSS”. Η συγκεκριμένη έχει αναπτυχθεί από μια ομάδα νέων φοιτητών του πανεπιστημίου Αιγαίου και περιέχει περισσότερες επιλογές και πληροφορίες για τον χρήστη. Αφότου συνδεθείς στην εφαρμογή υπάρχουν τέσσερις βασικές επιλογές. Η πρώτη είναι μια σελίδα για την αναφορά κάποιου περιστατικού που με κάποιες επιλογές προς συμπλήρωση δίνονται επαρκή στοιχεία για το συμβάν. Οι αρμόδιες αρχές από εκεί και πέρα ενημερώνονται με αυτά τα στοιχεία και μπορούν να βοηθήσουν με μεγαλύτερη ευκολία στο που θα πάνε και τι θα αντιμετωπίσουν. Η δεύτερη σελίδα της εφαρμογής είναι ένας χάρτης που δείχνει που βρίσκονται οι εγγεγραμμένοι απινιδωτές πάνω στον χάρτη. Η Τρίτη περιέχει διάφορες συνομιλίες με φίλους, πρόσφατα μηνύματα και αιτήματα φιλίας. Τέλος υπάρχουν και οι επιλογές για να γίνουν αλλαγές στο προφίλ του κάθε χρήστη καθώς και πληροφορίες για το πλησιέστερο νοσοκομείο, τον πλησιέστερο απινιδωτή και διάφορα κέντρα εκπαίδευσης που όλα αυτά εμφανίζονται με οδηγίες κατεύθυνσης πάνω στον χάρτη.

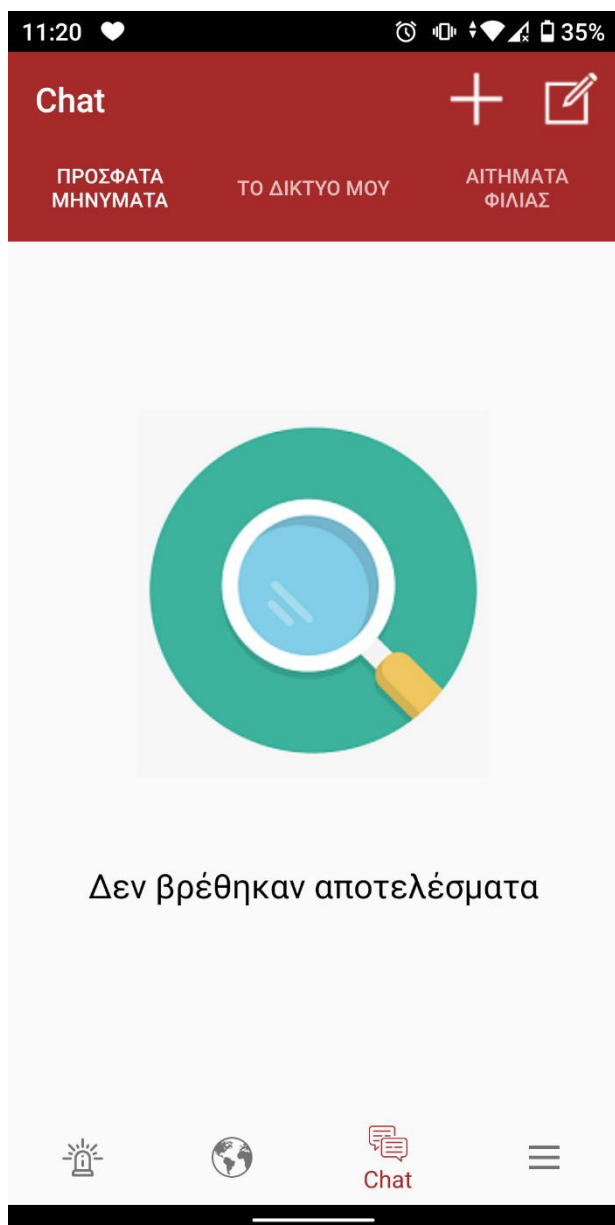


Εικόνα 2.11: Lifetime HSS περιστατικό



Εικόνα 2.12: Lifetime HSS χάρτης απινιδωτών

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου



Εικόνα 2.13: Lifetime HSS chat



Εικόνα 2.14: Lifetime HSS διάφορες επιλογές

	Staying alive	Cisali	St John	myResponder	Easy to help	Lifetime HSS
Χάρτης απινιδωτών	NAI	NAI	OXI	NAI	NAI	NAI
Κοντινότερος απινιδωτής	NAI	NAI	OXI	NAI	NAI	NAI
Αναφορά συμβάντος	NAI	NAI	OXI	OXI	OXI	NAI
Εγγραφή διασωστών	NAI	NAI	OXI	NAI	OXI	NAI
Αυτόματη ενημέρωση διασωστών για συμβάν	NAI (μέσω τρίτων)	NAI	OXI	NAI	OXI	NAI
Τηλέφωνο πρώτων βοηθειών/διασωστών	NAI	NAI	NAI	NAI	NAI	NAI
Οδηγίες πρώτων βοηθειών χρήσης απινιδωτών ή ΚΑΡΠΑ	NAI	OXI	NAI	NAI	NAI	NAI
Κάλυψη επιπλέον περιστατικών εκτός απινιδωτών	OXI	OXI	OXI	NAI	OXI	NAI
Οδηγίες πρώτων βοηθειών για άλλα περιστατικά	OXI	OXI	OXI	NAI	NAI	NAI

Πίνακας 2.1: Σύγκριση υπαρχόντων εφαρμογών

2.3 Συμπεράσματα επί των εφαρμογών

Η καινοτομία μας σε σχέση με τις γενικές εφαρμογές του χώρου είναι ότι ο χρήστης δημιουργεί ένα περιστατικό και αυτό αποστέλλεται στους διασώστες σε ζωντανό χρόνο. Ο άνθρωπος που θα έρθει αντιμέτωπος με ένα πρόβλημα ή κάποιος που θα θέλει να βοηθήσει έναν συνάνθρωπο του δεν θα είναι σε θέση να χρησιμοποιήσει τον απινιδωτή όπως ένας διασώστης. Μια χρήση ενός απινιδωτή χωρίς γνώσεις μπορεί να φέρει τα αντίθετα αποτελέσματα από τα επιθυμητά. Οπότε η εφαρμογή μας εκτός από το να εντοπίζει στον χάρτη που είναι οι απινιδωτές, δίνει την δυνατότητα στους συμπολίτες μας που είναι διασώστες να προσφέρουν τις γνώσεις τους και τις ικανότητες τους ως μία εναλλακτική πρώτη βοήθεια αν αργήσει ή δεν είναι διαθέσιμη η δημόσια υπηρεσία βοήθειας.

Όπως προαναφέρθηκε βέβαια υπάρχουν δύο παρόμοιες εφαρμογές που μπορούν να κάνουν αίτηση για εγγραφή κάποιοι που έχουν γνώσεις χρήσης απινιδωτή. Το πρόβλημα είναι ότι για την Ελλάδα συγκεκριμένα αυτές οι παρόμοιες εφαρμογές δεν έχουν επαρκή δεδομένα απινιδωτών και οι μεταφράσεις είναι από την google. Παράλληλα δεν υπάρχει ένας ενιαίος εύκολος τρόπος να γίνει εγγραφή των διασωστών από μια έγκυρη πηγή ή έναν διαχειριστή ώστε να μην υποβάλλονται ψευδείς αιτήσεις.

Η διαφοροποίηση μας, σε σχέση με τα ελληνικά δεδομένα, είναι ότι όλοι οι διαθέσιμοι διασώστες (ανά περιοχή/δήμο που εγγράφονται) θα λαμβάνουν ειδοποίηση για κάθε συμβάν και αυτά θα φαίνονται σε μια λίστα πραγματικού χρόνου. Ο απώτερος σκοπός της εφαρμογής δεν είναι απλά να εμφανίζει τους απινιδωτές αλλά να βάλει ενεργά τους διασώστες στην αντιμετώπιση τέτοιων γεγονότων, που σε συνδυασμό με τις πρώτες βοήθειες που προσφέρει το κράτος να σώζονται όσες περισσότερες ζωές είναι δυνατόν. Σαφώς δεν υπάρχει αμφιβολία για την χρησιμότητα αντίστοιχων εφαρμογών όμως μια τέτοια πλατφόρμα που θα εστιάζει μόνο στην Ελλάδα είναι λογικό να κατανοεί, να ενσωματώνει και να ανταποκρίνεται καλύτερα στα ελληνικά δεδομένα και τις απαιτήσεις του τόπου μας.

3. Μεθοδολογία

3.1 Τα μεθοδολογικά βήματα και βοηθήματα

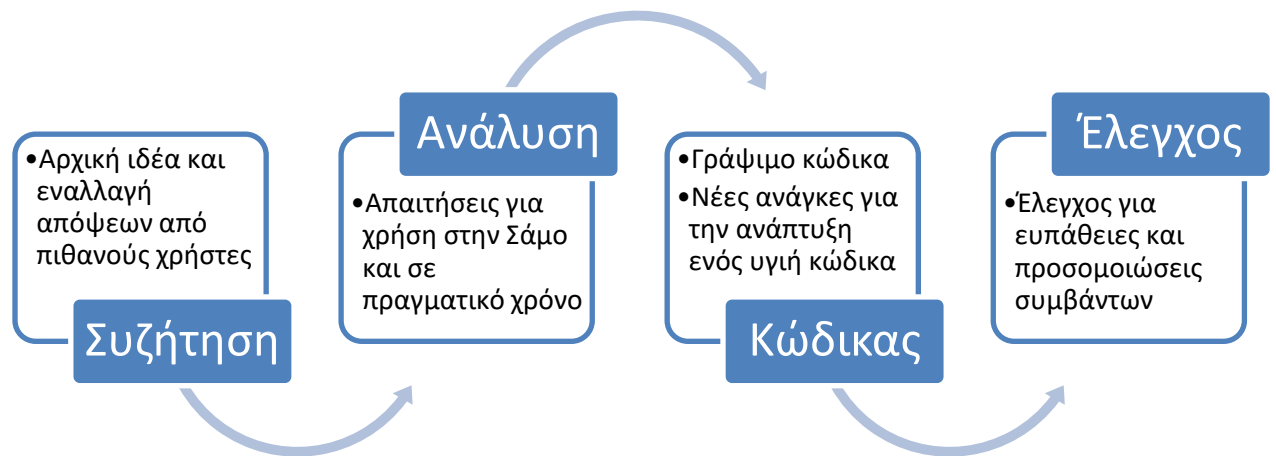
Η ιδέα για αυτήν την διπλωματική προήλθε από τον Κύριο Χαραλαμπίδη λόγω ανάγκης διαχείρισης των δημόσιων απινιδωτών. Έπειτα από μια συνάντηση με την ομάδα διασωστών στην Σάμο είδαμε ότι η εφαρμογή δεν είναι απλά μια καλή ιδέα αλλά είναι απαραίτητη. Εφόσον ρωτήσαμε και μάθαμε για τις βασικές ανάγκες και απαιτήσεις μιας τέτοιας εφαρμογής αποφασίσαμε να ξεκινήσουμε την ανάπτυξη της πτυχιακής μας, το Samos Alive.

Βήμα 1: Εφόσον μας ανατέθηκε μια τέτοια σοβαρή εφαρμογή έπρεπε αρχικά να δούμε πως ξεκίνησε αυτή η ανάγκη για την δημιουργία της και να συζητήσουμε με ανθρώπους του χώρου όπου θα εφαρμοστεί για πιθανές ιδέες και προτάσεις. Προσπαθήσαμε να κατανοήσουμε την διασύνδεση των απλών χρηστών με τους διασώστες μέσω της εφαρμογής.

Βήμα 2: Το δεύτερο βήμα ήταν εφόσον κατανοήσαμε τον πυρήνα του Samos Alive να αναλύσουμε περαιτέρω και διεξοδικά κάθε πτυχή των αναγκών μιας τέτοιας ιδέας. Πέρα από τα βασικά βήματα, δηλαδή την δημιουργία συμβάντος και την λύση του από τους διασώστες υπάρχουν πολλές ακόμα λειτουργίες και απαιτήσεις για την αρμονική χρήση της εφαρμογής. Σκεφτήκαμε δηλαδή ότι προβλήματα σε απινιδωτές πρέπει να αναφέρονται, να επιδιορθώνονται και να ενημερώνουν την κατάσταση των απινιδωτών για να μην γίνεται λάθος συνεννόηση. Ανακοινώσεις για νέους απινιδωτές πρέπει να είναι ορατές στους πολίτες ανεξαρτήτως ρόλου.

Βήμα 3: Πλέον με τις βασικές σκέψεις και σελίδες στο μυαλό μας συνεχίζουμε στην επιλογή κατάλληλων προγραμμάτων για εγγραφή κώδικα. Εφόσον λοιπόν στήσουμε το background έγινε η αρχή μιας client-server εφαρμογής που προστίθεται κάθε αρχική ιδέα βήμα-βήμα. Ένα απλό μενού με τις αρχικές σελίδες έχει αρχίσει να δημιουργείτε και τελικά κάθε φορά που δημιουργούσαμε κάτι παρατηρούσαμε μία νέα ανάγκη στην εφαρμογή. Αυτές οι ανάγκες είχαν να κάνουν με λειτουργίες της εφαρμογής που χωρίς αυτές θα ήταν ελλιπή όταν θα έβγαιναν σε ζωντανή χρήση. Επίσης αναπτύχθηκαν και άλλες ανάγκες λόγω ελαχιστοποίησης ή σωστής διαμόρφωσης κώδικα για να επιτύχει η εφαρμογή της μη-λειτουργικές απαιτήσεις μας παράλληλα με τις λειτουργικές.

Βήμα 4: Με διαρκής ελέγχους και test ελέγχουμε σε κάθε βήμα την εφαρμογή που δημιουργούσαμε. Είτε αυτό σήμαινε να κάνουμε εσκεμμένες αρνητικές πράξεις και να προσπαθούμε να προκαλέσουμε bug στην εφαρμογή, είτε να κάνουμε προσομοιώσεις με απινιδωτές και συμβάντα στις περιοχές μας για να δούμε πόσο εύχρηστη είναι η εφαρμογή σε πραγματική χρήση. Βέβαια λόγω της κατάστασης του κορωνοϊού και τον περιορισμό των επαφών και των μετακινήσεων δεν μπορέσαμε να πραγματοποιήσουμε live test στον δρόμο με πραγματικούς διασώστες όπως κάναμε όσο καλύτερες προσομοιώσεις και ελέγχους μας επιτρέπονταν.



Σχήμα 3.1: Βήματα μεθοδολογίας για ανάπτυξης της εφαρμογής

4. Απαιτήσεις εφαρμογής

4.1 Λειτουργικές απαιτήσεις

4.1.1 Ανάλυση απαιτήσεων

Μια εφαρμογή που αφορά την υγεία και έχει ως σκοπό να την χειρίζεται εύκολα ένα ευρύ κοινό, πρέπει να διαθέτει κάποια οργανωμένη αρχιτεκτονική και να πληρεί κάποιες απαιτήσεις για την σωστή και εύχρηστη λειτουργία της. Αυτές οι απαιτήσεις εμπεριέχουν τις ανάγκες της εφαρμογής ως προς τις λειτουργίες που θα περιέχει, όπως αυτές ορίζονται κατά τον αρχικό σχεδιασμό, με τρόπο που θα είναι κατανοητό και εύχρηστο (λειτουργικές απαιτήσεις). Παράλληλα είναι σημαντικό να περιέχονται θέματα που αφορούν την υλοποίηση του κώδικα και την αρχιτεκτονική που θα ακολουθήσει η εφαρμογή κατά το στήσιμο και την καθημερινή χρήση της (μη λειτουργικές απαιτήσεις).

4.1.2 Απαιτήσεις για μια πλατφόρμα

Λόγω των παραπάνω συμβάντων και δεδομένων δημιουργήθηκε η ανάγκη για μια εφαρμογή που συντονίζει διασώστες και απινιδωτές στην Ελλάδα. Αυτή η εφαρμογή έχει σαν σκοπό να είναι μια παρόμοια λύση με την κλήση στο 166. Χωρίς να έχει σκοπό να αντικαταστήσει τις πρώτες βοήθειες δίνοντας όμως εναλλακτικές δυνατότητες σε κάποιον πολίτη.

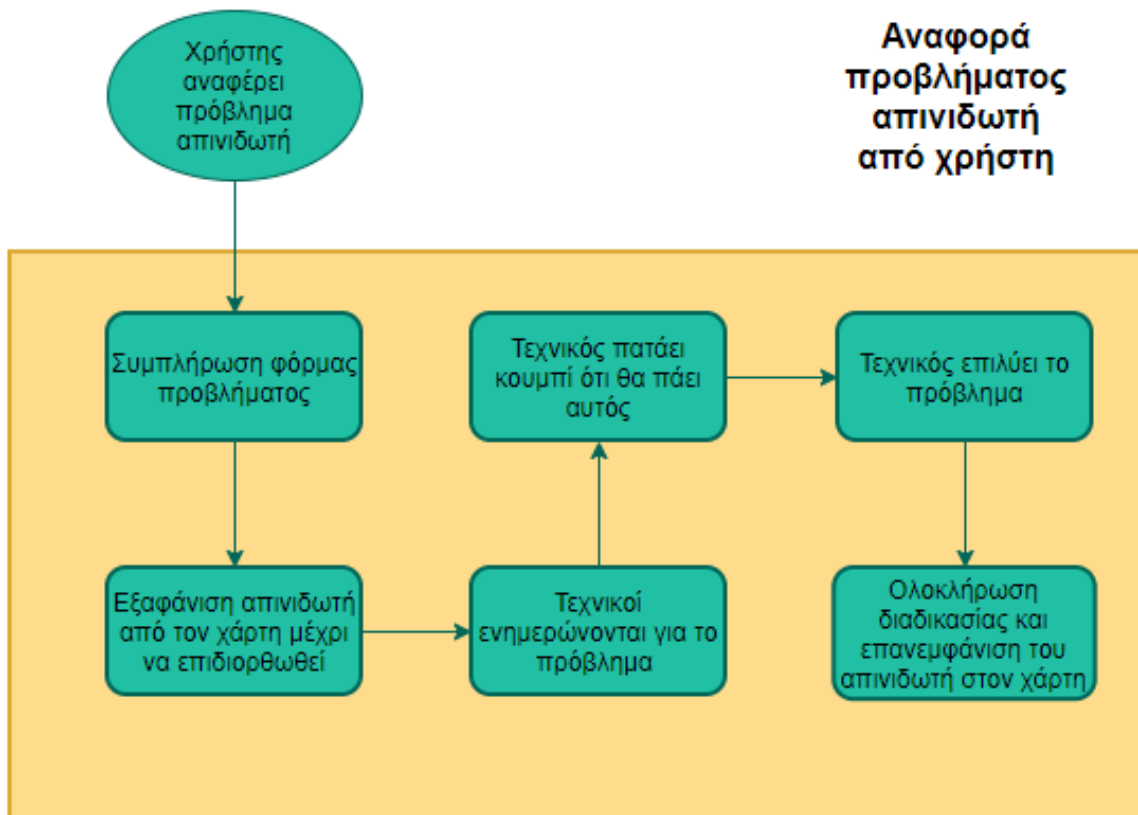
Μια κινητή πλατφόρμα που κατεβαίνει σε όλα τα κινητά είναι η πιο εύκολη και βολική λύση αν αναλογιστούμε ότι η πλειοψηφία των πολιτών έχει πάνω του το κινητό του όλη την μέρα. Σε αυτήν την εφαρμογή θα θέλαμε ανά περιοχή να εγγραφούν οι χρήστες και να αποθηκευτούν οι συντεταγμένες των διαθέσιμων απινιδωτών. Έτσι θα υπήρχε εύκολη πρόσβαση σε όλους αποφεύγοντας όμως τους ανεπιθύμητους-κακόβουλους χρήστες καθώς όλοι θα ταυτοποιούνται.

4.1.3 Οι απινιδωτές

Σκοπός των δημόσιων απινιδωτών είναι να γίνεται και χρήση τους από τους απλούς πολίτες ή τους διασώστες. Άρα εφόσον έχουν συλλεχθεί οι πληροφορίες όπως το που βρίσκονται ή τι τύπου είναι, θα ήταν πιο κατατοπιστικό να τους τοποθετήσουμε μέσα στον τοπικό χάρτη. Αυτό προϋποθέτει να έχουμε μια αρχική λίστα με αυτούς που υπάρχουν ήδη αλλά και κάθε φορά που τοποθετείτε ένας καινούργιος να γίνεται σχετικά γρήγορη εγγραφή του.

Εφόσον γίνουν όλα αυτά θέλουμε να υπάρχει μια απλή λειτουργία όπου ο χρήστης μπορεί να βλέπει τους κοντινούς απινιδωτές στον χάρτη και αντίστοιχα ο διασώστης όταν χρειαστεί σε κάποιο περιστατικό. Εξετάζοντας όλα τα ενδεχόμενα κάποιος απινιδωτής μπορεί να έχει τεχνικό πρόβλημα ή να απουσιάζει από την αρχική του θέση. Έτσι ο κάθε απινιδωτής πρέπει να έχει και μια κατάσταση και καθώς γίνεται αναζήτηση στον χάρτη να γίνεται φιλτράρισμα μόνο των διαθέσιμων προς εμφάνιση.

Το ενδεχόμενο που έχουμε για αλλαγή κατάστασης ενός απινιδωτή και αντίστοιχη εμφάνιση ή εξαφάνιση του από τον χάρτη γίνεται μέσω της διαδικασίας αναφοράς προβλήματος που θέλουμε να έχει την εξής δομή.



Σχήμα 4.2: Αναφορά προβλήματος απινιδωτή από χρήστη

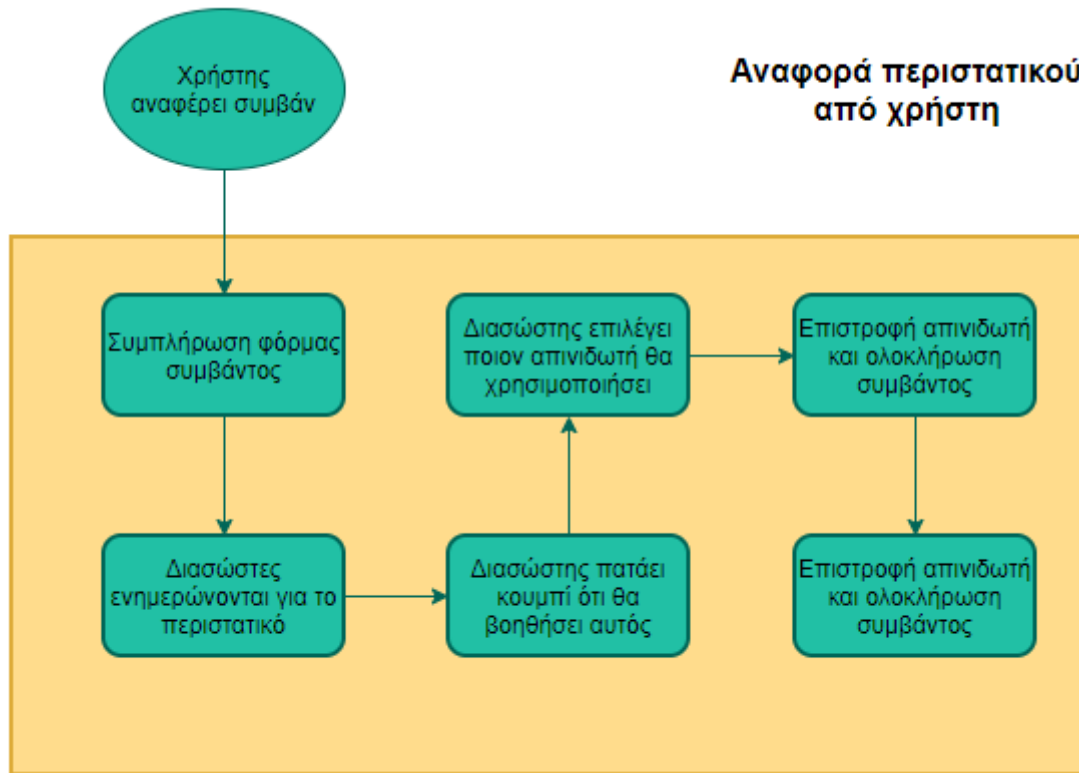
4.1.4 Συμβάν

Ως βασική απαίτηση της εφαρμογής είναι να επιτύχουμε τον στόχο της κάλυψης ενός συμβάντος. Έτσι πρέπει να υπάρχει ένα αντίστοιχο κουμπί που θα σε οδηγήσει σε μία φόρμα στοιχείων για να υποβάλει ο πολίτης. Αυτό θα πρέπει να διαμοιράζεται σε όλους τους διασώστες.

Εφόσον δημιουργηθεί αυτή η αναφορά πρέπει οι διασώστες που ενδιαφέρονται να βοηθήσουν να έχουν πρόσβαση στον χάρτη με τους απινιδωτές. Αυτός ο χάρτης βέβαια πρέπει πλέον να είναι σχετικός και με το σημείο του περιστατικού. Δηλαδή δεν μας ενδιαφέρει να παρουσιάσουμε άσκοπη πληροφορία στον χρήστη αλλά να τον κατατοπίσουμε σε κάτι που να του φανεί χρήσιμο.

Μία ακόμα ιδέα ήταν να υπάρχει μια ομαδική συνομιλία για της επικοινωνία όσων σχετίζονται με την αναφορά. Δηλαδή ο χρήστης που το δημιουργεί και διασώστες που θέλουν να προσφέρουν βοήθεια.

Ένα πλήρες παράδειγμα αναφοράς συμβάντος θα θέλαμε να είναι της εξής μορφής:



Σχήμα 4.3: Αναφορά περιστατικού από χρήστη

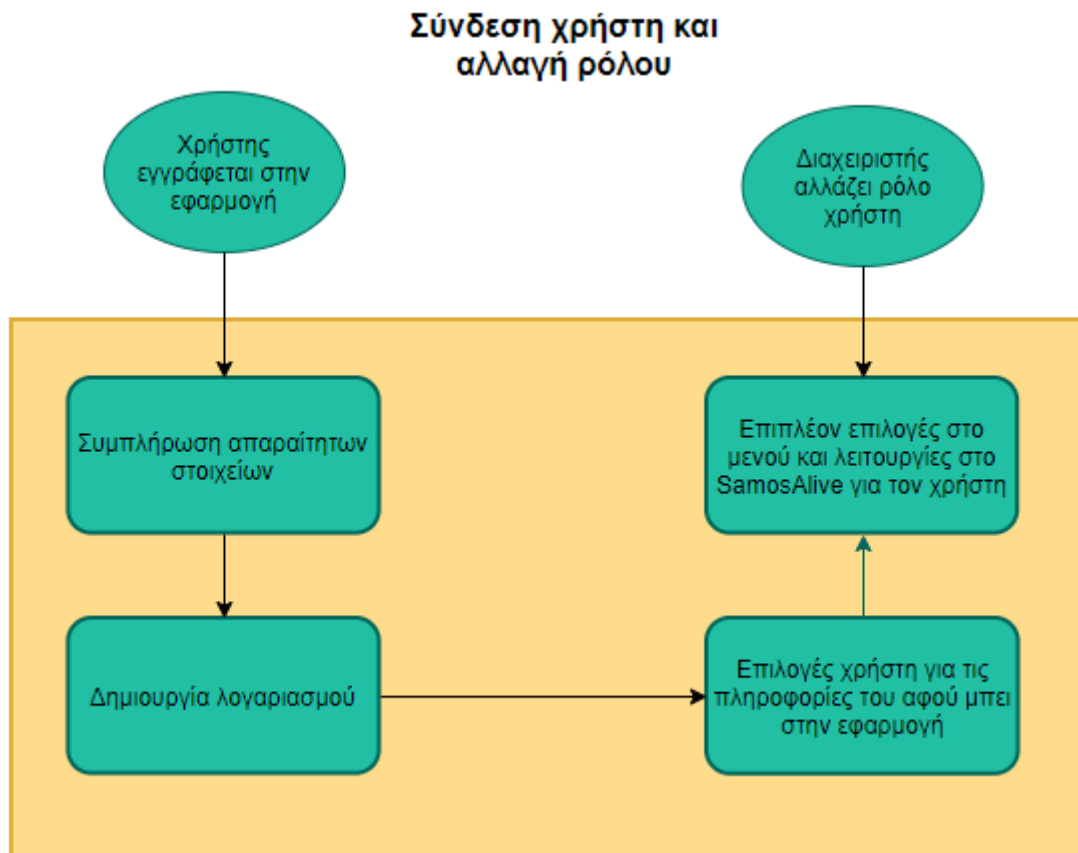
4.1.5 Πληροφορίες και προφίλ του χρήστη

Ο κάθε χρήστης θα παρέχει κάποιες βασικές πληροφορίες στην εφαρμογή για να αποφεύγεται η εντελώς ανώνυμη χρήση της εφαρμογής για ευνόητους λόγους που έχουν προαναφερθεί. Αυτές οι πληροφορίες για αρχή θα θέλαμε να ήταν ένα email, όνομα και κωδικό για να συνδέεται στην εφαρμογή καθώς και το όνομα και το επώνυμο του. Παράλληλα με αυτά όταν ο χρήστης ανοίγει πρώτη φορά την εφαρμογή θα πρέπει να αποδέχεται να έχει πρόσβαση στην τοποθεσία του καθώς βοηθάει σε χαρακτηριστικά γεωγραφικού περιεχομένου, για παράδειγμα σε κάποιο συμβάν θα κεντράρει στον χάρτη αμέσως την τοποθεσία του χρήστη αν έχει δώσει πρόσβαση στην τοποθεσία γλιτώνοντας έτσι πολύτιμο χρόνο. Μία ακόμα ιδέα ήταν να υπάρχει μια ομαδική συνομιλία για της επικοινωνία όσων σχετίζονται με την αναφορά. Δηλαδή ο χρήστης που το δημιουργεί και διασώστες που θέλουν να προσφέρουν βοήθεια.

Το προφίλ του κάθε χρήστη επίσης πρέπει να περιέχει κάποιες πληροφορίες ιδιωτικότητας όπως το τι θα εμφανίζεται δημόσια και τι όχι. Σαν επιλογή θα θέλαμε επίσης να προσθέσουμε να εμφανίζεται η διαθεσιμότητα του (συνδεδεμένος, απασχολημένος κλπ.) καθώς και τον ρόλο του.

Όσο αφορά τους βασικούς ρόλους των χρηστών θα υπάρχει ο απλός χρήστης, ο διασώστης και ο γενικός διαχειριστής. Ο κάθε ρόλος θα έχει επιπλέον λειτουργίες ανάλογα με την αρμοδιότητα του. Σαν εναλλακτικοί ρόλοι θα μπορούσαν να υπάρχουν οι τεχνικοί των απινιδωτών για να διορθώσουν κάποιο πιθανό πρόβλημα, διάφοροι οργανισμοί και ο τοπικός διαχειριστής ή αλλιώς διαχειριστής περιοχής.

Στο παρακάτω σχήμα αναπαριστάτε μια απλή ακολουθία βημάτων για την είσοδο του χρήστη στην εφαρμογή και την απόκτηση ιδιοτήτων/ρόλου από κάποιον διαχειριστή.



Σχήμα 4.4: Σύνδεση χρήστη και αλλαγή ρόλου

4.1.6 Ρόλοι του χρήστη στην εφαρμογή

Ιεραρχία κατηγοριών και ρόλων

Σε φθίνουσα σειρά η ιεραρχία κατηγοριών ρόλων είναι:

Κατηγορία Διαχειριστικών Ρόλων

1. General Admin (Γενικός Διαχειριστής)
2. Area Admin (Διαχειριστής Περιοχής)

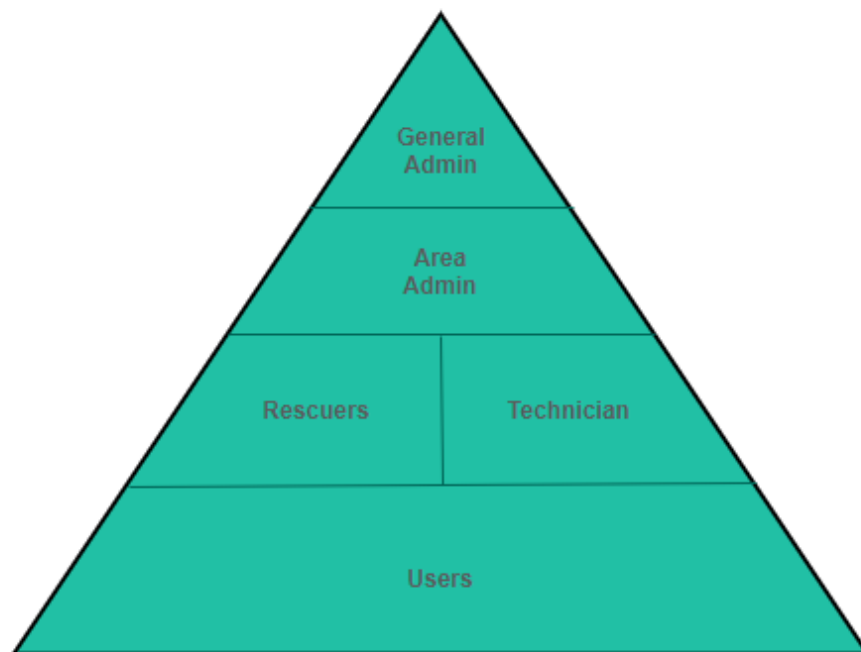
Κατηγορία Υπηρεσιακών Ρόλων

- Rescuer (Διασώστης)
- Technician (Τεχνικός)

Κατηγορία Απλών Ρόλων Χρηστών

- User (Απλός Χρήστης)

Στο σχήμα φαίνονται οι ρόλοι συνοπτικά καθώς και η ιεραρχία μεταξύ τους.



Σχήμα 4.5: Ιεραρχία ρόλων στην εφαρμογή SamosAlive

4.1.6.1 Διαχειριστικοί ρόλοι χρηστών

General Admin (Γενικός Διαχειριστής)

Επισκόπηση ρόλου:

Ο γενικός διαχειριστής της εφαρμογής SamosAlive, στέκεται υψηλότερα όλων των άλλων ρόλων στην ιεραρχία. Αναλαμβάνει την διαχείριση της εφαρμογής με πρόσβαση σε κάθε υπηρεσία, είναι επίσης υπεύθυνος για την συντήρηση και την εξουσιοδότηση ρόλων και καθηκόντων στα παρακάτω μέλη του συστήματος. Ο ρόλος αυτός για λόγους ασφαλείας δεν θα πρέπει να έχει συχνή χρήση στην εφαρμογή, παρά μόνο να συντηρεί και να εξουσιοδοτεί μέλη που βρίσκονται στην ακριβώς από κάτω βαθμίδα ιεραρχίας.

Απαιτήσεις πεδίου:

Για να γίνει κάποιος General Admin, θα πρέπει να γνωρίζει βασικούς τρόπους ασφαλείας, με το οποίο συνεπάγεται να διατηρεί αυστηρή ρουτίνα ασφάλειας στην συσκευή με την οποία συνδέεται, με σκοπό να αποφευχθεί οποιαδήποτε υποκλοπή στοιχείων σύνδεσης ή των token που λαμβάνει από την εφαρμογή.

Απαιτήσεις συμπεριφοράς:

Διατηρώντας τον ισχυρότερο ρόλο της εφαρμογής, τα άτομα που θα αναλάβουν τον ρόλο θα τους συντροφεύει η μεγαλύτερη ευθύνη. Βασιζόμενο στην προηγούμενη αναφορά, οι χρήστες που έχουν αυτό τον ρόλο πρέπει να είναι υπεύθυνα και επιφυλακτικά άτομα.

Απαιτήσεις εφαρμογής:

Υποχρέωση του γενικού διαχειριστή είναι να γνωρίζει κάθε λειτουργικότητα της εφαρμογής, των υπηρεσιών και των δικαιωμάτων που έχει κάθε ρόλος για την ολοκληρωμένη και σωστή μεταχείριση της εφαρμογής.

Λειτουργίες εφαρμογής: Έχει κάθε δυνατότητα και πρόσβαση σε όλες τις υπηρεσίες του back-end. Οι κύριες λειτουργίες του General Admin ωστόσο είναι να αναθέτει ρόλους, να κάνει εποπτεία των πόρων και ασφάλειας της εφαρμογής και αν χρειαστεί διαχείριση των χρηστών της.

Area Admin (Διαχειριστής Περιοχής)

Επισκόπηση ρόλου:

Ο διαχειριστής περιοχής βρίσκεται κάτω από τον γενικό διαχειριστή της εφαρμογής SamosAlive. Αναλαμβάνει την διαχείριση της εφαρμογής σε τοπικό επίπεδο περιοχής, δηλαδή είναι υπεύθυνος για τις υπηρεσιακές ομάδες και εξοπλισμούς που βρίσκονται στην περιοχή που του έχει ανατεθεί. Συντηρεί και την εξουσιοδοτεί ρόλους και

καθήκοντα στα παρακάτω μέλη του συστήματος με την προϋπόθεση ότι είναι στην περιοχή του.

Απαιτήσεις πεδίου:

Αντίστοιχα με τον γενικό διαχειριστή, οι βασικοί τρόποι ασφαλείας είναι σημαντικό στοιχείο που πρέπει να καταλαμβάνει ένα άτομο στο διαχειριστικό επιτελείο, που συνάδει με αυστηρότητα στην ρουτίνα ασφάλειας στην συσκευή με την οποία συνδέεται, με σκοπό να αποφευχθεί οποιαδήποτε υποκλοπή στοιχείων σύνδεσης ή των token που λαμβάνει από την εφαρμογή.

Απαιτήσεις συμπεριφοράς:

Αντίστοιχα με τον γενικό διαχειριστή, διατηρώντας ρόλο που ανήκει στο διαχειριστικό επιτελείο της εφαρμογής, τα άτομα με αυτή την ανάληψη ευθυνών είναι αναγκαία υποχρέωση να είναι υπεύθυνα και επιφυλακτικά άτομα.

Απαιτήσεις εφαρμογής:

Ο διαχειριστής περιοχής δεν είναι αναγκαίο να γνωρίζει όλες τις λειτουργικότητες της εφαρμογής αλλά κυρίως αυτές που απασχολούνται συχνά από τον ρόλο του. Καλή γνώση των δικαιωμάτων που έχει κάθε ρόλος για την ολοκληρωμένη και σωστή μεταχείριση της εφαρμογής.

Λειτουργίες εφαρμογής:

Οι κύριες λειτουργίες του διαχειριστή περιοχής είναι οι διαχείριση των ρόλων στους υπηρεσιακούς χρήστες της εφαρμογής αλλά και η εποπτεία γεγονότων και χρηστών για αξιολόγηση καταστάσεων και ενίσχυση των δυνατοτήτων της περιοχής με καλύτερη οργάνωση.

4.1.6.2 Υπηρεσιακοί ρόλοι χρηστών

Rescuer (Διασώστης)

Επισκόπηση ρόλου:

Ο διασώστης είναι ένα άτομο που σώζει ανθρώπινες ζωές από κίνδυνο, είναι υπεύθυνος για επίλυση περιστατικών υγείας στην εφαρμογή SamosAlive.

Απαιτήσεις πεδίου:

Τα άτομα που έχουν αναλάβει αυτό τον ρόλο οφείλουν να είναι εκπαιδευμένοι στην τεχνική διάσωσης με απινιδωτή, να διαθέτουν κάποιο πιστοποιητικό ή εμπειρία με προηγούμενη απασχόληση σε κάποιον φορέα πρώτων βοηθειών, στην περίπτωση που δεν προϋποθέτουν τίποτα από τα δύο, είναι αναγκαίο να προβλεφθεί από την εφαρμογή και από τους υπόλοιπους διασώστες ή υπεύθυνους διασωστών στην περιοχή να υπάρξει ένα πρόγραμμα εκπαίδευσης τους.

Απαιτήσεις συμπεριφοράς:

Τα άτομα που βρίσκονται σε αυτό τον ρόλο όσον αφορά την συμπεριφορά και στάση τους πρέπει να είναι συνεργάσιμα με συναδέλφους, χρήστες και υπόλοιπο προσωπικό, αφοσιωμένοι - πιστοί στον σκοπό τους και να λειτουργούν με μεγάλη υπευθυνότητα ως προς τα γεγονότα των χρηστών.

Απαιτήσεις εφαρμογής:

Να γίνει καλή μελέτη των απινιδωτών της περιοχής και όλων των ιδιαιτεροτήτων που μπορεί να έχουν και απαραίτητη η γνώση λειτουργίας της εφαρμογής SamosAlive για να μπορούν εύκολα να μεταχειρίζονται τις δυνατότητες που παρέχονται στον διασώστη.

Λειτουργίες εφαρμογής:

Διαθέτουν αποκλειστικό κανάλι και λειτουργικότητα στην εφαρμογή SamosAlive που ακούνε για γεγονότα που συμβαίνουν σε πραγματικό χρόνο. Σε περίπτωση γεγονότος, ο διασώστης δέχεται ειδοποίηση στις ώρες που είναι ενεργός για περιστατικό υγείας που είναι γύρω από μια ορισμένη ακτίνα που ακούει ο ίδιος, ορίζει έναν από τους προτεινόμενους απινιδωτές στην περιοχή και την βέλτιστη διαδρομή για τον επιλεγμένο απινιδωτή μέχρι το περιστατικό και αναλαμβάνει να εξυπηρετήσει το γεγονός. Μετά την λήξη του περιστατικού, είναι υποχρεωμένος να επιστρέψει τον απινιδωτή στον χώρο του ή να τον μεταφέρει σε άλλον διασώστη ή να το χρησιμοποιήσει άμεσα για άλλο περιστατικό, για κάθε σενάριο δανειζόμενος τον απινιδωτή είναι και παροδικά υπεύθυνος οπότε είναι υποχρεωμένος να ενημερώσει την εφαρμογή για την κατάσταση του. Όντας υπεύθυνος κατά την λειτουργία του, αν συμβεί κάτι στον απινιδωτή οφείλει να το αναφέρει στα τεχνικά προβλήματα με λεπτομερή ανάλυση αν γίνεται για το τι προέκυψε και να σταλεί στον χώρο του για επισκευή από τεχνικούς.

Technician (Τεχνικός)

Επισκόπηση ρόλου:

Ένας τεχνικός είναι εργαζόμενος σε έναν τομέα τεχνολογίας που είναι ικανός στη σχετική δεξιότητα και τεχνική, με σχετικά πρακτική κατανόηση των θεωρητικών αρχών.

Απαιτήσεις πεδίου:

Στην εφαρμογή SamosAlive ο ρόλος του τεχνικού, σε αυτή την φάση είναι υπεύθυνος για επίλυση τεχνικών προβλημάτων που παρουσιάζονται στις συσκευές απινιδωτών. Απαιτήση είναι ο απασχολούμενος σε αυτό τον ρόλο να έχει κάποιο πτυχίο μηχανικού/τεχνικού συσκευών ή να διαθέτει εμπειρία επισκευών πάνω σε απινιδωτές, επίσης θεωρείται ωφέλιμο να υπάρχουν άτομα που είναι καλοί αναλυτές συσκευών και

καταστάσεων, βρίσκουν εύκολα την αιτία και λύση και μπορούν να καθοδηγήσουν ανθρώπους για την επαναλειτουργία συσκευών χωρίς άμεση επαφή από τους ίδιους.

Απαιτήσεις συμπεριφοράς:

Όσα άτομα απασχολούνται με τον ρόλο αυτό, είναι σημαντικό να είναι υπομονετικοί ακροατές που ακούνε τα προβλήματα των χρηστών της εφαρμογής για τις συσκευές που προσπαθούν να αναλύσουν.

Απαιτήσεις εφαρμογής:

Να γίνει καλή μελέτη των προδιαγραφών των απινιδωτών της περιοχής και όλων των ιδιοτήτων που μπορεί να έχουν και απαραίτητη η γνώση λειτουργίας της εφαρμογής SamosAlive.

Λειτουργίες εφαρμογής:

Διαθέτουν κανάλι και λειτουργικότητα στην εφαρμογή χρήστη που ακούνε για προβλήματα που παρουσιάζονται σε μια περιοχή και επιλέγουν να αναλάβουν να επιδιορθώσουν το σφάλμα. Ένας τεχνικός μπορεί να αναλάβει να επιδιορθώσει άπειρες συσκευές ταυτόχρονα, ωστόσο αυτό θα προσαρμοστεί αναλόγως το δυναμικό των τεχνικών στην περιοχή για την καλύτερη κατανομή του φόρτου εργασίας. Οι τεχνικοί μπορούν επίσης να εργαστούν ως εκπρόσωποι για κατασκευαστές απινιδωτών, παρέχοντας συμβουλές σχετικά με την εγκατάσταση, τη λειτουργία και τη συντήρηση του σύνθετου εξοπλισμού και μπορεί να γράψουν προδιαγραφές και τεχνικά εγχειρίδια. Ευπρόσδεκτη είναι η διαθεσιμότητα τους να πληροφορήσουν χρήστες της εφαρμογής όταν παρουσιαστεί δυσκολία για το χειρισμό των απινιδωτών.

4.1.6.3 Απλοί χρήστες

User (Απλός Χρήστης)

Επισκόπηση ρόλου: Ο ρόλος των απλών χρηστών της εφαρμογής SamosAlive.

Απαιτήσεις συμπεριφοράς:

Να υπάρχει σεβασμός μεταξύ όλων των χρηστών. Οι χρήστες πρέπει να χρησιμοποιούν την εφαρμογή με κοινωνική ευθύνη, διότι η λανθασμένη χρήση της ή η κατάχρηση της διαταράσσει την ομαλότητα της εφαρμογής και το ανθρώπινο δυναμικό που δουλεύει γύρω από αυτή.

Λειτουργίες εφαρμογής:

Έχουν την δυνατότητα να αναφέρουν περιστατικά υγείας ή τεχνικά ζητήματα συσκευών και να παρακολουθούν την εξέλιξη τους, να αναζητήσουν οι ίδιοι απινιδωτές σε περιοχή, ή άλλους χρήστες αν τους ενδιαφέρει και να είναι σε ζωντανή επικοινωνία με τους εθελοντές-διασώστες και διαχειριστές της εφαρμογής για εξυπηρέτηση τους.

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου

Ετικέτα/tags	Τίτλος λειτουργικής απαίτησης	Περιγραφή λειτουργικής απαίτησης
Σύνδεση/Εγγραφή	Εγγραφή	Εγγραφή με νέα στοιχεία στην εφαρμογή
Σύνδεση/Εγγραφή	Σύνδεση	Σύνδεση με υπάρχον όνομα και κωδικό
Σύνδεση/Εγγραφή	Στοιχεία εγγραφής	Τα στοιχεία εγγραφής είναι: username, κωδικό, email, όνομα, επώνυμο, φωτογραφία (προαιρετικό)
Ρόλοι	Αρχικός ρόλος	Όλοι όσοι εγγράφονται έχουν τον ρόλο του «Χρήστη»
Ρόλοι	Αλλαγή ρόλου	Ο διαχειριστής μπορεί να αλλάξει, να προσθέσει ή να αφαιρέσει ρόλους σε κάθε χρήστη
Ρόλοι	Οι ρόλοι	Οι πιθανοί ρόλοι στην εφαρμογή είναι διαχειριστής, χρήστης, διασώστης, τεχνικός.
Απινιδωτές/Χάρτης	Προσθήκη απινιδωτή	Οι διαχειριστές προσθέτουν τους απινιδωτές με την διεύθυνση τους μέσω φόρμας
Απινιδωτές/Χάρτης	Αναζήτηση απινιδωτή	Αναζήτηση και εμφάνιση κοντινών απινιδωτών στον χάρτη
Απινιδωτές/Χάρτης	Αναζήτηση διαθέσιμου απινιδωτή	Αναζήτηση και εμφάνιση κοντινών διαθέσιμων απινιδωτών για αποφυγή χρόνου
Απινιδωτές/Χάρτης	Κατάσταση απινιδωτή	Ανανέωση κατάστασης απινιδωτών ανάλογα με τα προβλήματα και την χρήση τους
Απινιδωτές/Χάρτης	Οδηγίες προς απινιδωτή	Οδηγίες μετάβασης σε απινιδωτή (και σε συμβάν) πάνω στον χάρτη
Συμβάν	Κουμπί βοήθειας	Κουμπί για εμφάνιση κοντινών απινιδωτών
Συμβάν	Κουμπί βοήθειας	Κουμπί για αναφορά συμβάντος σε διασώστες

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου

Συμβάν	Φόρμα περιστατικού	Συμπλήρωση απαραίτητων πληροφοριών για το συμβάν όπως τύπος, σχόλιο κλπ.
Συμβάν	Σημείο περιστατικού	Αναφορά θέσης σε χάρτη για να ξέρουν οι διασώστες που να κατευθυνθούν
Συμβάν	Λεπτομέρειες για συμβάν	Ανακατεύθυνση σε νέα σελίδα με το συμβάν και «τοπική» συζήτηση για τους συμμετέχοντες
Συμβάν	Ειδοποίηση διασωστών	Ειδοποίηση διασωστών, όπου μέσα από Ζωντανή λίστα ή επιλεγμένη αναζήτηση βρίσκουν το συμβάν
Συμβάν	Κοντινοί απινιδωτές	Εμφάνιση κοντινών απινιδωτών σε διασώστες και οδηγίες μετάβασης προς τον επιλεγμένο απινιδωτή και έπειτα το συμβάν
Προφίλ	Ρυθμίσεις πληροφοριών	Ο χρήστης μπορεί να αλλάξει πληροφορίες στο προφίλ του και την εικόνα του (avatar)
Προφίλ	Ρυθμίσεις ιδιωτικότητας	Ο χρήστης μπορεί να αλλάξει διάφορες ρυθμίσεις ιδιωτικότητας
Ανακοινώσεις	Δημιουργία ανακοίνωσης	Δημιουργία ανακοίνωσης από επιλεγμένους χρήστες με επιλογή κατηγορίας που ανήκουν (tag)
Ανακοινώσεις	Αναζήτηση ανακοίνωσης	Αναζήτηση ανακοινώσεων από όλους μέσω tag
Προβλήματα	Αναφορά προβλήματος	Αναφορά προβλήματος ή έλλειψη απινιδωτή
Προβλήματα	Ενημέρωση χάρτη	Αλλαγή κατάστασης σε μη διαθέσιμο και εξαφάνιση από τον χάρτη προσωρινά
Προβλήματα	Αναζήτηση/επίλυση προβλήματος	Οι τεχνικοί βλέπουν τα προβλήματα και όταν τα διορθώνουν γίνεται επαναφορά διαθεσιμότητας
Σχόλια/Ιδέες	Σχόλια χρηστών	Φόρμα ανάρτησης κάποιας ιδέας ή σχολίων/κριτικών για την εφαρμογή

Ρυθμίσεις	Γενικές ρυθμίσεις	Μενού ρυθμίσεων για βασικές επιλογές που αφορούν την πλατφόρμα όπως γλώσσα και φόντο
-----------	-------------------	--

Πίνακας 4.2: Λειτουργικές απαιτήσεις

4.2 Μη λειτουργικές απαιτήσεις

4.2.1 Απόδοση

Για να έχουμε μια αποδοτική εφαρμογή θα πρέπει να λειτουργεί με “non-blocking” τρόπο ώστε να μπορούν να εξυπηρετηθούν πολλοί χρήστες ταυτόχρονα, με την καλύτερη διαχείριση των πόρων μας. Επίσης η χρήση του RSocket θα βοηθούσε στην αυξημένη ταχύτητα της μετάδοσης δεδομένων αν λάβουμε υπόψη ότι το πρωτόκολλο στέλνει δεδομένα με δυαδική μορφή. Θέλουμε η επικοινωνία μεταξύ client και server να τρέχει σε ένα μόνιμο ανοιχτό κανάλι τύπου WebSockets... . Τέλος μια in-memory database θα επιτάχυνε επιπλέον την ανάκληση των δεδομένων.

Η κατεύθυνση της εφαρμογής client θα είναι αυτή της “SPA” όπου ο χρήστης θα περιηγείται σε μία μόνο σελίδα στην οποία ανανεώνεται δυναμικά η διεπαφή της ανάλογα με την πλοήγηση που κάνει ο χρήστης. Αυτό εξοικονομεί πόρους που αφορούν το κατέβασμα νέων σελίδων.

4.2.2 Χρηστικότητα

Δεδομένου ότι η εφαρμογή προσπαθεί να επιλύσει θέματα υγείας και σοβαρά περιστατικά μέσα σε σύντομο χρονικό διάστημα, πρέπει το περιβάλλον της διεπαφής του χρήστη να είναι κατανοητό, με αυτοματοποιημένες διαδικασίες και χωρίς περιττές πληροφορίες. Ως προς την αντιμετώπιση συμβάντων μια φόρμα με τις απαραίτητες πληροφορίες αλλά παράλληλα με την μικρότερη παρέμβαση του χρήστη είναι ένας επιθυμητός συνδυασμός πάνω στην κρίσιμη ώρα του περιστατικού.

Άλλη μία απαίτηση της πλατφόρμας είναι η δυνατότητα επιλογής γλώσσας τουλάχιστον σε Ελληνικά και Αγγλικά. Η λειτουργία σε πολλαπλές γλώσσες είναι πάντα επιθυμητή αλλά μια αρχική προσέγγιση με τις δύο αυτές βασικές είναι αναγκαία.

4.2.3 Ασφάλεια

Η εφαρμογή θα πρέπει να λειτουργεί με tokens. Το access token είναι αυτό που ο χρήστης χρειάζεται να έχει για να επαληθευτεί και να μπορεί να τρέξει τις δυνατότητες της πλατφόρμας. Αντίστοιχα το refresh token είναι αυτό που ανανεώνει το access token του χρήστη. Το τελευταίο είναι απαραίτητο για την αναγνώριση του ρόλου του χρήστη, το όνομα του και διάφορες άλλες πληροφορίες γύρω από αυτόν.

Οι υπηρεσίες θα είναι διαθέσιμες προς όλους τους χρήστες ή σε επιλεγμένους ρόλους. Αυτές που ελέγχουν καταστάσεις προφίλ ενός χρήστη θα περιορίζονται σε μικρή ομάδα διαχειριστικών ρόλων. Γενικά στην εφαρμογή και μεταξύ των διαχειριστικών ρόλων υπάρχει μια ιεραρχία δεντρικής μορφής και αλλαγές κατάστασης χρηστών που αφορούν ρόλους σε ανώτερη βαθμίδα στην ιεραρχία δεν επιτρέπονται.

Όλα τα δεδομένα τα οποία εισάγονται από την μεριά του χρήστη πρέπει να επαληθεύονται ως προς την ορθότητα τους ώστε να αποφευχθούν τυχόν ανεπιθύμητες ενέργειες και να προστατεύεται η εφαρμογή και η βάση δεδομένων. Ένα ακόμα ζήτημα όταν φτάσει η ώρα της χρήσης της πλατφόρμας είναι η έκδοση κάποιων certificates για παροχή ασφάλειας τα οποία θα εκδίδονται από μια “CA” και θα επαληθεύονται από τους browser.

4.2.4 Ιδιωτικότητα

Τα εγγεγραμμένοι μέλη της εφαρμογής θα έχουν έναν πίνακα με επιλογές ιδιωτικότητας όπου από εκεί θα ρυθμίζουν την ορατότητα των πληροφοριών τους.

4.2.5 Ευελιξία - Δυνατότητα Συντήρησης

Θεωρούμε ότι είναι αναγκαίος ένας ενιαίος και συμβατικός τρόπος ανάπτυξης κώδικα μέσα στην ομάδα. Η απομόνωση λειτουργικού κώδικα σε μικρότερα κομμάτια θα μπορέσει να συμβάλει στην ευκολότερη συντήρηση της εφαρμογής. Με έναν τέτοιο διαχωρισμό μορφής micro services η εφαρμογή μπορεί να πάρει διαστάσεις μιας cloud υποδομής. Παράλληλα θέλουμε έναν κεντρικό έλεγχο για τον συντονισμό των απαιτήσεων και των ρυθμίσεων περιβάλλοντος των micro services σε περιπτώσεις αλλαγής.

4.2.6 Επαληθευσσιμότητα

Η επαλήθευση της σωστής λειτουργίας των συστημάτων του server θα αξιοποιείται μέσω unit-testing χρησιμοποιώντας τεχνικές προσομοίωσης και άλλων τυπικών μεθόδων.

4.2.7 Δυνατότητα επαναχρησιμοποίησης

Η χρήση της πλατφόρμας Kubernetes δίνει την δυνατότητα επαναχρησιμοποίησης ενός λειτουργικού module. Λόγω υποδομής cloud της εφαρμογής όπως αναφέρθηκε παραπάνω τα διάφορα λειτουργικά modules θα γίνονται instances ως Kubernetes containers. Με αυτό τον τρόπο θα αποφευχθεί το πρόβλημα του single point of failure εφόσον τα λειτουργικά instances έχουν πολλαπλασιαστεί, άρα σε περίπτωση βλάβης κάποιου τα υπόλοιπα θα συνεχίσουν την λειτουργία τους.

4.2.8 Φορητότητα

Η εφαρμογή προορίζεται να τρέξει κυρίως σε κινητές συσκευές όμως υπάρχει περίπτωση χρήσης της και σε browser. Άρα η σύνταξη του κώδικα πρέπει να βασίζεται σε web τεχνολογίες ώστε να μπορούν να τρέξουν σε διάφορα περιβάλλοντα. Η χρήση του “PWA” είναι μια επιθυμητή επιλογή και προσφέρει τεράστιες δυνατότητες στην φορητότητα της εφαρμογής αφού αυτός ο κώδικας που θα τρέχει σε browser, χωρίς μεγάλες αλλαγές θα μπορεί εύκολα να τρέξει σε desktop και σε κινητό.

Ετικέτα/tags	Τίτλος μη λειτουργικής απαίτησης	Περιγραφή μη λειτουργικής απαίτησης
Απόδοση	Non-block για πολλαπλούς χρήστες	Non-blocking back-end για πολλαπλή εξυπηρέτηση χρηστών ταυτόχρονα, με την καλύτερη διαχείριση των πόρων
Απόδοση	Rsocket	Χρήση του RSocket πρωτοκόλλου για επικοινωνία μεταξύ client-server λόγω τεχνικών δυνατοτήτων
Απόδοση	In-memory βάση δεδομένων	Χρήση In-Memory database για ταχεία ανάκληση δεδομένων
Απόδοση	Single Page Application	Κατεύθυνση front-end εφαρμογής ως SPA αντί για MPA
Απόδοση	Αξιολόγηση από Lighthouse	Ικανοποιητική απόδοση εφαρμογής χρήστη μέσα από την αξιολόγηση του Lighthouse
Χρηστικότητα	Λιτό και αυτοματοποιημένο	Λιτό, εύκολο και με αυτοματοποιημένες λειτουργικότητες περιβάλλον εφαρμογής χρήστη
Χρηστικότητα	Ελάχιστη δυνατή επέμβαση χρήστη	Αίτηση συμπλήρωσης για συμβάν με την λιγότερη αναγκαία παρέμβαση από τον χρήστη
Χρηστικότητα	Πολλαπλές γλώσσες	Χρήση πολλαπλών γλωσσών(Ελληνικά, Αγγλικά)
Ασφάλεια	Επαλήθευση χρηστών	Λειτουργία με JWT για επαλήθευση χρηστών και

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου

		εξουσιοδότηση σε υπηρεσίες
Ασφάλεια	Access Token	Access Tokens για πρόσβαση σε υπηρεσίες (μικρή διάρκεια ζωής)
Ασφάλεια	Refresh Token	Refresh Tokens για ανανέωση του access token (μεγάλη διάρκεια ζωής)
Ασφάλεια	Υπηρεσίες ανά ρόλο	Υπηρεσίες θα είναι διαθέσιμες προς όλους τους χρήστες ή σε ανατεθειμένη ομάδα ρόλων
Ασφάλεια	Υπηρεσίες για διαχειριστές	Υπηρεσίες ελέγχου χρηστών από ομάδα διαχειριστικών ρόλων τηρώντας τις ιεραρχικές συνθήκες
Ασφάλεια	Επαλήθευση ορθότητας	Οι υπηρεσίες επαληθεύουν τα δεδομένα αίτησης του χρήστη ως προς την ορθότητα τους
Ασφάλεια	Απόκρυψη κρυφών υπηρεσιών	Απομόνωση των κρυφών υπηρεσιών σε εσωτερικό δίκτυο όπως βάσεις δεδομένων
Ασφάλεια	Certificate από CA	Έκδοση certificate από επαληθευμένο CA
Ιδιωτικότητα	Ρυθμίσεις ιδιωτικότητας	Επιλογή ρυθμίσεων ιδιωτικότητας και ορατότητας πληροφοριών από τους χρήστες
Συντήρηση	Cloud υποδομή	Back-end εφαρμογή ως μικρό υπηρεσίες – Cloud υποδομή
Συντήρηση	Κεντρικός έλεγχος	Κεντρικός έλεγχος των ρυθμίσεων περιβάλλοντος από ξεχωριστό server
Συντήρηση	Συμβατικός τρόπος κώδικα	Δομημένος συμβατικός τρόπος ανάπτυξης του κώδικα μέσα στην ομάδα
Επαληθευσιμότητα	Unit testing	Unit testing πάνω στις υπηρεσίες ή σε απομονωμένη μονάδα λειτουργικού κώδικα που ανήκει σε υπηρεσία
Επαναχρησιμοποίηση	Χρήση Kubernetes containers	Kubernetes πλατφόρμα για εγκατάσταση των

		microservices ως containers στην VM
Επαναχρησιμοποίηση	Instances σε container	Μέσω του Kubernetes δυνατότητα πολλαπλασιασμού των instances ενός container
Φορητότητα	Διαθεσιμότητα εφαρμογής	Διαθεσιμότητα της εφαρμογής σε mobile, desktop
Φορητότητα	Εγκατάσταση μέσω browser	Χρήση του PWA για δυνατότητα εγκατάστασης της εφαρμογής από τον browser στην πλατφόρμα του λειτουργικού συστήματος

Πίνακας 4.3: Μη λειτουργικές απαιτήσεις

4.2.9 Σημαντικότητα σωστής ανάλυσης απαιτήσεων

Όλες οι απαιτήσεις λειτουργικές και μη είναι σημαντικές για την ετοιμότητα μιας παραγωγικής έκδοσης της εφαρμογής SamosAlive. Το πλάνο πριν την υλοποίηση κώδικα σχεδιάστηκε ενσωματωμένο με απαιτήσεις που αναλύθηκαν για την σωστή λειτουργική ροή της εφαρμογής στο περιεχόμενο διαχείρισης εκτατών αναγκών υγείας, με ευκολία εποπτείας πόρων, συντήρησης αλλά και επεκτασιμότητας. Σε ένα τόσο απαιτητικό περιβάλλον αλλά και μεταβαλλόμενο λόγω της εξέλιξης των τεχνολογικών δυνατοτήτων τροποποιήθηκαν και προστέθηκαν και θα συνεχίζονται να προστίθενται απαιτήσεις που έχουν σκόπιμη χρήση την ενίσχυση της εφαρμογής τόσο λειτουργικά όσο και αρχιτεκτονικά.

5. Αρχιτεκτονική συστήματος και τεχνολογίες προγραμματισμού

5.1 Back-end

Το back-end κομμάτι του κώδικα είναι υλοποιημένο στην Java γλώσσα, έκδοση 15. Εκτός από τον απλό κώδικα σε «Java» τελικός στόχος ήταν να λειτουργεί η πλατφόρμα με αντιδραστικό («reactive») τρόπο που προσφέρει ασύγχρονη non-blocking παραλληλία με μικρό αριθμό νημάτων, αποτελεσματικότητα και επέκταση με λιγότερους πόρους.

Οι βασικές Spring βιβλιοθήκες οι οποίες χρησιμοποιήθηκαν για την υλοποίηση του back-end.

Spring-Webflux

Web Framework με reactive υποδομή χάρης το Reactor Netty. Το Webflux είναι web server που χρησιμοποιείται εκτενώς από όλα τα module του back-end συστήματος. Προσφέρει διάφορα functional endpoints, ένα event loop μοντέλο παραλληλίας με πολύ λίγα threads και σε συνδυασμό με την βιβλιοθήκη Netty παρέχει Mono και Flux τύπους δεδομένων που «τυλίζουν» ένα υπάρχον αντικείμενο για την ροή του μέσα στο publish-subscribe event loop.

Spring-RSocket

Το RSocket είναι ένα νέο πρωτόκολλο ανταλλαγής μηνυμάτων που έχει σχεδιαστεί για την επίλυση ορισμένων συνηθισμένων προκλήσεων επικοινωνίας για microservices. Το RSocket είναι ευέλικτο πρωτόκολλο που λειτουργεί μέσω TCP ή WebSockets. Αυτό σημαίνει ότι η επικοινωνία γίνεται με δυαδικά μηνύματα χωρίς μετατροπή. Με σύγχρονα στοιχεία ελέγχου, όπως **multiplexing**, **back-pressure**, **resumption**, **routing** και διαθέτει πολλαπλές λειτουργίες ανταλλαγής μηνυμάτων, όπως **fire-and-forget**, **request-response**, και **streaming**. Το RSocket είναι επίσης πλήρως αντιδραστικό, επομένως είναι ιδανικό για τις εφαρμογές microservices υψηλής απόδοσης.

Spring-Security

Ισχυρό και εξαιρετικά προσαρμόσιμο πλαίσιο ελέγχου ταυτότητας και πρόσβασης. Είναι το de-facto πρότυπο για την εξασφάλιση ασφάλειας εφαρμογών της Java Spring, μπορεί να επεκταθεί για να ικανοποιήσει τις προσαρμοσμένες απαιτήσεις.

Spring Cloud Gateway

Το Spring Cloud Gateway ⁴ στοχεύει να παρέχει έναν απλό, αλλά αποτελεσματικό τρόπο δρομολόγησης στα microservices του SamosAlive και διασφαλίζει λειτουργίες όπως: ασφάλεια, παρακολούθηση / μετρήσεις και ανθεκτικότητα.

Χαρακτηριστικά του Spring Cloud Gateway:

- Χτισμένο στο Spring Framework 5, Project Reactor και Spring Boot 2.0
- Δυνατότητα αντιστοίχισης διαδρομών σε οποιοδήποτε χαρακτηριστικό αίτησης.
- Τα Predicates και τα φίλτρα είναι ειδικά για τις διαδρομές προς τα microservices.
- Ενσωμάτωση Circuit Breaker.
- Ενσωμάτωση Spring Cloud Discovery Client
- Εύκολο να γραφτούν Predicates και φίλτρα
- Περιορισμός ποσοστού αιτήσεων
- Path Rewriting

Redis

Το Redis ³ χρησιμοποιείται σε μεγάλο βαθμό από το back-end, λόγω της ταχύτητας που προσφέρει στη ανάκληση των δεδομένων.

Overview:

Το Redis είναι ένας ανοιχτός κώδικας, ένας χώρος αποθήκευσης δεδομένων στη μνήμη, που χρησιμοποιείται ως βάση δεδομένων, cache και message broker. Το Redis παρέχει δομές δεδομένων, όπως συμβολοσειρές, κατακερματισμούς, λίστες, σύνολα, ταξινομημένα σύνολα με ερωτήματα εύρους, bitmap, hyperloglogs, γεωχωρικούς δείκτες και ροές. Το Redis διαθέτει built-in replication και παρέχει υψηλή διαθεσιμότητα μέσω του Redis Sentinel και αυτόματο partitioning με το Redis Cluster.

Για να επιτύχει κορυφαία απόδοση, το Redis λειτουργεί με ένα σύνολο δεδομένων στη μνήμη.

Το Redis υποστηρίζει επίσης ασύγχρονο replication, με πολύ γρήγορο non-blocking first synchronization, αυτόματη επανασύνδεση με μερικό επανασυγχρονισμό.

Άλλα χαρακτηριστικά που διαθέτει το redis είναι:

- [Transactions](#)
- [Pub/Sub](#)
- [Lua scripting](#)
- [Keys with a limited time-to-live](#)
- [LRU eviction of keys](#)
- [Automatic failover](#)

Χρησιμοποιείται από το SamosAlive για:

- Να διατηρεί με key=value δομή, τα id των Access Token των χρηστών που είναι συνδεδεμένοι στο σύστημα, για την επαλήθευση των χρηστών στις back-end υπηρεσίες.
- Λειτουργεί ως MessageBroker, για να δημιουργεί κανάλια publish/subscribe. Οι χρήστες κάνουν subscribe μέσω ενός microservice και δέχονται τα publish μηνύματα από microservices που σχετίζονται με το channel αυτό.

Express ¹

Το Express είναι ένα minimal και ευέλικτο web application framework για javascript που τρέχει σε nodejs περιβάλλον. Παρέχει ένα ισχυρό σύνολο δυνατοτήτων για εφαρμογές Ιστού και για κινητές συσκευές. Χρησιμοποιείται σε διάφορα microservices (όπως edge-service, cdn-service) του συστήματος λόγω της ελάχιστης κατανάλωσης μνήμης RAM αλλά και τις μεγάλες δυνατότητες που παρέχει σε ασύγχρονες IO διαδικασίες για μεταφορά αρχείων.

Οι υπηρεσίες που χρησιμοποιούνται:

Auth-Service

Το microservice που είναι υπεύθυνο για την ασφάλεια όλης της εφαρμογής. Ανεπτυγμένο σε Java με τα SpringBoot dependencies για reactive λειτουργία και ασφάλεια. Μεταξύ άλλων σημαντικών λειτουργιών ασφάλειας δημιουργεί JWT (Json Web Tokens), Access Tokens και Refresh Tokens για τους χρήστες που συνδέονται στην εφαρμογή. Τα tokens αυτά είναι σημαντικά για την αυθεντικοποίηση και επαλήθευση των χρηστών.

Συνοπτικά περιέχει τις εξής λειτουργίες:

- Αυθεντικοποίηση χρήστη μέσω του login.
- Εγγραφή λογαριασμού ενός νέου χρήστη
- Αλλαγή κωδικού χρήστη
- Αλλαγή ρόλου χρήστη (Μόνο Admin)
- Πρόσθεση ρόλου χρήστη (Μόνο Admin)
- Διαγραφή ρόλου χρήστη (Μόνο Admin)
- Ban χρηστών (Μόνο Admin)
- Unban χρηστών (Μόνο Admin)
- Διαγραφή όλων των tokens ενός χρήστη
- Αποσύνδεση χρήστη
- Ανανέωση του access token
- Επιστροφή πληροφοριών χρήστη με νέο access token

Aed-Service

Το microservice που είναι υπεύθυνο για τις λειτουργίες των απινιδωτών, γεγονότων-συμβάντων και προβλημάτων απινιδωτών.

Οι λειτουργίες που παρέχονται είναι οι εξής:

1. AedDevices

- Καταχώρηση απινιδωτή με πληροφορίες
- Αναζήτηση απινιδωτή με βάση το Id του
- Αναζήτηση απινιδωτών με βάση το nickname, επιστροφή προεπισκόπησης του προφίλ των απινιδωτών που περιέχουν το αλφαριθμητικό
- Επεξεργασία πληροφοριών απινιδωτή
- Αναζήτηση όλων των απινιδωτών σε επιλεγμένη κοντινή ακτίνα
- Αναζήτηση διαθέσιμων απινιδωτών σε επιλεγμένη κοντινή ακτίνα
- Αναζήτηση διαθέσιμων απινιδωτών σε επιλεγμένη κοντινή ακτίνα με πληροφορίες καθοδήγησης

2. AedEvent

- Δημιουργία συμβάντος (Event)
- Αναζήτηση συμβάντων με κριτήρια τύπου και κατάστασης
- Αναζήτηση των εκκρεμών συμβάντων
- Αναζήτηση συμβάντος με βάση το Id
- Ανάκτηση πληροφοριών διασώστη και συσκευής απινιδωτή ενός γεγονότος
- Προεπισκόπηση πληροφοριών χρηστών που συμμετέχουν σε γεγονός
- Subscribe συνδεδεμένων διασωστών για να “ακούνε” εκκρεμή events
- Subscribe σε αλλαγές κατάστασης συγκεκριμένου event
- Subscribe σε αλλαγές της ορισμένης συσκευής απινιδωτή ενός συγκεκριμένου event
- Subscribe σε αλλαγές του ορισμένου διασώστη ενός συγκεκριμένου event
- Subscribe σε αλλαγές των συμμετεχόντων χρηστών συγκεκριμένου event
- Subscribe σε νέα σχόλια συγκεκριμένου event
- Ανάκτηση των σχολίων που ανήκουν σε σελίδα σε συγκεκριμένο event
- Ανάρτηση σχολίου από συμμετέχων σε συγκεκριμένο event
- Ανάθεση συμβάντος σε διασώστη
- Ολοκλήρωση συμβάντος από τον διασώστη

3. AedProblems

- Δημιουργία προβλήματος απινιδωτή
- Αναζήτηση των προβληματικών απινιδωτών
- Αναζήτηση πληροφοριών προβλήματος με βάση το Id
- Subscribe συνδεδεμένων τεχνικών για να “ακούνε” για προβληματικούς απινιδωτές
- Ανάθεση επιδιόρθωσης προβληματικού απινιδωτή από τεχνικό
- Ολοκλήρωση επιδιόρθωσης προβληματικού απινιδωτή από τεχνικό

Config-Server

Το microservice υπεύθυνο για συγκεντρωτισμό και διανομή των config properties των microservices που είναι αποθηκευμένα σε GitHub repository. Τα microservices κατά την εκκίνηση τους στέλνουν request στο config-server και δέχονται response βασισμένα στο spring.application.name, το αντίστοιχο config property.

Gateway-Service

Το microservice που λειτουργεί ως reverse proxy μεταξύ της front-end εφαρμογής και των microservices.

Cdn-Service

Το microservice υπεύθυνο για ανέβασμα αρχείων φωτογραφιών σε εξωτερικό cdn server (cloudinary). Έχει υλοποιηθεί με javascript που τρέχει πάνω σε nodejs περιβάλλον και χρησιμοποιεί για web framework το express και express-fileupload για το ανέβασμα αρχείων.

Edge-Service

Το microservice υπεύθυνο για το κατέβασμα της εφαρμογής front-end (html, js, css). Ο χρήστης που θα επισκεφτεί το domain του SamosAlive θα λάβει τα απαραίτητα αρχεία για την εμφάνιση της αρχικής σελίδας.

News-Service

Το microservice υπεύθυνο για το συντονισμό των ανακοινώσεων, δηλαδή της δημιουργίας τους και την αναζήτηση τους είτε συνολικά είτε με κριτήρια.

Οι λειτουργίες που περιέχει είναι οι παρακάτω:

- Ανάρτηση ανακοίνωσης με επιλογή κατηγορίας που ανήκει (επιτρέπεται μόνο με κατοχή συγκεκριμένων ρόλων)
- Αναζήτηση ανακοινώσεων με βάση την κατηγορία τους
- Ανάκτηση πληροφοριών συγκεκριμένης ανακοίνωσης με βάση το Id
- Αναζήτηση και προβολή όλων των ανακοινώσεων από όλες τις κατηγορίες

User-Service

Το microservice που είναι υπεύθυνο για τις λειτουργίες χρηστών.

Συνοπτικά περιέχει τις εξής λειτουργίες:

- Αναζήτηση χρήστη με βάση το Id του, φιλτράρισμα των πληροφοριών που δίνει με βάση τις ρυθμίσεις ιδιωτικότητας που έχει για τον λογαριασμό του.
- Αναζήτηση χρηστών με βάση το username, επιστροφή προεπισκόπησης του προφίλ των χρηστών που περιέχουν το αλφαριθμητικό.
- Αναζήτηση χρηστών με βάση το username, επιστροφή των usernames που περιέχουν το αλφαριθμητικό αυτό.
- Επεξεργασία και ορισμός των ρυθμίσεων ιδιωτικότητας του προφίλ του χρήστη.
- Επεξεργασία και ορισμός των προσωπικών πληροφοριών του προφίλ του χρήστη.
- Επιστροφή των ρυθμίσεων ιδιωτικότητας στον ίδιο τον χρήστη ή διαχειριστή.
- Αλλαγή κατάστασης του χρήστη στην εφαρμογή

5.2 Front-end

Για το μέρος της εργασίας που φαίνεται στον χρήστη, δηλαδή το front-end έχουμε κάνει μια ανάμειξη γλωσσών και τεχνολογιών. Έχουμε προσαρμόσει διάφορα συστατικά του «Vuetify» σε έναν κώδικα τύπου Vue. Βέβαια δεν θα μπορούσαμε να εμφανίζουμε τα χαρακτηριστικά στον χρήστη χωρίς να έχουμε τρόπο να τα παίρνουμε από τον server. Γίνεται επίσης χρήση της TypeScript για να συνεργάζονται καλύτερα οι προγραμματιστές μεταξύ τους και να είναι εύκολα διαχειρίσιμο το έργο. Η εφαρμογή θεωρήσαμε ότι είναι καλό να είναι «single-page application». Αυτό σημαίνει με απλά λόγια ότι κάθε εναλλαγή μεταξύ των σελίδων είναι μεταφορική καθώς έχουμε μια ενιαία σελίδα στην οποία απλώς ξαναγράφονται και ανανεώνονται τα δεδομένα της μέσω δυναμικής επεξεργασίας της JavaScript. Ως πρόγραμμα για γράψιμο του κώδικα προτιμήσαμε πάλι την εταιρία «JetBrains» με την χρήση αυτή την φορά του «WebStorm».

5.2.1 Απαραίτητα plugins για την λειτουργικότητα της εφαρμογής

Vue I18n:

Για την εναλλαγή μεταξύ των γλωσσών αποφασίσαμε να χρησιμοποιήσουμε την επέκταση i18n του Vue, η οποία προσφέρει ένα εύχρηστο format όπου μέσα από μία key-value δομή επιλέγεις τα επιθυμητά κομμάτια μετάφρασης.

Vuex ⁶:

Είναι διαχειριστής κατάστασης το οποίο αποθηκεύει κρίσιμα δεδομένα της εφαρμογής του client (front-end). Λειτουργεί ως μια κεντρική διαχείριση για όλα τα components και τις σελίδες μιας εφαρμογής με κανόνες που διασφαλίζουν ότι η κατάσταση μπορεί να μεταλλαχθεί μόνο με προβλέψιμο τρόπο. Στα modules του Vuex λαμβάνει χώρα η επικοινωνία μεταξύ client-server.

Vue Router ⁵:

Αυτή η επέκταση ενσωματώνεται βαθιά στον πυρήνα του Vue και είναι το αναπόσπαστο κομμάτι που κάνει την εφαρμογή SPA. Εφόσον οριστούν τα routes για τις διάφορες σελίδες της εφαρμογής το Vue Router μας βοηθάει να διαχειριζόμαστε και να πλοηγούμαστε σε αυτές.

Περιέχει και τα εξής χαρακτηριστικά:

- Nested route/view mapping
- Modular, component-based router configuration
- Route params, query, wildcards
- View transition effects powered by Vue.js' transition system
- Fine-grained navigation control
- Links with automatic active CSS classes
- HTML5 history mode or hash mode, with auto-fallback in IE9
- Customizable Scroll Behavior

Vuetify:

Το Vuetify είναι ένα πλήρες UI framework που δημιουργήθηκε πάνω από το Vue.js. Ο στόχος του έργου είναι να παρέχει στους προγραμματιστές τα εργαλεία που χρειάζονται για να δημιουργήσουν πλούσιες και ενδιαφέρουσες εμπειρίες χρηστών. Σε αντίθεση με άλλα frameworks, το Vuetify έχει σχεδιαστεί από τη βάση προς τα πάνω ώστε να είναι εύκολο να μάθει και να ανταμείβει την εξάσκηση με εκατοντάδες προσεκτικά κατασκευασμένα εξαρτήματα από τις προδιαγραφές Material Design . Το Vuetify ακολουθεί μια πρώτη προσέγγιση για σχεδίαση για κινητά, πράγμα που σημαίνει ότι η εφαρμογή front-end λειτουργεί ακριβώς out-of-box, είτε πρόκειται για κινητό, tablet ή desktop υπολογιστή.

Axios:

Το Axios είναι ένας client HTTP που βασίζεται στα Promises της JavaScript που μπορεί να χρησιμοποιηθεί στην εφαρμογή front-end του SamosAlive και για να επικοινωνήσει με το back-end. Χρησιμοποιώντας το Axios είναι εύκολο να σταλεί

ασύγχρονο αίτημα HTTP σε REST τελικά σημεία και εκτέλεση λειτουργιών CRUD. Η βιβλιοθήκη Axios χρησιμοποιείται στην frontend όπου υπάρχει Http Request-Response επικοινωνία και δεν χρησιμοποιείται η RSocket επικοινωνία.

Leaflet:

Το Leaflet είναι κορυφαία βιβλιοθήκη JavaScript ανοιχτού κώδικα για διαδραστικούς χάρτες φιλικούς προς κινητά. Ζυγίζοντας περίπου 39 KB JS, έχει όλες τις δυνατότητες χαρτογράφησης που χρειάζονται οι περισσότεροι προγραμματιστές. Το Leaflet έχει σχεδιαστεί με γνώμονα την απλότητα, την απόδοση και τη χρηστικότητα. Λειτουργεί αποτελεσματικά σε όλες τις μεγάλες πλατφόρμες desktop και κινητών συσκευών, μπορεί να επεκταθεί με πολλές προσθήκες, έχει ένα όμορφο, εύχρηστο και καλά τεκμηριωμένο API.

5.2.2 Μεταφορά και εμφάνιση σε κινητή συσκευή – PWA

Μια προοδευτική διαδικτυακή εφαρμογή (PWA) είναι ένας τύπος λογισμικού εφαρμογών που παρέχεται μέσω του διαδικτύου, δημιουργήθηκε χρησιμοποιώντας κοινές τεχνολογίες ιστού, συμπεριλαμβανομένων των HTML, CSS και JavaScript. Προορίζεται να λειτουργεί σε οποιαδήποτε πλατφόρμα που χρησιμοποιεί πρόγραμμα περιήγησης συμβατό με πρότυπα, συμπεριλαμβανομένων desktop και κινητών συσκευών. Τα χαρακτηριστικά PWA περιορίζουν το χάσμα μεταξύ της εμπειρίας χρήστη σε διαδικτυακές και εγγενείς εφαρμογές.

Δεδομένου ότι μια προοδευτική εφαρμογή ιστού είναι ένας τύπος ιστοσελίδας ή ιστότοπου γνωστός ως εφαρμογή ιστού, δεν απαιτούν ξεχωριστή ομαδοποίηση ή διανομή. Συγκεκριμένα, δεν απαιτείται απαίτηση για προγραμματιστές ή χρήστες να εγκαταστήσουν τις εφαρμογές ιστού μέσω ψηφιακών συστημάτων διανομής όπως το Apple, App Store ή το Google Play. Σε διάφορους βαθμούς, τα μεγάλα καταστήματα εφαρμογών άρχισαν να αγκαλιάζουν PWA, επιτρέποντάς τους να βρίσκονται σε καταστήματα εφαρμογών. Από το 2021, οι λειτουργίες που σχετίζονται με το PWA υποστηρίζονται σε διάφορους βαθμούς από τα προγράμματα περιήγησης Google Chrome, Apple, Safari, Firefox για Android και Microsoft Edge. Αρκετές επιχειρήσεις επισημαίνουν σημαντικές βελτιώσεις σε μια μεγάλη ποικιλία βασικών δεικτών απόδοσης μετά την εφαρμογή PWA, όπως αυξημένος χρόνος που αφιερώνεται στη σελίδα, μετατροπές ή έσοδα.

Με τις δυνατότητες του PWA και του service-worker που την συντροφεύει (για offline, cache κτλ.) η front-end εφαρμογή του SamosAlive είναι εύκολα επεκτάσιμη και διαχειρίσιμη από τους προγραμματιστές της με έναν codebase για κάθε πλατφόρμα.

5.3 Βάση αποθήκευσης δεδομένων

Όσο αφορά την αποθήκευση των δεδομένων είχαμε αρκετές επιλογές όμως προτιμήσαμε μια «NoSql» προσέγγιση και συγκεκριμένα κάναμε χρήση του **MongoDB**². Τα δεδομένα της εφαρμογής αποθηκεύονται ως “έγγραφα” με όλες τις πληροφορίες να είναι άμεσα προσβάσιμες για ανάγνωση ή εγγραφή. Τα δεδομένα θα πρέπει να αλλάζουν συνεχώς σε πραγματικό χρόνο, να διατίθεται geospatial λειτουργίες και η βάση δεδομένων πρέπει να κάνει εύκολα scaling, η MongoDB κάλυπτε τα παραπάνω βασικά κριτήρια που χρειάζεται η εφαρμογή SamosAlive.

Η MongoDB πέραν της αναφοράς των βασικών κριτηρίων, έχει αρχιτεκτονικά στην υλοποίηση της λειτουργικότητας και χαρακτηριστικά που εξασφαλίζουν για πολύ εύκολη συντήρηση και scaling για το μέλλον των δεδομένων της εφαρμογής.

Κύρια Χαρακτηριστικά

- **Ad-hoc ερωτήματα:** Το MongoDB υποστηρίζει αναζητήσεις πεδίου, ερωτήματος εύρους και κανονικής έκφρασης. Τα ερωτήματα μπορούν να επιστρέψουν συγκεκριμένα πεδία εγγράφων και επίσης να περιλαμβάνουν λειτουργίες JavaScript καθορισμένες από το χρήστη. Τα ερωτήματα μπορούν επίσης να ρυθμιστούν ώστε να επιστρέφουν ένα τυχαίο δείγμα αποτελεσμάτων ενός δεδομένου μεγέθους.
- **Indexing:** Τα πεδία σε ένα έγγραφο MongoDB μπορούν να γίνουν indexing με πρωτεύοντες και δευτερεύοντες δείκτες.
- **Replication:** Το MongoDB παρέχει υψηλή διαθεσιμότητα με σερβιέρ. Ένα σύνολο αντιγράφων αποτελείται από δύο ή περισσότερα αντίγραφα των δεδομένων. Κάθε μέλος που ορίζεται σε ρεπλίκα μπορεί να ενεργεί στο ρόλο του πρωτεύοντος ή του δευτερεύοντος αντιγράφου ανά πάσα στιγμή. Όλες οι εγγραφές και οι αναγνώσεις γίνονται στο πρωτότυπο αντίγραφο από προεπιλογή. Τα δευτερεύοντα αντίγραφα διατηρούν ένα αντίγραφο των δεδομένων του πρωτεύοντος χρησιμοποιώντας ενσωματωμένη αναπαραγωγή. Όταν αποτυγχάνει ένα πρωτότυπο αντίγραφο, το σερβιέρ πραγματοποιεί αυτόματα μια διαδικασία εκλογής για να προσδιορίσει ποιο δευτερεύον θα πρέπει να γίνει το πρωτεύον. Οι δευτερεύοντες μπορούν προαιρετικά να εξυπηρετούν λειτουργίες ανάγνωσης, αλλά αυτά τα δεδομένα είναι τελικά τελικά συνεπή από προεπιλογή.
- **Load balancing:** Το MongoDB κλιμακώνεται οριζόντια χρησιμοποιώντας shards. Ο χρήστης επιλέγει ένα shard key, το οποίο καθορίζει τον τρόπο διανομής των δεδομένων σε μια συλλογή. Τα δεδομένα χωρίζονται σε περιοχές (βάσει το shard key) και κατανέμονται σε πολλά shards. (Ένα shard είναι master με ένα ή περισσότερα αντίγραφα.) Εναλλακτικά, το shard key μπορεί να κατακερματιστεί για να χαρτογραφηθεί σε shard - επιτρέποντας μια ομοιόμορφη κατανομή δεδομένων. Το MongoDB μπορεί να εκτελεστεί σε πολλούς διακομιστές, εξισορροπώντας το φορτίο ή αντιγράφοντας δεδομένα για να διατηρήσει το σύστημα σε λειτουργία και λειτουργία σε περίπτωση βλάβης του υλικού.
- **File storage:** Το MongoDB μπορεί να χρησιμοποιηθεί ως σύστημα αρχείων, που ονομάζεται GridFS, με δυνατότητες εξισορρόπησης φορτίου και αναπαραγωγής δεδομένων σε πολλά μηχανήματα για την αποθήκευση αρχείων. Αυτή η λειτουργία, που ονομάζεται σύστημα αρχείων πλέγματος, περιλαμβάνεται σε προγράμματα οδήγησης MongoDB. Το MongoDB εκθέτει λειτουργίες για χειρισμό αρχείων και περιεχόμενο στους προγραμματιστές. Το GridFS χωρίζει ένα αρχείο σε τμήματα ή κομμάτια και αποθηκεύει κάθε ένα από αυτά τα κομμάτια ως ξεχωριστό έγγραφο.

- **Aggregation:** Το MongoDB παρέχει τρεις τρόπους για εκτέλεση των aggregation:

το aggregation pipeline, map-reduce function και single-purpose aggregation μεθόδους. Η μείωση χαρτών μπορεί να χρησιμοποιηθεί για την επεξεργασία παρτίδων δεδομένων και διαδικασίες aggregation. Αλλά σύμφωνα με την τεκμηρίωση της MongoDB, ο aggregation pipeline παρέχει καλύτερη απόδοση για τις περισσότερες aggregation λειτουργίες. Το aggregation framework επιτρέπει στους χρήστες να λαμβάνουν το είδος των αποτελεσμάτων για τα οποία χρησιμοποιείται η ρήτρα SQL GROUP BY. Οι aggregation operators μπορούν να ενωθούν για να σχηματίσουν έναν αγωγό - ανάλογο με τους pipeline των Unix. Το aggregation framework περιλαμβάνει τον τελεστή αναζήτησης \$ που μπορεί να ενώσει έγγραφα από πολλές συλλογές, καθώς και στατιστικούς φορείς όπως η τυπική απόκλιση.

- **Εκτέλεση JavaScript από διακομιστή:** Το JavaScript μπορεί να χρησιμοποιηθεί σε ερωτήματα, συναρτήσεις συνάθροισης (όπως το MapReduce) και να σταλεί απευθείας στη βάση δεδομένων για εκτέλεση.
- **Capped collections:** Το MongoDB υποστηρίζει συλλογές σταθερού μεγέθους που ονομάζονται περιορισμένες συλλογές. Αυτός ο τύπος συλλογής διατηρεί τη σειρά εισαγωγής και, μόλις επιτευχθεί το καθορισμένο μέγεθος, συμπεριφέρεται σαν κυκλική ουρά.

5.4 Οι υπηρεσίες ως containers

Containers

Τα container είναι παρόμοια με τα VM, αλλά έχουν χαλαρές ιδιότητες απομόνωσης για κοινή χρήση του λειτουργικού συστήματος (OS) μεταξύ των εφαρμογών. Επομένως, τα container θεωρούνται ελαφριά. Παρόμοια με ένα VM, ένα container έχει το δικό του σύστημα αρχείων, μέρος της CPU, μνήμη, χώρο επεξεργασίας και πολλά άλλα. Καθώς αποσυνδέονται από την υποκείμενη υποδομή, είναι φορητές σε σύννεφα και διανομές λειτουργικών συστημάτων.

Τα container έχουν γίνει δημοφιλή επειδή παρέχουν επιπλέον πλεονεκτήματα, όπως:

- Ευέλικτη δημιουργία και ανάπτυξη εφαρμογών: αυξημένη ευκολία και αποτελεσματικότητα της δημιουργίας image του container σε σύγκριση με τη χρήση image της VM.
- Συνεχής ανάπτυξη, ενσωμάτωση και deployment: παρέχει αξιόπιστη και συχνή δημιουργία και ανάπτυξη image του container με γρήγορες και αποτελεσματικές επαναφορές (λόγω της αμετάβλητου image).
- Διαχωρισμός ανησυχιών για προγραμματιστές και λειτουργίες: δημιουργία των images εφαρμογών κατά το χρόνο κατασκευής / έκδοσης αντί για χρόνο ανάπτυξης, αποσυνδέοντας έτσι εφαρμογές από την υποδομή.
- Η δυνατότητα παρακολούθησης όχι μόνο εμφανίζει πληροφορίες και μετρήσεις σε επίπεδο λειτουργικού συστήματος, αλλά και την κατάσταση των εφαρμογών και άλλα σήματα.
- Περιβαλλοντική συνέπεια σε όλη την ανάπτυξη, τις δοκιμές και την παραγωγή: Τρέχει το ίδιο σε φορητό υπολογιστή όπως και στο cloud.

- Φορητότητα διανομής Cloud και OS: Εκτελείται σε Ubuntu, RHEL, CoreOS, εσωτερικά, σε μεγάλα δημόσια σύννεφα και οπουδήποτε αλλού.
- Διαχείριση με επίκεντρο την εφαρμογή: Αυξάνει το επίπεδο αφαίρεσης από την εκτέλεση ενός λειτουργικού συστήματος σε εικονικό υλικό σε εκτέλεση μιας εφαρμογής σε ένα λειτουργικό σύστημα χρησιμοποιώντας λογικούς πόρους.
- Χαλαρά συνδεδεμένα, κατανεμημένα, ελαστικά, απελευθερωμένα microservices: οι εφαρμογές χωρίζονται σε μικρότερα, ανεξάρτητα κομμάτια και μπορούν να αναπτυχθούν και να διαχειριστούν δυναμικά - όχι μια μονολιθική στοίβα που λειτουργεί σε ένα μεγάλο μηχάνημα μιας χρήσης.
- Απομόνωση πόρων: προβλέψιμη απόδοση εφαρμογής. Χρήση πόρων: υψηλή απόδοση και πυκνότητα.

Γίνεται χρήση της πλατφόρμας Kubernetes για την ανάπτυξη των υπηρεσιών της εφαρμογής SamosAlive ως containers αλλά και χρήση υπηρεσιών τρίτων οι οποίες είναι άμεσα συνδεδεμένες με τις αναπτυσσόμενες δικές μας υπηρεσίες. Μέσα σε αυτές περιέχονται οι Redis, MongoDB, nominatim και osrm. Υπάρχει δυνατότητα δημιουργίας εσωτερικών δικτύων για να παραμένουν κάποιες υπηρεσίες private, αλλά παράλληλα να χρησιμοποιούνται εντός άλλων δημόσιων υπηρεσιών.

To Kubernetes ως Container Orchestrator

Overview

Το Kubernetes είναι μια φορητή, επεκτάσιμη, ανοιχτού κώδικα πλατφόρμα για τη διαχείριση φορτίων εργασίας και υπηρεσιών με containers, που διευκολύνει τόσο τη δηλωτική διαμόρφωση όσο και τον αυτοματισμό.

Γιατί το Kubernetes είναι απαραίτητο για το SamosAlive;

Σε ένα περιβάλλον παραγωγής, πρέπει να υπάρχει οι δυνατότητα από τους ανθρώπους που θα συντηρούν το SamosAlive να διαχειριστούν τα container που εκτελούν τις εφαρμογές και να διασφαλιστεί ότι δεν θα υπάρχει διακοπή λειτουργίας. Για παράδειγμα, εάν ένα container κατεβαίνει, πρέπει να ξεκινήσει ένα άλλο container. Αυτή την συμπεριφορά και πολλά άλλα χαρακτηριστικά προσφέρει το σύστημα του Kubernetes για το SamosAlive στην παραγωγή.

Χαρακτηριστικά Kubernetes

Αυτοματοποιημένες κυκλοφορίες και επαναφορές

Το Kubernetes παρουσιάζει σταδιακά αλλαγές στην εφαρμογή ή στη διαμόρφωσή της, παρακολουθώντας ταυτόχρονα την υγεία της εφαρμογής για διασφάλιση ότι δεν θα σκοτώσει όλες τις παρουσίες σας ταυτόχρονα. Εάν κάτι πάει στραβά, το Kubernetes θα επαναφέρει την αλλαγή για εσάς. Επωφεληθείτε από ένα αναπτυσσόμενο οικοσύστημα λύσεων ανάπτυξης.

Ανακάλυψη υπηρεσιών και εξισορρόπηση φορτίου

Δεν χρειάζεται να τροποποιήσετε την εφαρμογή σας για να χρησιμοποιήσετε έναν άγνωστο μηχανισμό ανακάλυψης υπηρεσιών. Το Kubernetes δίνει στους Pods τις

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου

δικές τους διευθύνσεις IP και ένα μόνο όνομα DNS για ένα σύνολο Pods και μπορούν να ισορροπήσουν φορτία μεταξύ τους.

Τοπολογία υπηρεσίας

Δρομολόγηση της κυκλοφορίας υπηρεσιών με βάση την τοπολογία συμπλέγματος.

Ενορχήστρωση αποθήκευσης

Αυτόματη σύνδεση σε οποιοδήποτε σύστημα αποθήκευσης, είτε από τοπικό χώρο αποθήκευσης, δημόσιο πάροχο cloud όπως GCP ή AWS, ή σύστημα αποθήκευσης δικτύου όπως NFS, iSCSI, Gluster, Ceph, Cinder ή Flocker.

Διαχείριση μυστικών και διαμόρφωσης

Ανάπτυξη και ενημέρωση μυστικών και διαμόρφωση εφαρμογών χωρίς να αναδημιουργία των images των microservices του SamosAlive και χωρίς να αποκάλυψη των μυστικών στη διαμόρφωση της στοίβας.

Αυτόματη συσκευασία κάδου

Τοποθετεί αυτόματα container με βάση τις απαιτήσεις πόρων και άλλους περιορισμούς, χωρίς να θυσιάζει τη διαθεσιμότητα. Συνδυάζει κρίσιμους και καλύτερους φόρτους εργασίας για αύξηση χρήσης και να εξοικονόμηση ακόμα περισσότερων πόρων.

Μαζική εκτέλεση

Εκτός από τις υπηρεσίες, το Kubernetes μπορεί να διαχειριστεί το φόρτο εργασίας batch και CI, αντικαθιστώντας τα container που αποτυγχάνουν, εάν είναι επιθυμητό.

Διπλή στοίβα IPv4 / IPv6

Κατανομή διευθύνσεων IPv4 και IPv6 σε Pods και Υπηρεσίες

Οριζόντια κλιμάκωση

Κλιμάκωση των microservices της εφαρμογής πάνω και κάτω με μια απλή εντολή, με διεπαφή χρήστη ή αυτόματα με βάση τη χρήση της CPU.

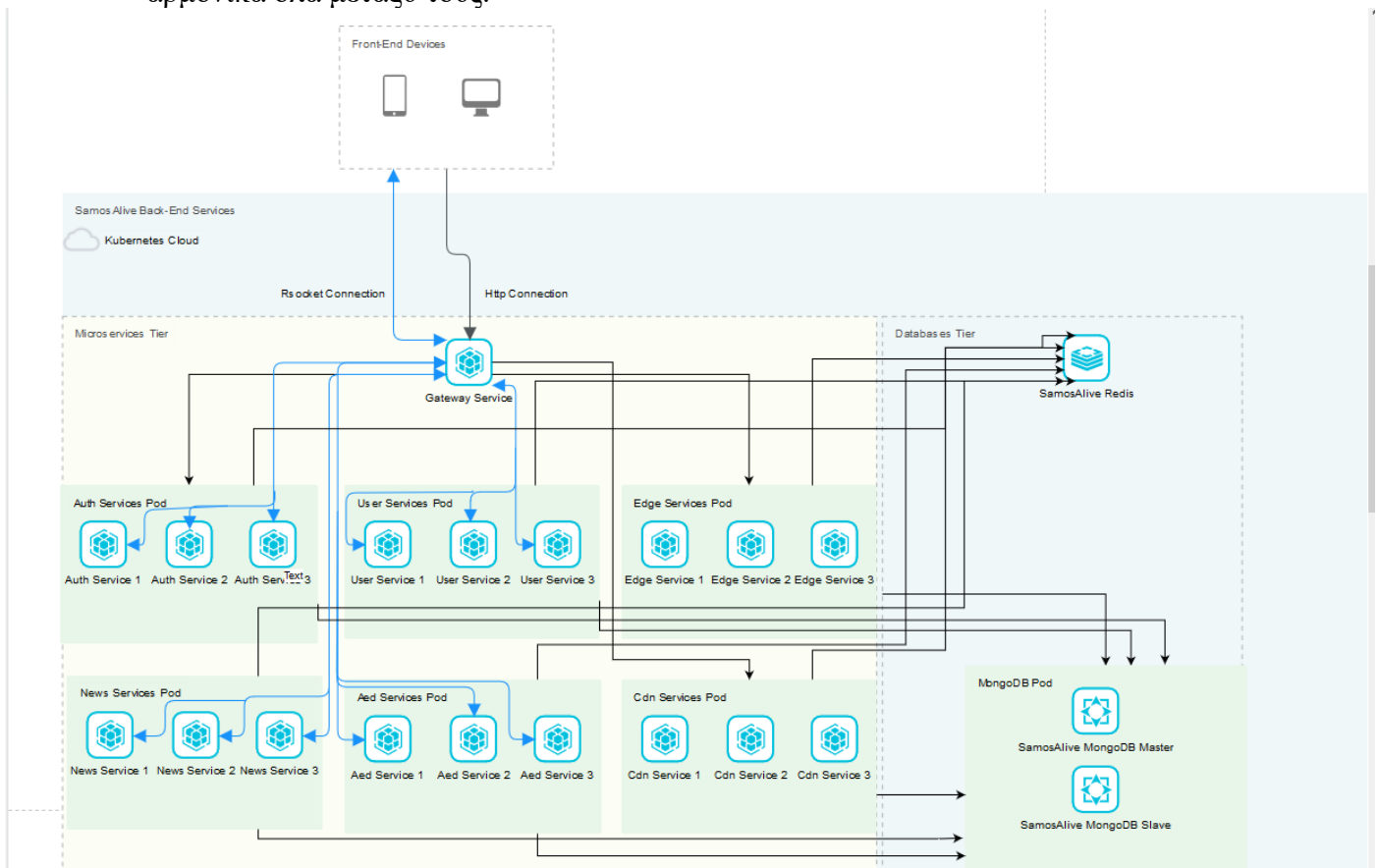
Αυτοθεραπεία

Επανεκκίνηση container που αποτυγχάνουν, αντικαθιστά και επαναπρογραμματίζει τα container όταν πεθαίνουν οι κόμβοι, σκοτώνει container που δεν ανταποκρίνονται στον έλεγχο υγείας που καθορίζεται από τον χρήστη και δεν τα διαφημίζει σε άλλους clients έως ότου είναι έτοιμα για προβολή.

5.5 Flowcharts της εφαρμογής SamosAlive

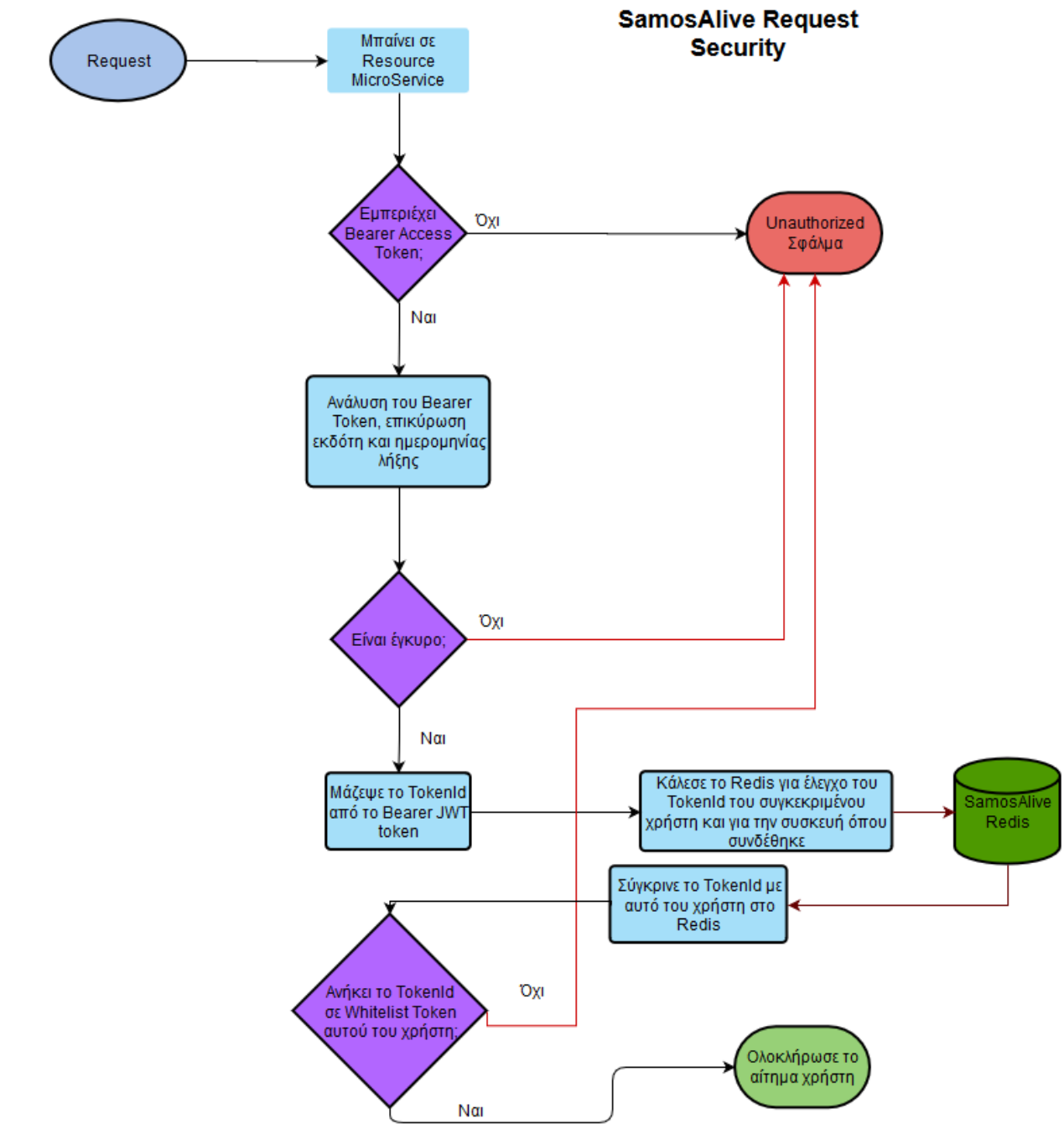
Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου

Αρχικά βλέπουμε το πως είναι αρχιτεκτονικά όλη η εφαρμογή και πως λειτουργούν αρμονικά όλα μεταξύ τους.



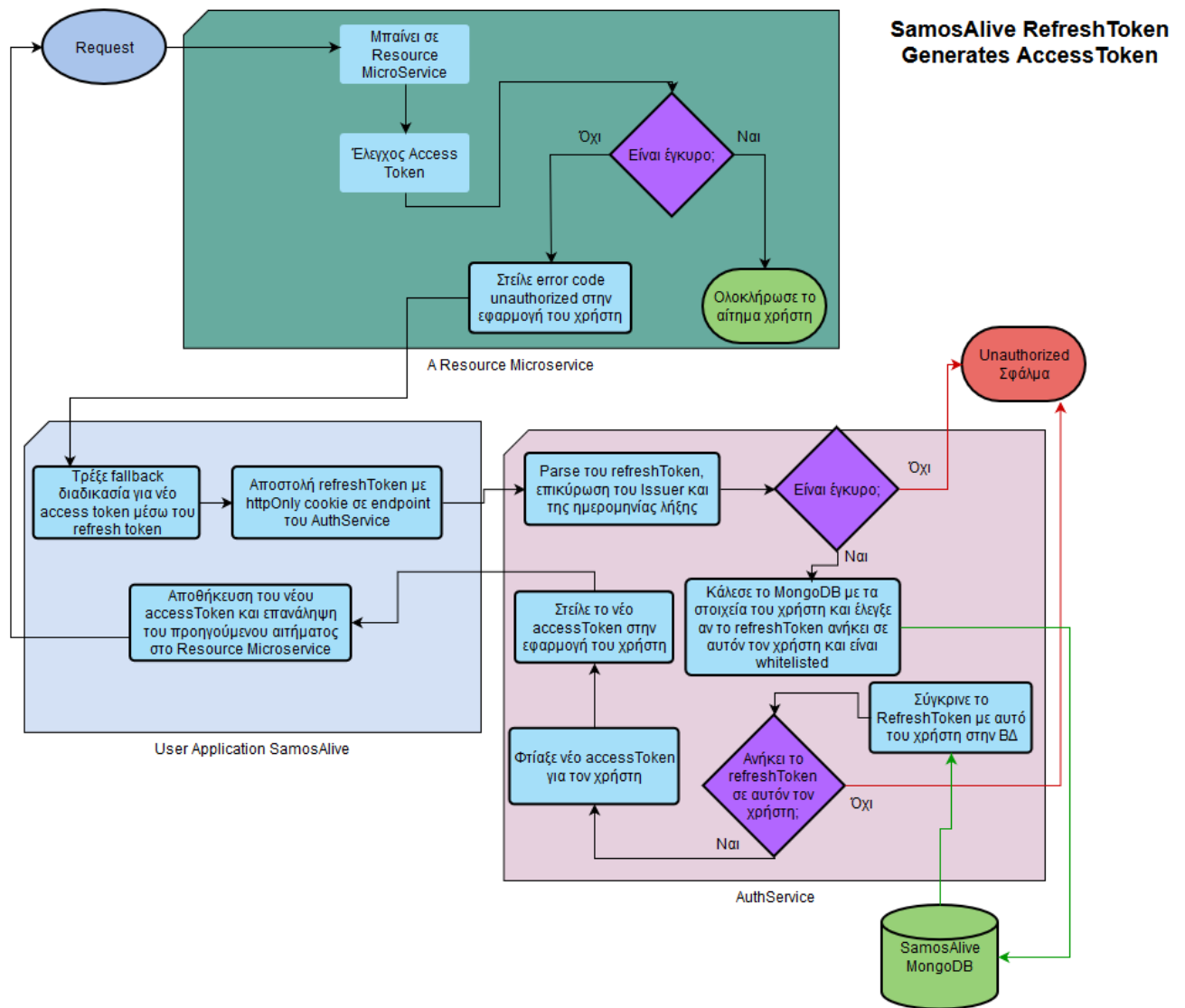
Σχήμα 5.6: Αρχιτεκτονική του συστήματος

Εδώ φαίνεται πως από την κινητή συσκευή γίνεται η σύνδεση στα διάφορα services που απαιτούνται για την λειτουργία που θέλει να εκτελέσει ο χρήστης. Για να γίνει όπως πιο κατανοητό το σχήμα αλλά και όλη η διασύνδεση της εφαρμογής, θα γίνει διαχωρισμός και μοντελοποίηση κάποιων βασικών λειτουργιών στην εφαρμογή ξεκινώντας από τους τρόπους απόκτησης, ανανέωσης και επαλήθευσης των token του χρήστη.



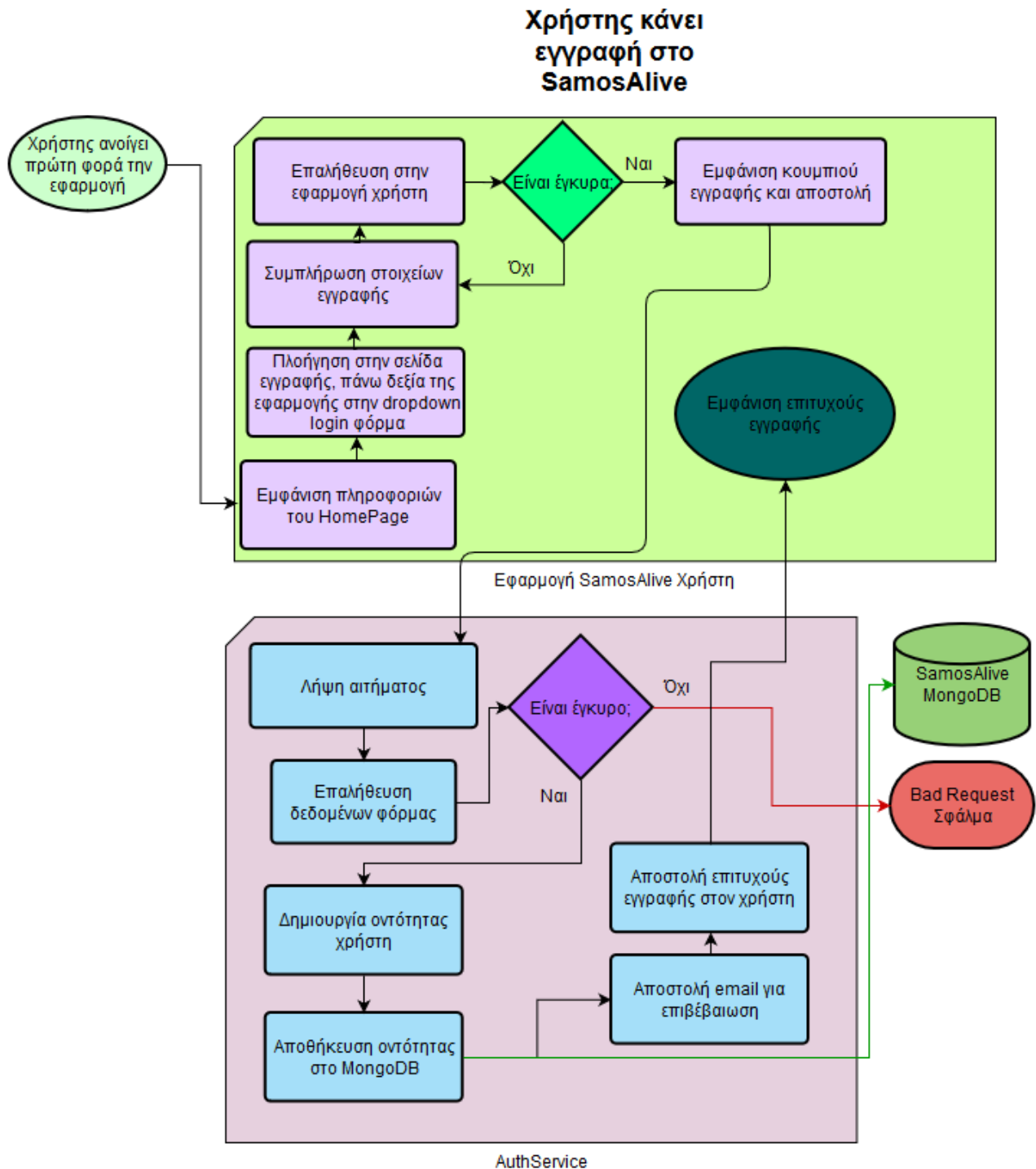
Σχήμα 5.7: Η ασφάλεια πίσω από ένα αίτημα στο SamosAlive

Σχεδίαση και Ανάπτυξη Πρωτότυπης Εφαρμογής σε Κινητές Συσκευές για την Διαχείριση Απινιδωτών, με εφαρμογή στο Καρλόβασι Σάμου



Σχήμα 5.8: Ανανέωση του Access Token μέσω του Refresh Token

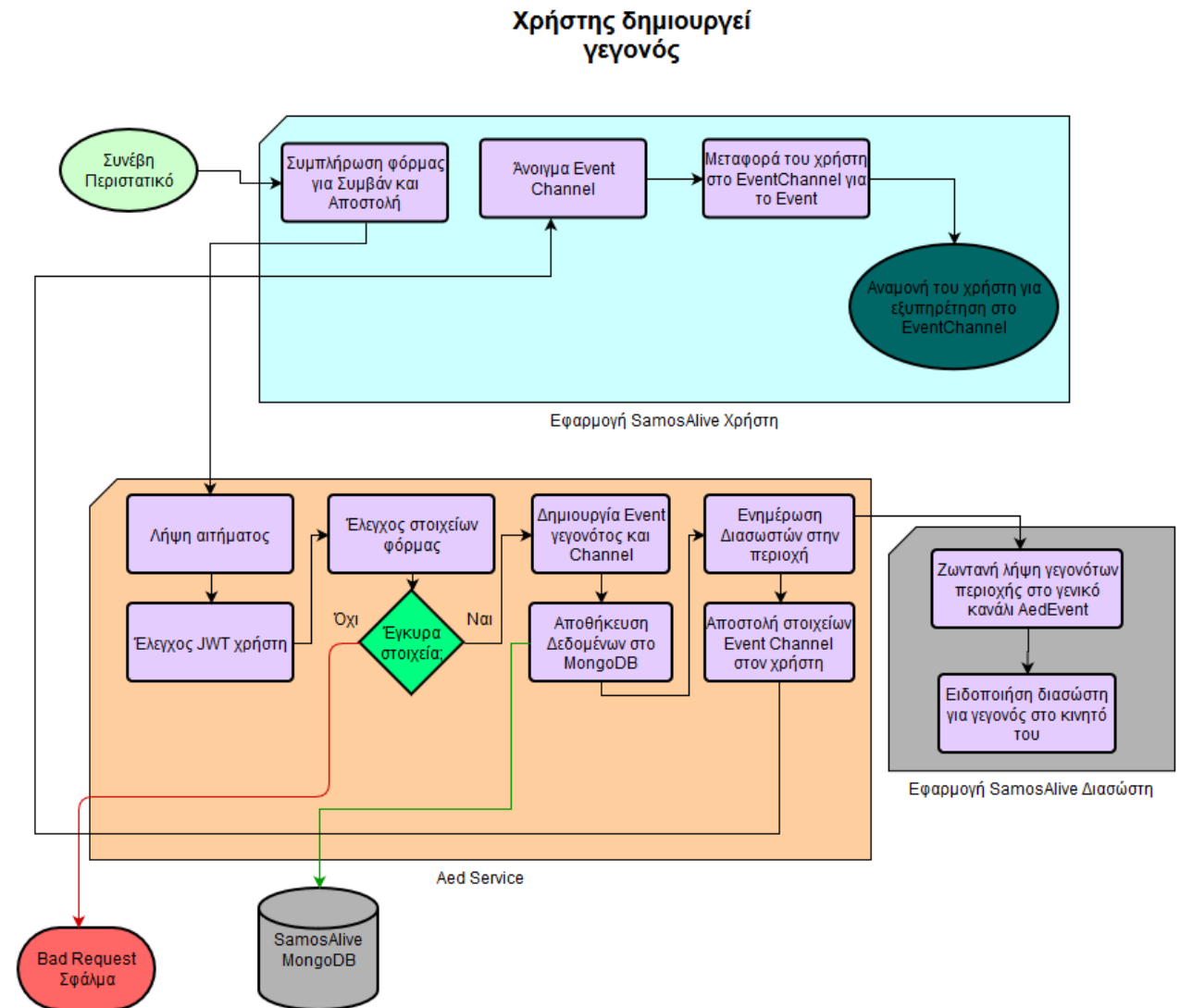
Συνεχίζουμε με τους τρόπους που κάποιος χρήστης μπορεί να συνδεθεί στην εφαρμογή.



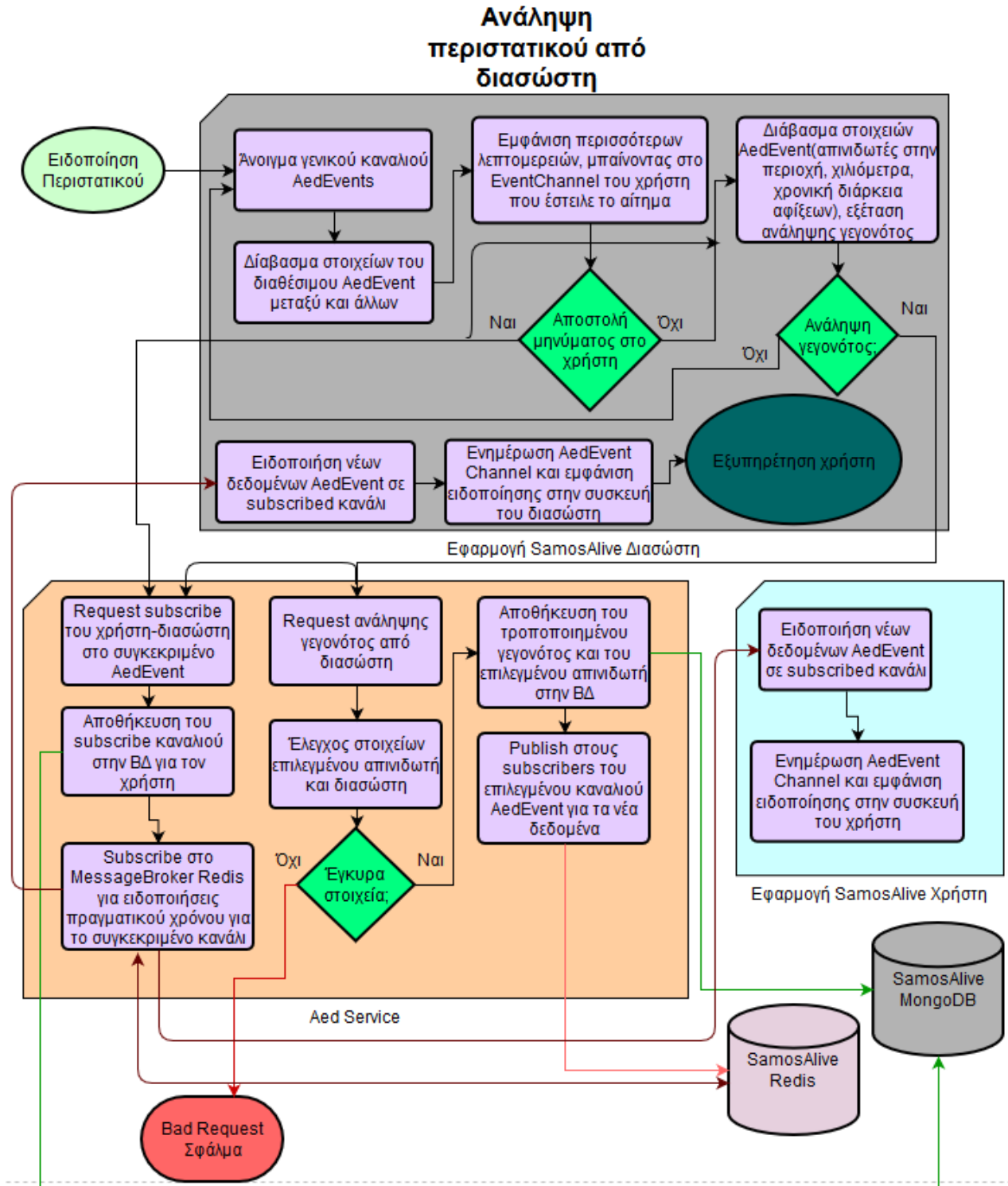
Σχήμα 5.9: Εγγραφή χρήστη στην πλατφόρμα



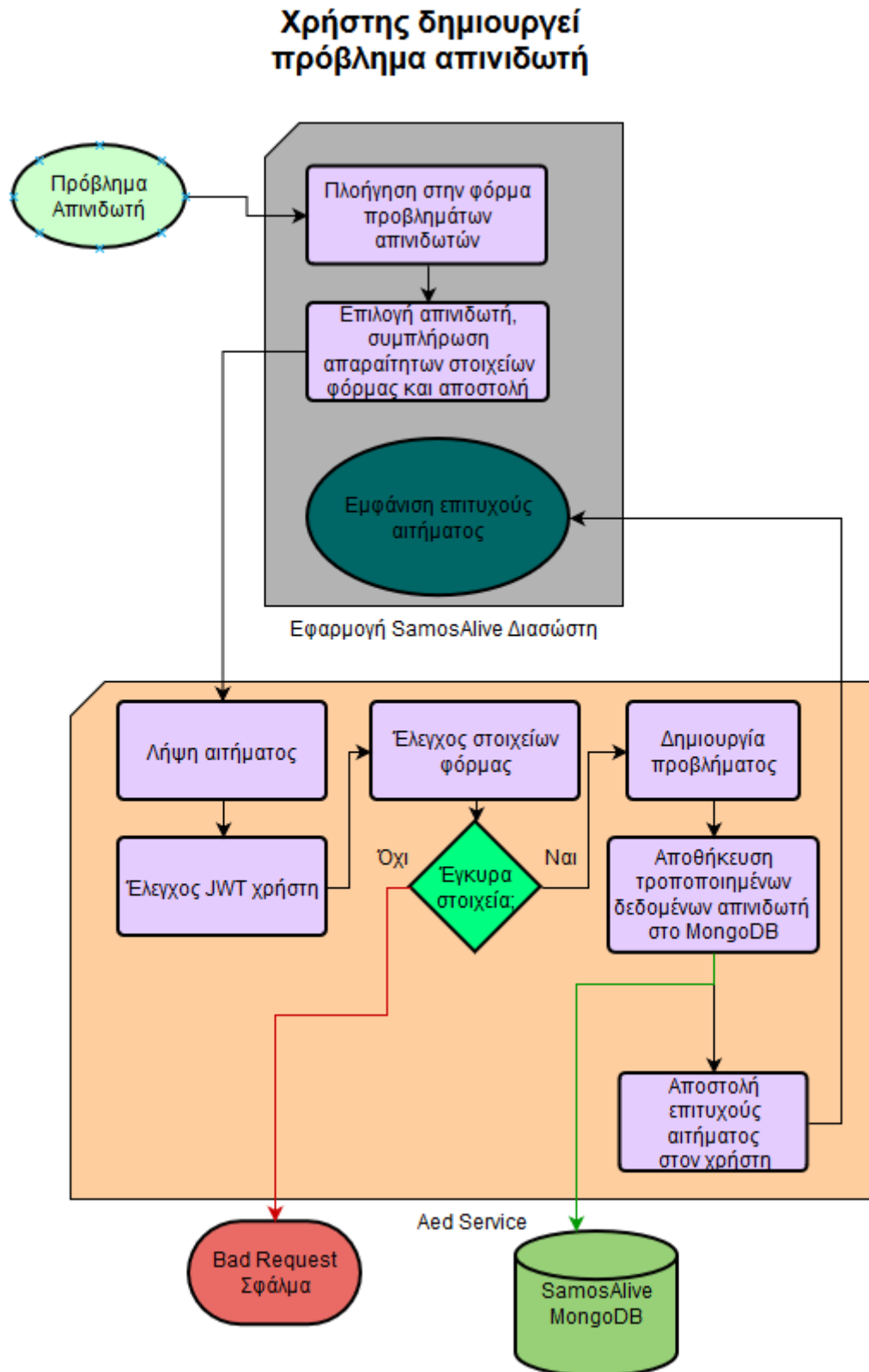
Τέλος θα γίνει αναπαράσταση κάποιων κύριων λειτουργιών που φιλοξενεί η πλατφόρμα όσο αφορά το «συμβάν», τους διασώστες και τους απινιδωτές.



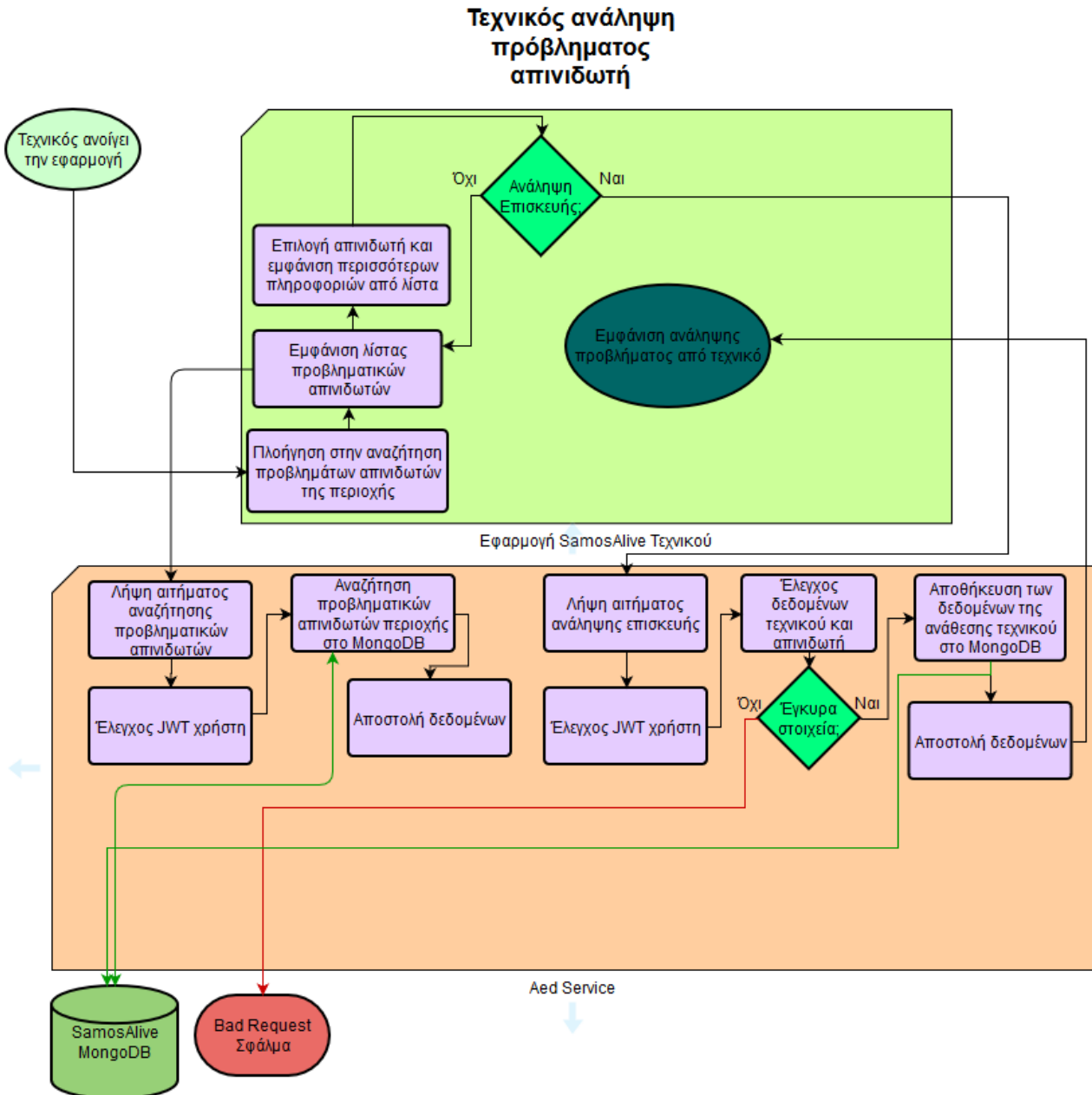
Σχήμα 5.11: Ο χρήστης αναφέρει κάποιο περιστατικό



Σχήμα 5.12: Ο διασώστης επιλέγει από την λίστα με τα συμβάντα κάποιο για να βοηθήσει



Σχήμα 5.13: Αναφορά προβλήματος που αφορά έναν απινιδωτή από χρήστη



Σχήμα 5.14: Τεχνικός αναλαμβάνει πρόβλημα για να το επιλύσει

Με παρόμοιο τρόπο όπως το συμβάν ή το πρόβλημα του απινιδωτή γίνονται και οι υπόλοιπες λειτουργίες στην πλατφόρμα. Η επικοινωνία με το αντίστοιχο service κάθε φορά, η επικοινωνία με την βάση δεδομένων καθώς και το θετικό ή αρνητικό τελικό μήνυμα στον χρήστη περνάνε μέσα από μια διαδικασία ώστε να διασφαλίζεται η ομαλή χρήση της εφαρμογής καθώς και η ασφάλεια των χρηστών και του SamosAlive.

6. Ανάλυση πλατφόρμας

6.1 Η πρώτη επαφή με την εφαρμογή

Ο κάθε πολίτης που θα θέλει να κατέχει την εφαρμογή θα πρέπει να την κάνει εγκατάσταση στο κινητό του. Όταν ανοίγει το Samos Alive για πρώτη φορά θα ζητηθούν δικαιώματα για να έχει η εφαρμογή πρόσβαση στην τοποθεσία του χρήστη. Στο επόμενο βήμα ο χρήστης κάνει εγγραφή στην πλατφόρμα ακολουθώντας μια απλή διαδικασία ή σύνδεση αν έχει κάνει ήδη εγγραφή. Μετά την δημιουργία του λογαριασμού μπορεί ο καθένας ο χρησιμοποιήσει πλήρως την εφαρμογή.

Εικόνα 6.10: Εγγραφή νέου χρήστη στην εφαρμογή

Εικόνα 6.11: Σύνδεση υπάρχοντος χρήστη με τα στοιχεία του

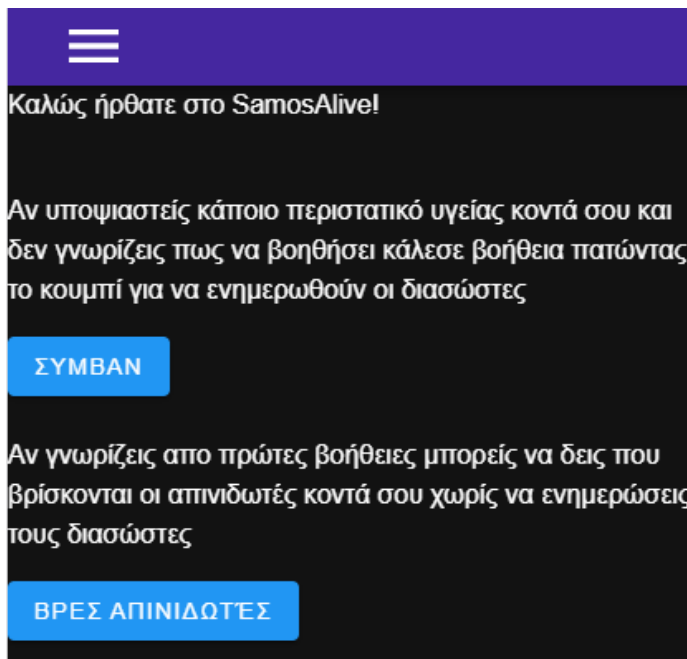
6.2 Η αναφορά ενός συμβάντος

Στην αρχική σελίδα της εφαρμογής ο χρήστης θα βρεθεί ανάμεσα σε δύο βασικές επιλογές (βλ. Εικόνα 12). Αν έχει συμβεί κάποιο περιστατικό κοντά του και δεν γνωρίζει από πρώτες βοήθειες μπορεί να πατήσει το κουμπί «Συμβάν». Με το πάτημα του κουμπιού θα γίνει ανακατεύθυνση στην υπεύθυνη φόρμα για αναφορά ενός γεγονότος όπου ενημερώνονται και οι διασώστες. Η δεύτερη επιλογή που έχει αν γνωρίζει από χρήση απινιδωτών και πρώτων βοηθειών είναι να βρει κοντινούς απινιδωτές από τον ίδιο πατώντας το δεύτερο κουμπί. Ο χρήστης όμως πρέπει να είναι σίγουρος ότι ξέρει πως να βοηθήσει γιατί αλλιώς παίρνει μεγάλο ρίσκο.

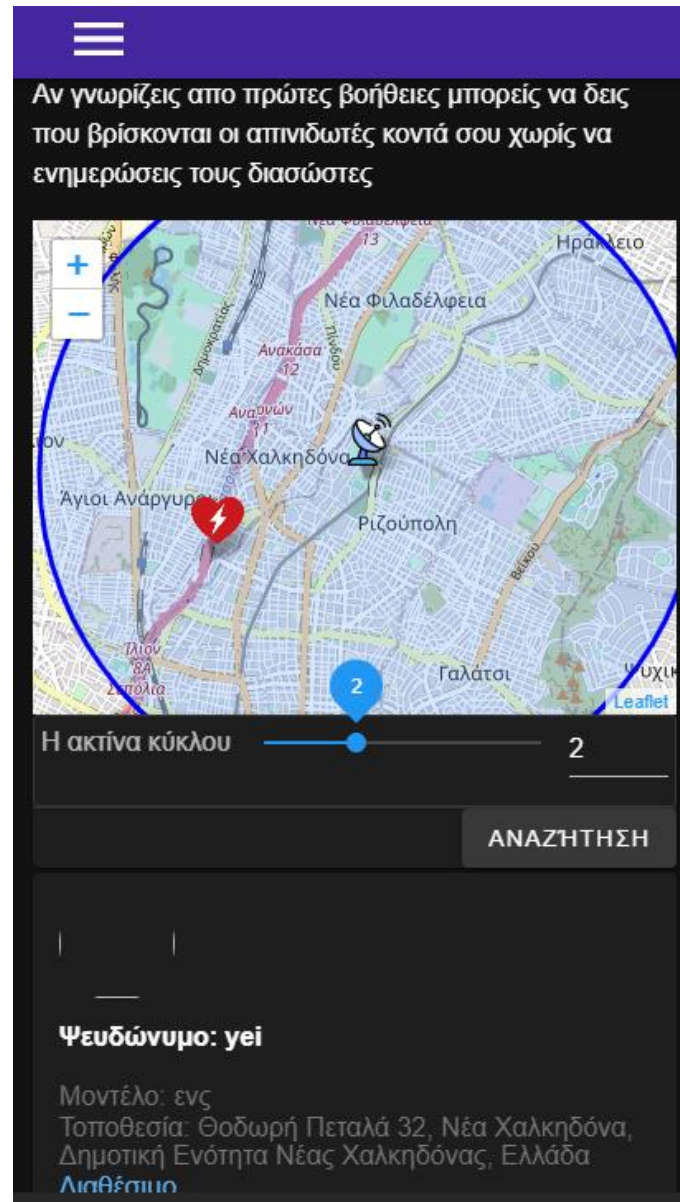
Με την τελευταία αυτή επιλογή ο χρήστης κατευθύνεται σε μια άλλη σελίδα με έναν χάρτη όπου από εκεί μπορεί να κάνει αναζήτηση απινιδωτών (βλ. εικόνα13). Κουνώντας την κουκίδα του στον χάρτη μπορεί να επιλέξει την γεωγραφική του θέση αν δεν είναι σωστή εξαρχής. Από εκεί και πέρα έχει την επιλογή να διαλέξει πόσα χιλιόμετρα μακριά θέλει να ορίζει την περίμετρο της αναζήτησης απινιδωτή. Όταν επιλέξει τον επιθυμητό αριθμό από την συρόμενη μπάρα πατάει αναζήτηση και του εμφανίζονται στον χάρτη αλλά και από κάτω σαν λίστα οι κοντινοί διαθέσιμοι απινιδωτές με σχήμα καρδιάς. Με την επιλογή κάποιου απινιδωτή πηγαίνει στο προφίλ του όπου μπορεί να ελέγχει διάφορες πληροφορίες για αυτόν όπως τον τύπο του αλλά και την ακριβή γεωγραφική του θέση πάνω στον χάρτη.

Η άλλη επιλογή του χρήστη είναι να αναφέρει το συμβάν μέσω του κουμπιού «Συμβάν» και να ειδοποιηθούν οι διασώστες. Μέσα σε λίγα βήματα ο χρήστης συμπληρώνει και αποστέλλει την φόρμα η οποία περιέχει αρκετές πληροφορίες αυτοματοποιημένες ώστε να μην χάνεται άσκοπος χρόνος.

Όταν ξεκινάει η φόρμα πρέπει να επιλέξει τι είδος συμβάντος διαδραματίζεται (βλ. εικόνα 14). Στην συνέχεια συμπληρώνει το τηλέφωνο του και το όνομα του για να γνωρίζουν οι διασώστες με ποιον πρέπει να επικοινωνήσουν αν χρειαστεί. Έπειτα επιλέγει την τοποθεσία που βρίσκεται το συμβάν με την βοήθεια ενός χάρτη. Γράφοντας ένα προαιρετικό σχόλιο για την διεύθυνση ή το περιστατικό ολοκληρώνει την φόρμα όπου περιέχονται όλες οι αναγκαίες πληροφορίες.



Εικόνα 6.12: Αρχική σελίδα SamosAlive



Εικόνα 6.13: Σελίδα αναζήτησης κοντινών απινιδωτών

Συμπληρώστε την φόρμα

Επέλεξε κατάσταση ασθενούς

Έχει αρρυθμίες

Καλών
liaros tom
10 / 120

Αρ. τηλεφώνου
6958574353
10 / 10

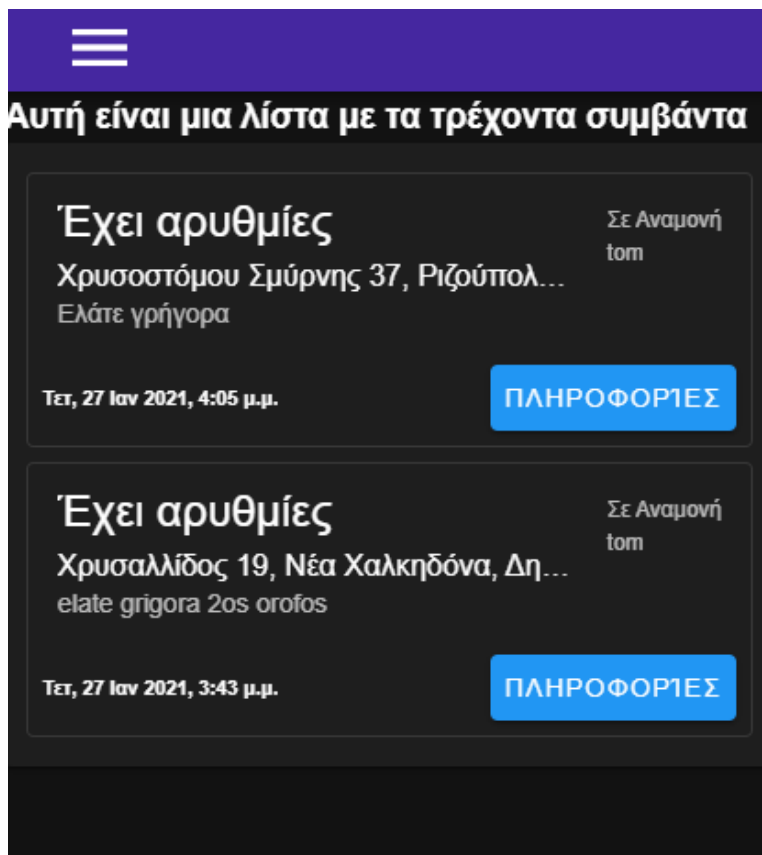
Διεύθυνση:
Χρυσοστόμου Σμύρνης 37, Ριζούπολη, Αθή

Εικόνα 6.14: Δημιουργία και αναφορά συμβάντος

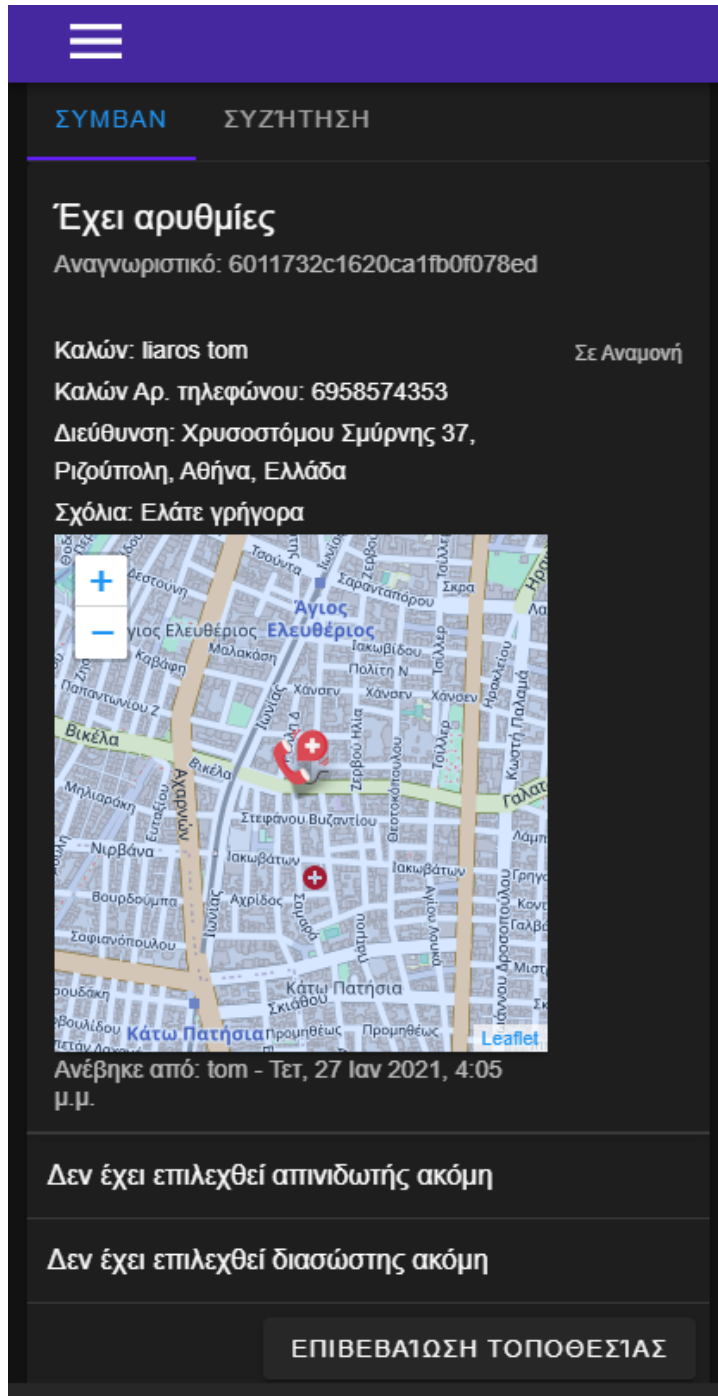
6.3 Η ανάθεση ενός γεγονότος σε διασώστη

Για την συνέχεια και την επίλυση του γεγονότος που αναφέρθηκε υπάρχει ειδική που έχουν πρόσβαση μόνο οι διασώστες όπου εμφανίζονται ζωντανά όλα τα συμβάντα (βλ. εικόνα 15). Ο κάθε διασώστης από εκεί και πέρα αν είναι διαθέσιμος να βοηθήσει επιλέγει ένα περιστατικό που είναι κοντά του και αφού δει τις κατάλληλες πληροφορίες που έχει συμπληρώσει ο χρήστης αποφασίζει αν θα βοηθήσει (βλ. εικόνα 16). Αν πατήσει το κουμπί για ανάθεση του εμφανίζονται στον χάρτη κοντινοί διαθέσιμοι απινιδωτές και η καλύτερη διαδρομή που μπορεί να ακολουθήσει από την θέση που βρίσκεται, στον απινιδωτή και έπειτα στο συμβάν.

Για να μην γίνει λάθος διαχείριση χρόνου και περιστατικών όταν ένας διασώστης αναλάβει κάποιο συμβάν και επιλέξει από τον απινιδωτή που θα πάρει οι «καταστάσεις» του γεγονότος και του απινιδωτή αλλάζουν. Σε περίπτωση λόγου χάρη που έχουμε σε μια περιοχή δύο απινιδωτές και δύο περιστατικά με την ανάθεση του ενός από αυτών τότε ο δεύτερος διασώστης μένει να πάει στο εναπομένον περιστατικό και στον χάρτη του εμφανίζεται μόνο ο διαθέσιμος απινιδωτής, λόγω του ότι ο άλλος είναι δεσμευμένος από τον πρώτο διασώστη. Μια ακόμα χρήσιμη λειτουργία είναι η παράλληλη ύπαρξη σελίδας για συζήτηση μέσα στο κάθε συμβάν, όπου όλοι οι συμμετέχοντες στο περιστατικό μπορούν να επικοινωνήσουν μεταξύ τους για την διεκπεραίωση τυχόν προβλημάτων και ότι άλλο προκύψει.



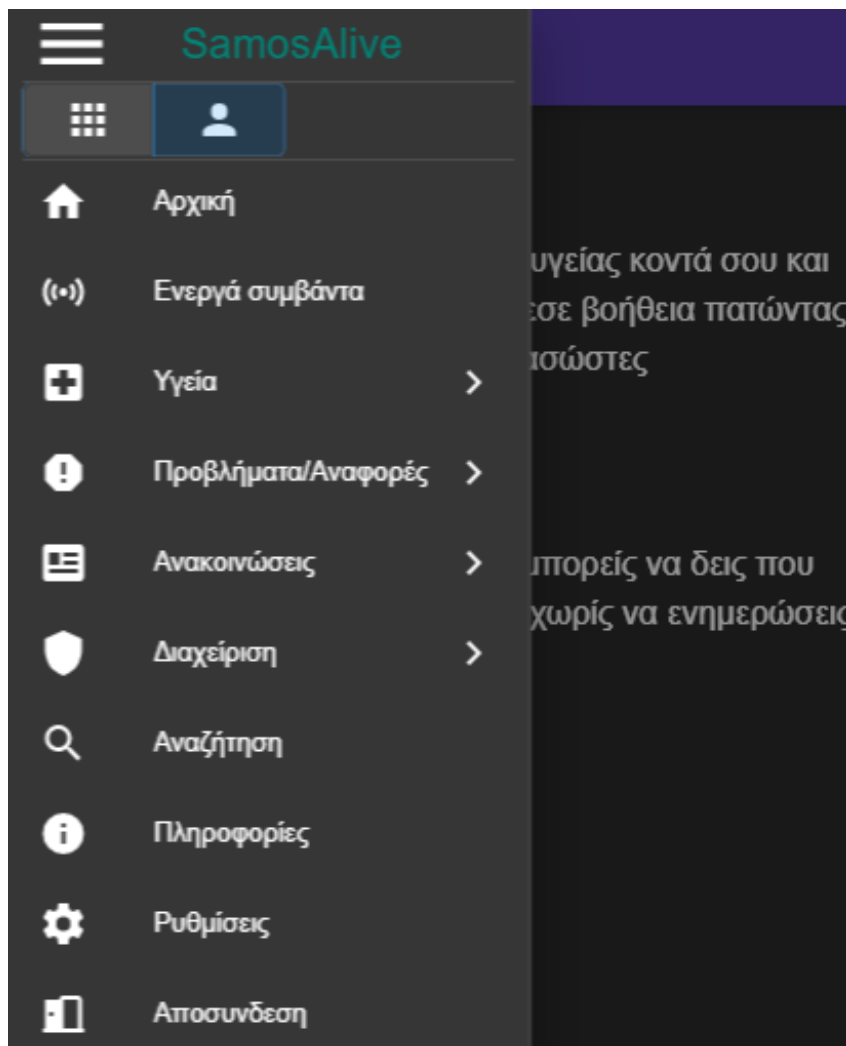
Εικόνα 6.15: Σελίδα με ζωντανά περιστατικά



Εικόνα 6.16: Πληροφορίες συμβάντος

6.4 Το μενού της εφαρμογής

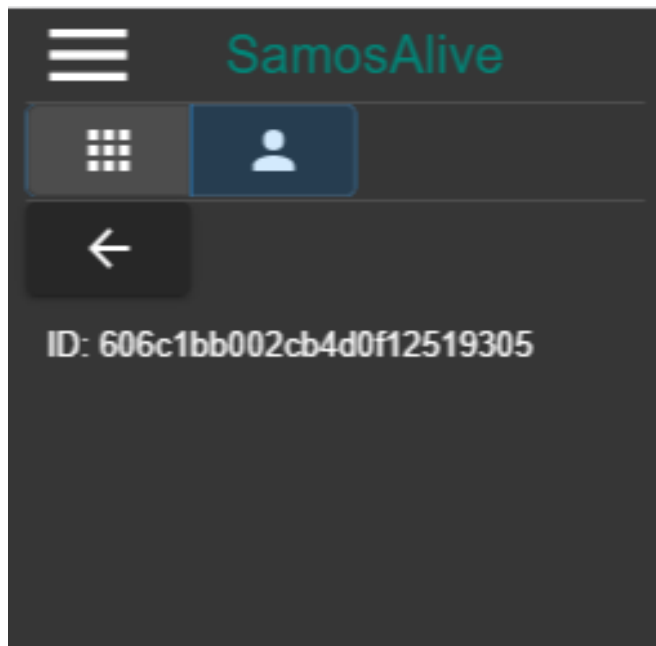
Η εφαρμογή πέρα από την δημιουργία και ανάθεση κάποιου περιστατικού υγείας παρέχει και άλλες λειτουργίες. Παρακάτω γίνεται μια σύντομη παρουσίαση του μενού του Samos Alive, εκτός των δύο βασικών λειτουργιών που αναφέραμε πριν (βλ. εικόνα 17). Βέβαια γίνεται παρουσίαση ολόκληρου του μενού, δηλαδή αυτό που φαίνεται στους διαχειριστές. Οι χρήστες με βάση τον ρόλο τους έχουν πρόσβαση σε περιορισμένα στοιχεία από αυτά.



Εικόνα 6.17: Βασικό μενού για εναλλαγή σελίδων στην εφαρμογή

6.4.1 Υγεία

Η επόμενη καρτέλα περιέχει επιλογές υγείας. Υπάρχει μια εναλλακτική διαδρομή στην φόρμα δημιουργίας συμβάντος αλλά και στα ζωντανά γεγονότα. Μια ακόμα σελίδα είναι εκείνη που μπορεί κάποιος διαχειριστής να εισάγει έναν απινιδωτή στην βάση δεδομένων συμπληρώνοντας τα στοιχεία του, φωτογραφίες και τοποθεσία (βλ. εικόνα 19). Τέλος μπορεί κάποιος να μπει από εκεί στην επιλογή παρακολούθησης συμβάντων για να δει την εξέλιξη των γεγονότων που έχει υποβάλει (βλ. εικόνα 18).



Εικόνα 6.18: Συμβάντα που έχει υπολάβει ο χρήστης

Nickname Device

Νέα συσκευή 4

13 / 30

Model Name

Aed2000

7 / 30

Model Description

Απινιδωτής

Give a Aed Device Description (Optional)

Aed Device Picture



defi.jpg (136.5 kB)





Address Picture



defi.jpg (136.5 kB)





Address

Ηρακλείου 356, Ηράκλειο, undefined, Ελλάδα







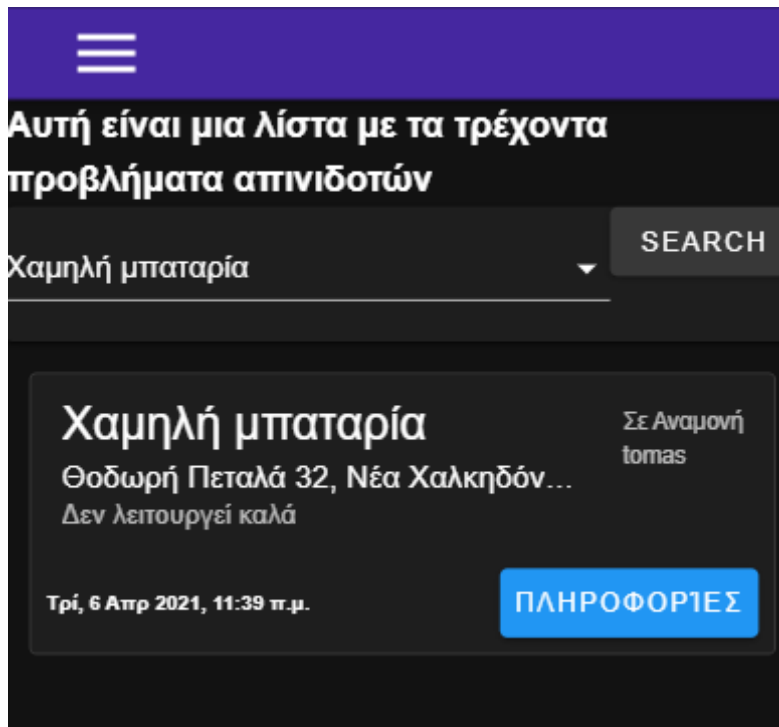


Εικόνα 6.19: Εγγραφή απινιδωτή με φωτογραφίες και τοποθεσία

6.4.2 Προβλήματα

Στην συνέχεια έχουμε την κατηγορία για τα προβλήματα. Οι απινιδωτές μπορεί να κλαπούν, να χαλάσουν ή να έχουν κάποιο πρόβλημα που να καθιστά την χρήση τους αδύνατη. Έτσι υπάρχει μια φόρμα αναφοράς προβλήματος όπου επιλέγοντας απινιδωτή και αναφέροντας το πρόβλημα του αλλάζει η διαθεσιμότητα του (βλ. εικόνα 20). Σε περίπτωση συμβάντος ένας τέτοιος απινιδωτής δεν εμφανίζεται σαν επιλογή για τους διασώστες. Επίσης υπάρχει αναζήτηση των ενεργών προβλημάτων ώστε κάποιος τεχνικός να μπορεί να δει πληροφορίες και να διορθώσει τυχόν βλάβες (βλ. εικόνα 21).

Εικόνα 6.20: Αναφορά προβλήματος σε επιλεγμένο απινιδωτή



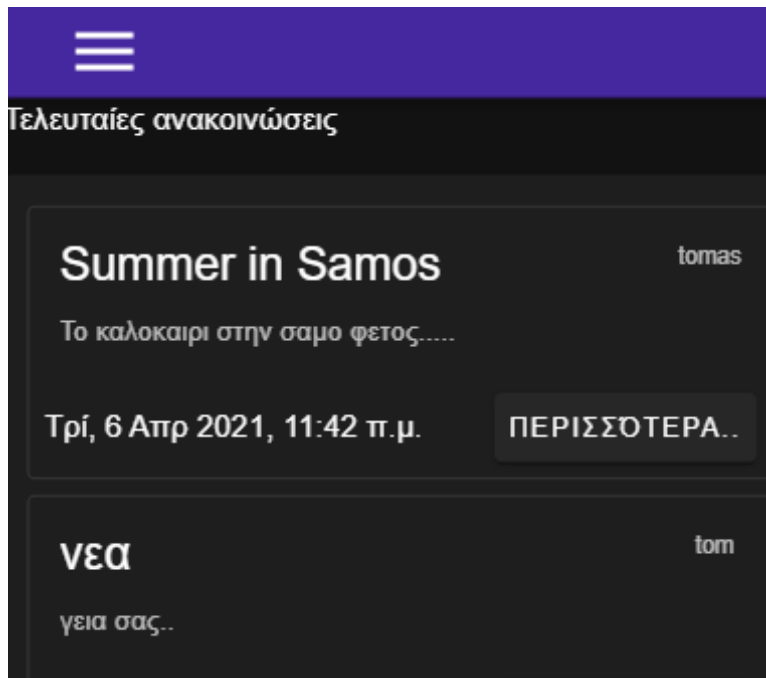
Εικόνα 6.21: Αναζήτηση προβλήματος με βάση την κατηγορία

6.4.3 Ανακοινώσεις

Υπάρχει για όλους πρόσβαση επίσης στην κατηγορία ανακοινώσεις. Μέσα από εκεί έχουμε την επιλογή για την ανάρτηση αλλά και την αναζήτηση. Κάποιες επιχειρήσεις, οι διαχειριστές αλλά και οι τεχνικοί μπορούν να ανεβάσουν μια ανακοίνωση που αφορά το θέμα της υγείας και των απινιδωτών (βλ. εικόνα 22). Για παράδειγμα την αγορά και την τοποθέτηση ενός νέου απινιδωτή ή της επιδιόρθωση ενός ώστε να μπορεί να επαναχρησιμοποιηθεί. Για την καλύτερη ταξινόμηση των ανακοινώσεων όταν δημιουργούνται επιλέγονται κάποιες κατηγορίες που ανήκουν. Έπειτα ο κάθε χρήστης που θα θέλει να αναζητήσει μια ανακοίνωση θα έχει την επιλογή να τις δει όλες με βάση την πιο πρόσφατη χρονολογικά, αλλά μπορεί να κάνει και επιλεγμένη αναζήτηση με βάση τον τίτλο ή την κατηγορία που ανήκει (βλ. εικόνα 23).

The screenshot shows a mobile application interface for posting an announcement. At the top, there is a purple header bar with a white hamburger menu icon. Below this is a blue header bar with the title "Αναρτηση ανακοινωσης" in white. The main content area has a dark background. It starts with the label "Τίτλος ανακοίνωσης" followed by the text "Summer in Samos". A horizontal line separates this from the next section, with a character count "15 / 50" on the right. The next section is labeled "Κείμενο ανακοινωσης" and contains the text "Το καλοκαιρι στην σαμο φετος...". Another horizontal line follows, with a character count "31 / 500" on the right. Below this is the label "Επιλογή κατηγορίας" and a dropdown menu. The dropdown menu has a "Tags" label and shows the selected option "SAMOS, OTHER" with a downward arrow. At the bottom right, there is a blue button with the text "ΑΝΑΡΤΗΣΗ".

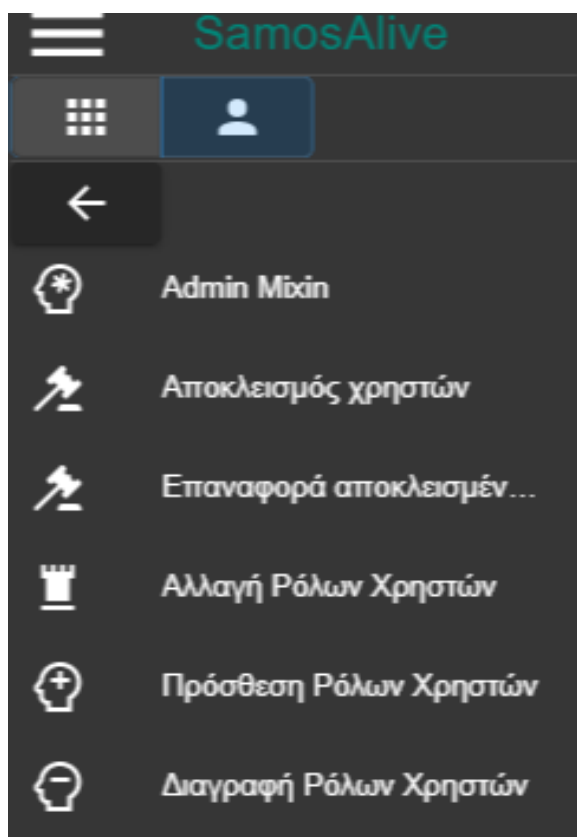
Εικόνα 6.22: Ανάρτηση ανακοίνωσης



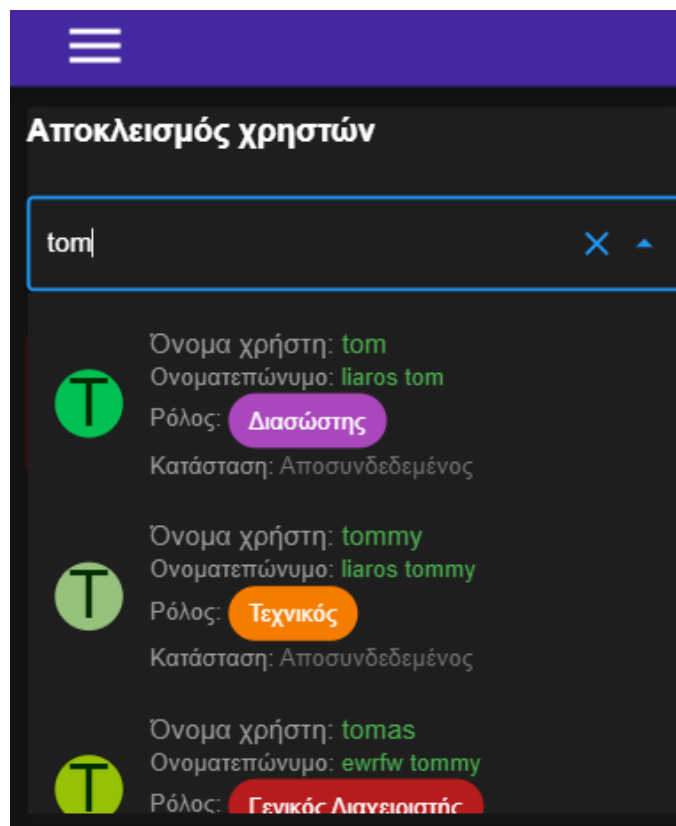
Εικόνα 6.23: Αναζήτηση ανακοινώσεων με επιλεγμένα tag

6.4.4 Σελίδα διαχείρισης

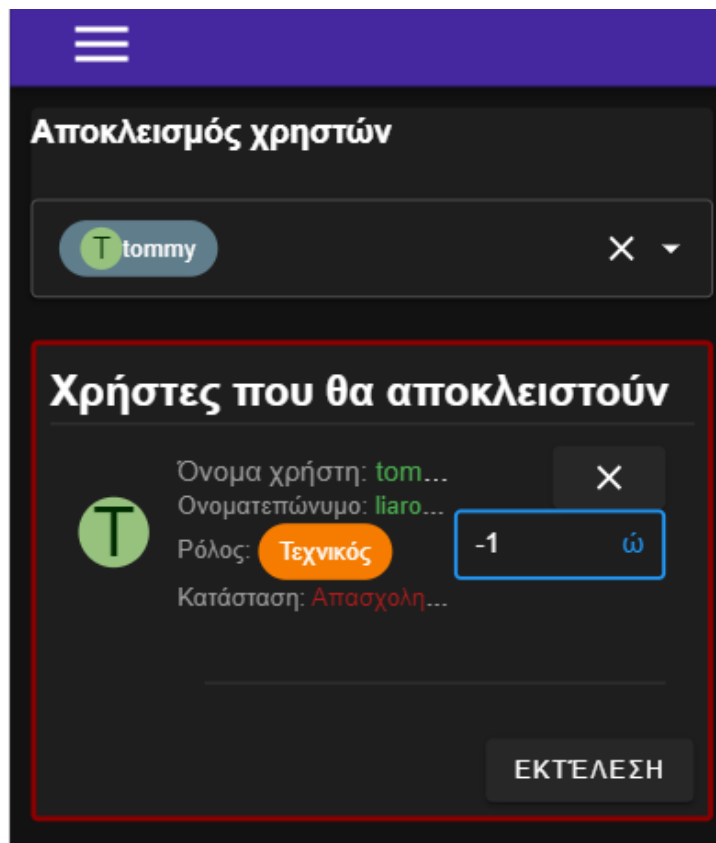
Αποκλειστικά για τους διαχειριστές παρέχετε μια σελίδα διαχείρισης των χρηστών της εφαρμογής. Κάποιος χρήστης που θα φερθεί χωρίς σύνεση και θα πατήσει λανθασμένα ή ψευδώς το κουμπί για έκτακτη ανάγκη ή αναφέρει πρόβλημα σε κάποιον απινιδωτή χωρίς να ισχύει πρέπει να τιμωρηθεί. Έτσι υπάρχει η απαίτηση για αφαίρεση την πρόσβασης του χρήστη από την εφαρμογή. Το χρονικό περιθώριο της επίπληξης θα το κρίνει ο διαχειριστής με βάση το πρόβλημα που προκλήθηκε και ορίζεται από μερικές ώρες περιορισμό μέχρι μόνιμο αποκλεισμό από την εφαρμογή. Παράλληλα πρέπει να γίνεται διαχείριση των ρόλων των χρηστών μέσα από διαφορετικές σελίδες. Αυτό περιέχει τρεις επιλογές που είναι προσθήκη ή διαγραφή ενός ρόλου από τον χρήστη και υπάρχει και η επιλογή να γίνει αλλαγή κάποιου ρόλου για συντομία χρόνου. Επίσης σε αυτές τις σελίδες παρέχεται ένα κουμπί για τον ρόλο και ένα για την εκτέλεση της διαδικασίας ώστε να μην γίνεται για τον κάθε χρήστη ξεχωριστά. Λόγου χάρη αν θέλουμε να προσθέσουμε αρκετούς διασώστες μαζί αρκεί να τους ψάξουμε, να επιλέξουμε τον ρόλο διασώστη και με ένα κλικ να γίνει η αλλαγή. Όλα αυτά υπάρχουν και σε μια συγκεντρωτική καρτέλα όπου όλες αυτές οι επιλογές είναι διαθέσιμες μαζί.



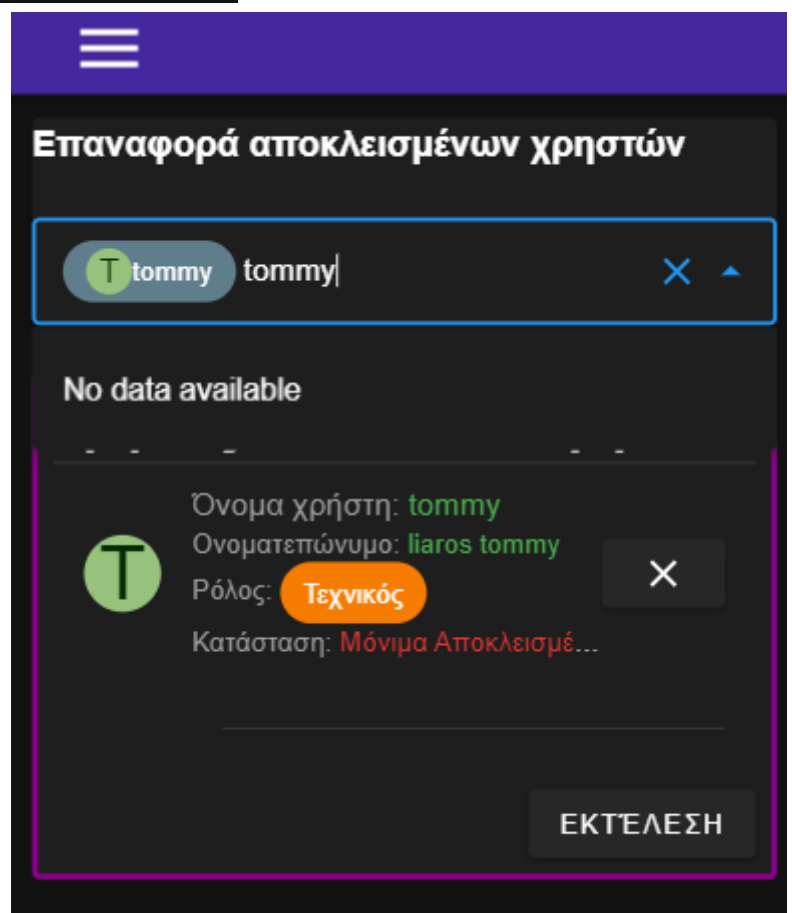
Εικόνα 6.24: Διαχειριστικές επιλογές για ενέργειες πάνω σε χρήστες



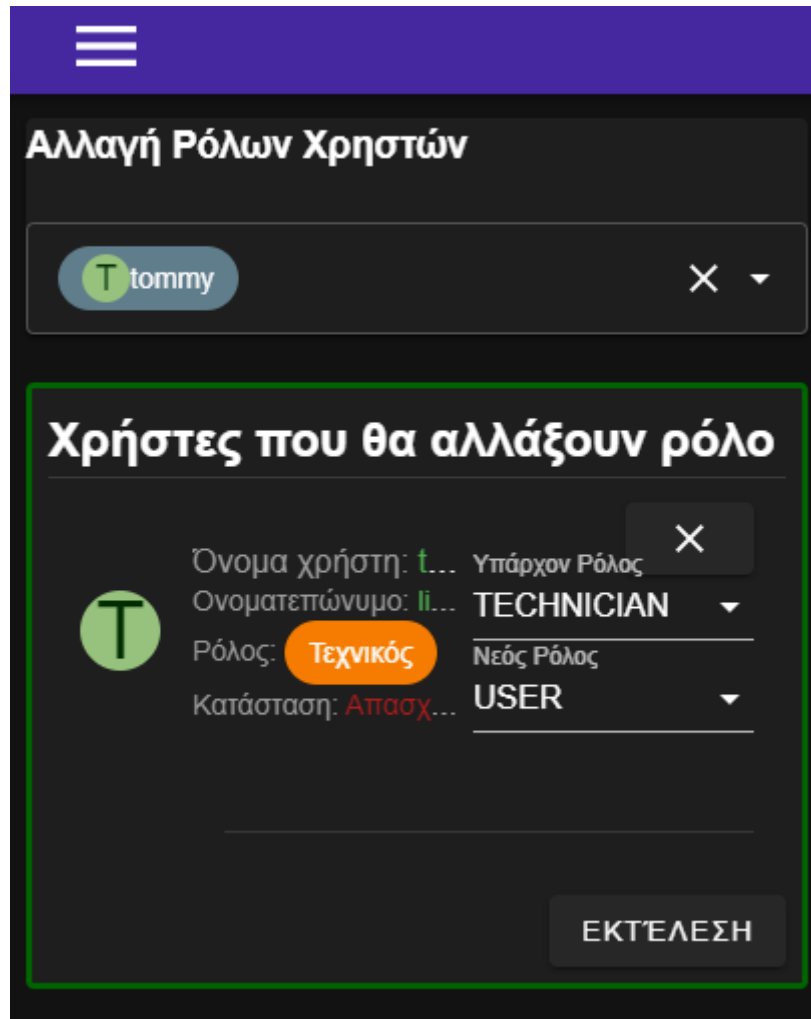
Εικόνα 6.25: Αναζήτηση χρηστών στο διαχειριστικό μενού



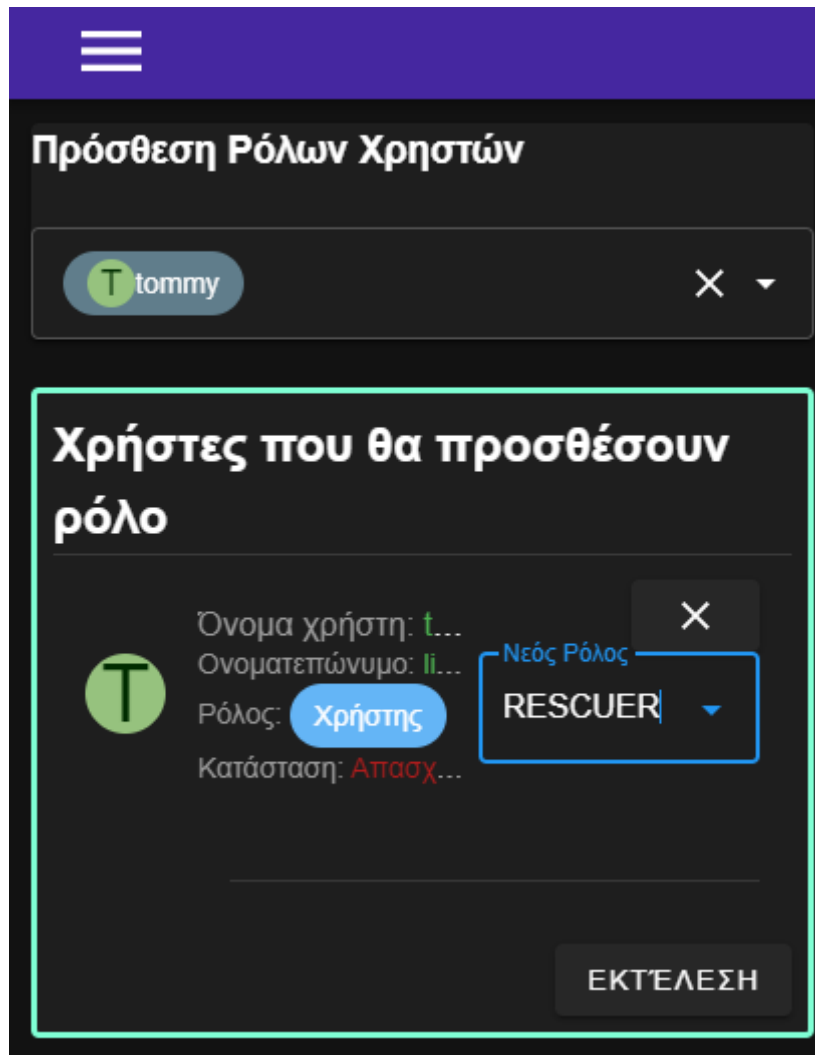
Εικόνα 6.26: Αποκλεισμός χρήστη



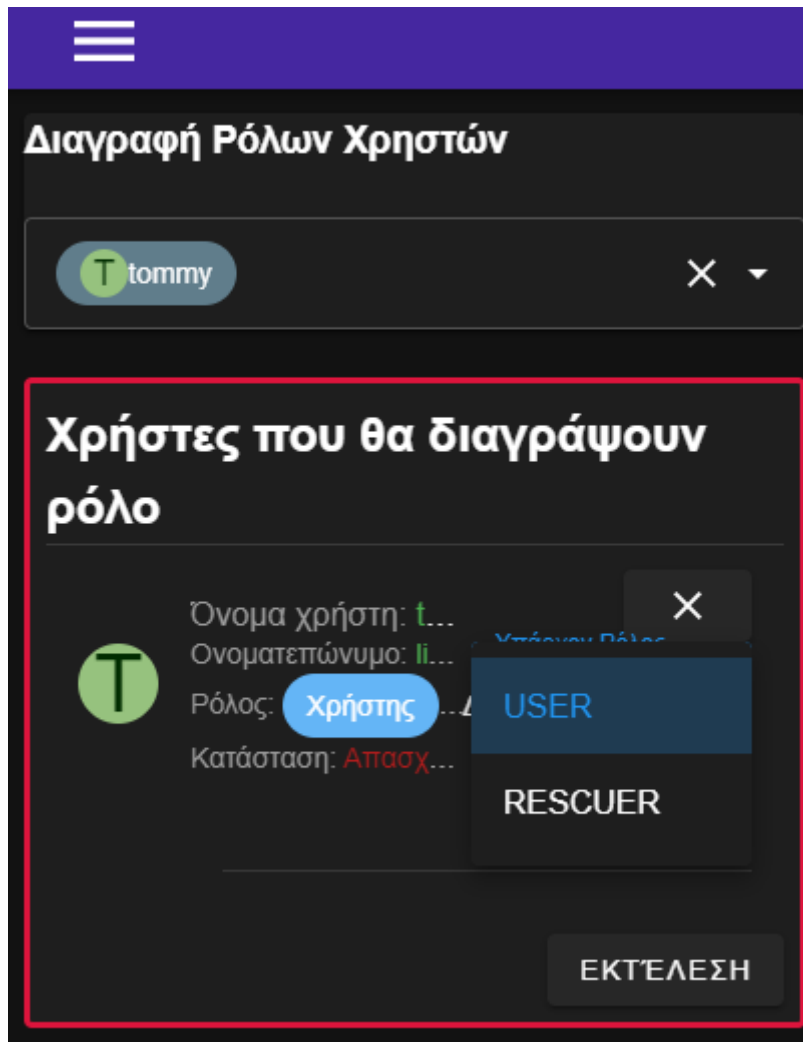
Εικόνα 6.27: Επαναφορά πρόσβασης χρήστη



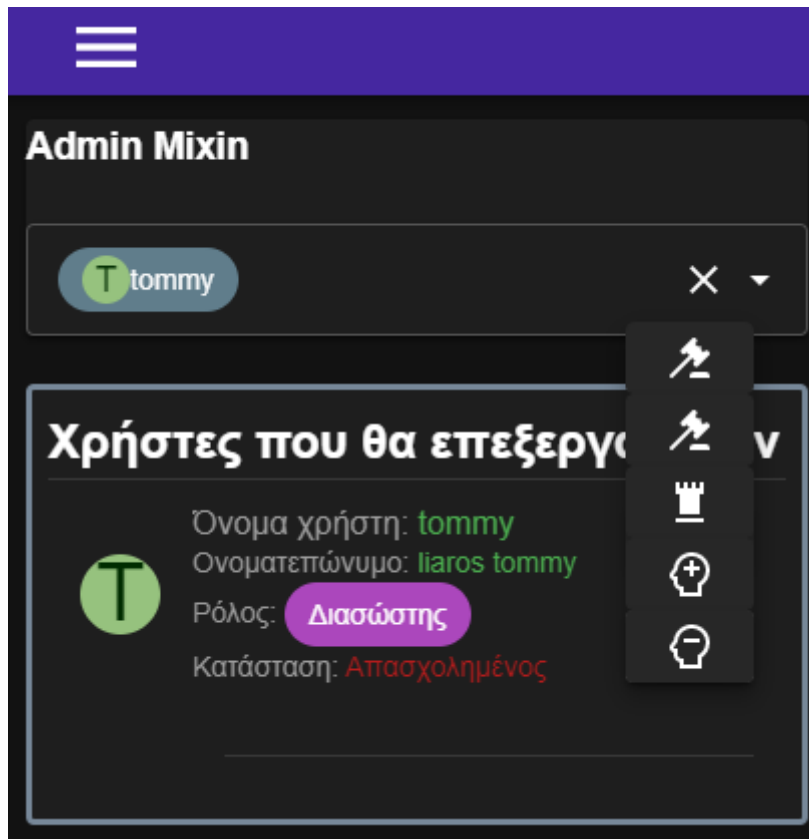
Εικόνα 6.28: Αλλαγή ρόλου σε χρήστη



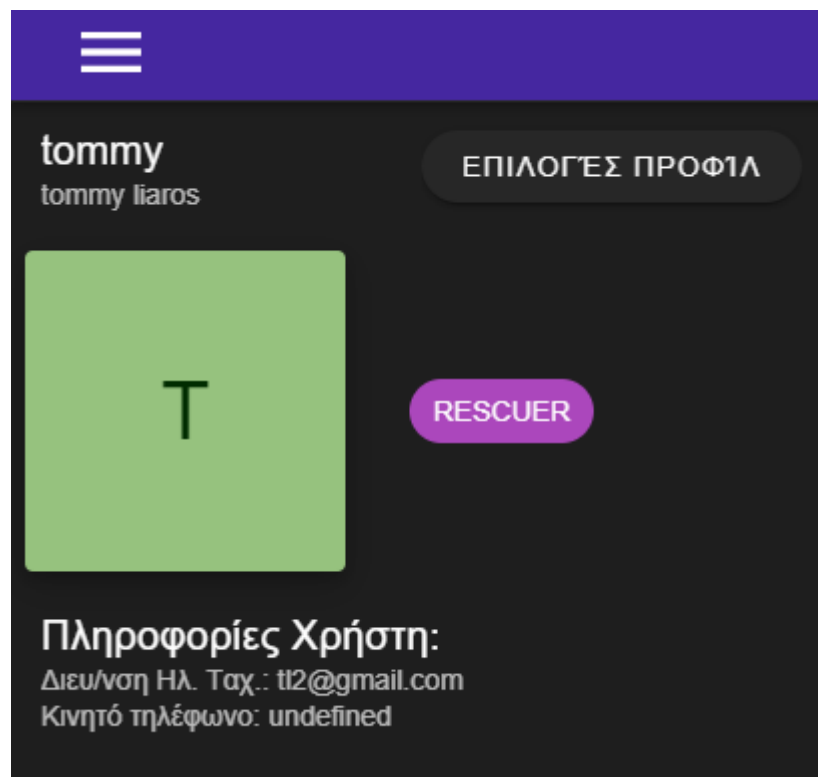
Εικόνα 6.29: Προσθήκη ρόλου σε χρήστη



Εικόνα 6.30: Διαγραφή ρόλου από χρήστη



Εικόνα 6.31: Όλες οι επιλογές μαζεμένες σε μια σελίδα



Εικόνα 6.32: Προφίλ του χρήστη με ορατές τις πληροφορίες που επιτρέπει ο ίδιος

6.4.5 Γενικές χρήσιμες σελίδες

Μια σελίδα ακόμα που θεωρήσαμε χρήσιμη ήταν μια ενιαία σελίδα αναζήτησης. Ο χρήστης επιλέγει αυτό που θέλει να αναζητήσει με βάση κατηγορίες όπως προφίλ, ανακοινώσεις ώστε να υπάρχει μια συγκεντρωτική αναζήτηση και να μην επιλέγει την κάθε αναζήτηση ξεχωριστά.

Κάθε νέος χρήστης μπορεί να αναζητήσει πληροφορίες για την εφαρμογή που κατέβασε και χρησιμοποιεί. Έτσι έχουμε υλοποιήσει μια σελίδα με πληροφορίες όπου αναφέρεται ένα disclaimer που καθιστά την παραπλανητική χρήση της πλατφόρμας νομικά διώξιμη και παρέχονται οδηγίες χρήσης απινιδωτών και γενικά πρώτων αναγκών. Επίσης αν κάποιος χρήστης θέλει να στείλει ένα σχόλιο για την εφαρμογή είτε για κάποια παρατήρηση, κάποιο παράπονο ή κάποια ιδέα βελτίωσης υπάρχει σχετική φόρμα εκεί μέσα. Αφού θα έχει περιηγηθεί στην πλατφόρμα και θα έχει διαβάσει ότι χρειάζεται για την εφαρμογή θα μπορεί να κρίνει αν χρειάζεται να το στείλει.

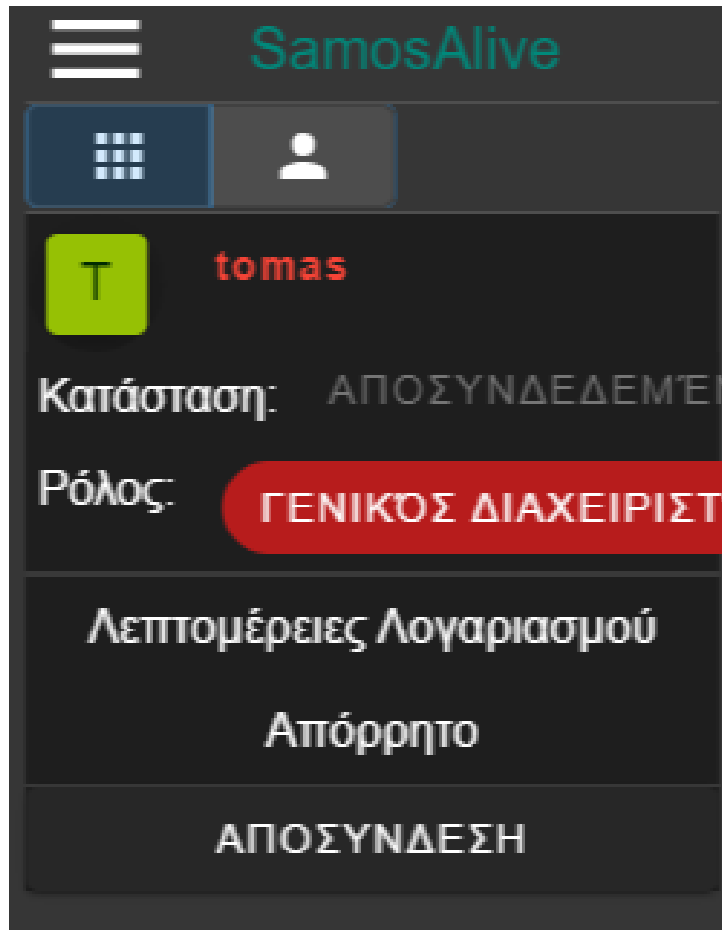
Όπως σε κάθε εφαρμογή παρέχονται επιλογές για διάφορες ρυθμίσεις. Στο πρώιμο στάδιο που βρίσκεται η εφαρμογή υπάρχει μόνο αλλαγή γλώσσας από ελληνικά σε αγγλικά και το αντίστροφο καθώς και επιλογή φόντου μεταξύ φωτεινού και σκοτεινού.

The screenshot shows a mobile application interface with a dark theme. At the top, a header in white text reads: "Γενικές πληροφορίες για την εφαρμογή αλλά για την υγεία γενικά". Below this is a section titled "Ανάρτησε ένα σχόλιο ή μια ιδέα για την εφαρμογή". Under the title, there are two input fields: "Τίτλος" (Title) and "Προσθήκη ..." (Add ...). To the right of the second field is a character count "12 / 50". Below these fields is a label "Κείμενο ανακοίνωσης" (Announcement text). The main text area contains the text "Θα ήθελα να κάνω πρόταση για προσθήκη μπλα μπλα.." with red underlines under "μπλα". Below the text area is another character count "49 / 50". At the bottom right, there is a blue button with the white text "ΑΝΑΡΤΗΣΗ" (Post).

Εικόνα 6.33: Πληροφορίες για εφαρμογή και ανάρτηση σχολίου

6.4.6 Δευτερεύον μενού για τον χρήστη

Τέλος υπάρχει ένα δευτερεύον μενού όπου αλλάζει την λίστα με τα στοιχεία σε νέες επιλογές για ρυθμίσεις προσωπικών δεδομένων χρήστη (βλ. εικόνα 34). Πάνω αριστερά στο μενού υπάρχει η επιλογή να αλλάξει το μενού με κατηγορίες για προσωπικές πληροφορίες του χρήστη όπως λεπτομέρειες λογαριασμού, απόρρητο και προτιμήσεις. Μέσα σε αυτές τις σελίδες ο χρήστης αλλάζει τα στοιχεία του, τον τρόπο που αυτά παρουσιάζονται στους υπόλοιπους καθώς και διάφορες άλλες λειτουργίες περί ιδιωτικότητας



Εικόνα 6.34: Δευτερεύον μενού με επιλογές ιδιωτικότητας

***Εικόνα 6.35:** Επεξεργασία των βασικών πληροφοριών του χρήστη*

Ρυθμίσεις Απορρήτου

Χρήστη: tomas

Κατάσταση Σύνδεσης:

Τύπος Εμφάνισης

Δημόσια

Τελευταία Σύνδεση:

Τελευταία Αποσύνδεση:

Πραγματικό Ονόμα:

Ηλνικο Ταχυδρομείο:

Δημιουργία Λογαριασμού:

Περιγραφή:

Διεύθυνση:

Κινητό Τηλέφωνο:

Εικόνα 6.36: Ρυθμίσεις απορρήτου

7. Συμπεράσματα και Προοπτικές (στην περίπτωση της Μελετητικής είναι το 6^ο Κεφάλαιο)

7.1. Συμπεράσματα επί των αποτελεσμάτων

Η πλατφόρμα SamosAlive απέκτησε νόημα και ενδιαφέρον καθ' όλη της διάρκεια της υλοποίησης της διότι έχει ως σκοπό την βοήθεια και την ευκολία των πολιτών να έχουν πρόσβαση σε ένα κομμάτι υγείας που μέχρι τώρα υστερούσε. Πλέον το σύστημα έχει περατωθεί σε ένα ικανοποιητικό επίπεδο για παραγωγική έκδοση της. Τα εκ των προτέρων σχέδια, εξελίχθηκαν περαιτέρω μέσα από συζητήσεις, αναλύσεις και τροποποιήσεις πάνω σε απαιτήσεις βέλτιστων πρακτικών που ενίσχυαν τόσο λειτουργικά όσο και αρχιτεκτονικά την εφαρμογή, διευκόλυναν την διαχείριση, την συντήρηση της και την επεκτασιμότητα της. Ζήτημα που παραμένει ως περιορισμός είναι η αποτροπή κακοπροαίρετης χρήσης της εφαρμογής στις περιπτώσεις δημιουργίας των γεγονότων, λύση του περιορισμού αυτού θα ήταν υποχρεωτική εγγραφή του κινητού του χρήστη και επαλήθευση του. Κάθε ιδέα για λειτουργικότητα και χρηστικότητα που μπορεί να βοηθήσει την εφαρμογή και την ομαλή χρήση της SamosAlive, μετά από αξιολόγηση θα προστεθεί στο μέλλον. Γι' αυτό τον λόγο δίνουμε και στους χρήστες δυνατότητα αποστολής σχολίων και προτάσεων στους διαχειριστές.

7.2 Συμπεράσματα επί της μεθοδολογίας

Η μεθοδολογία που επιλέξαμε να ακολουθήσουμε έδειχνε άρτια και βοήθησε αισθητά στην παραγωγικότητα της ομάδας μας. Τέθηκαν στάδια που έπρεπε να περάσουν οι ιδέες μέχρι να καταλήξουν σε υπηρεσίες, διαδικασίες ή περιεχόμενο της εφαρμογής και εν τέλει ελέγχους σεναρίων του κώδικα για την εξασφάλιση της ακεραια και απεγάδιαστης λειτουργίας τους. Κατά την διάρκεια της ανάπτυξης του συστήματος ανυψώθηκαν προβλήματα, τα οποία σε μετέπειτα στάδιο λύθηκαν, που αφορούσαν λειτουργικά ερωτήματα για το πεδίο της εφαρμογής καθώς και αρχιτεκτονικά για την επέκτασης της, δεν προβλέφθηκε εξαρχής από την ομάδα μας μια πλήρης εικόνα της εφαρμογής καθώς και επίγνωση όλων των υπάρχοντων ανοιχτού κώδικα εργαλείων που μπορούσαν να συνεισφέρουν.

7.3 Συμπεράσματα επί των εργαλείων

Η ομάδα μας επέλεξε να χρησιμοποιηθούν εργαλεία, βιβλιοθήκες και frameworks τα οποία εξυπηρετούν με μεγάλη επάρκεια τον σκοπό τους, διευκολύνουν τους προγραμματιστές στην παραγωγικότητα τους και στην εξοικονόμηση χρόνου, προσφέρουν καλύτερη χρηστικότητα και λειτουργικότητα και επεκτασιμότητα ακολουθώντας τις καλύτερες πρακτικές στο πεδίο τους. Διεξάχθηκε έρευνα για την αξιολόγηση των χρησιμότερων ανοιχτού κώδικα εργαλείων που χρειάζεται η εφαρμογή SamosAlive. Σε κάποιες βιβλιοθήκες κληθήκαμε να γράψουμε δικό μας κώδικα καθώς υπήρχαν περιορισμοί που αφορούσαν έλλειψη υποστήριξης και πηγών πληροφοριών λόγω νέας τεχνολογίας πχ RSocket. Ερωτηματικό ήταν για μεγάλη οριζόντια επεκτασιμότητα της εφαρμογής η επιλογή του MongoDB ως NoSQL βάση δεδομένων καθώς με όλα τα θετικά της, έχει ως καταναμεμημένο σύστημα master-slave προσέγγιση, που σημαίνει ένας master χρησιμοποιείται για τα writes και οι slaves για

τα reads, το πρόβλημα που ανέρχεται είναι όταν χρησιμοποιείται γιγαντιαίο throughput στο master για τα write της εφαρμογής, τότε θα υπάρξουν διάφορες καθυστερήσεις, ενδεχομένως η Cassandra θα έλυνε αυτό το πρόβλημα. Εργαλεία που αφορούσαν την επεκτασιμότητα της εφαρμογής σε cloud υποδομή στην πορεία αντικαταστάθηκαν με δυνατότητες container orchestration, όπως αυτό του Docker σε Kubernetes.

7.4 Προοπτικές/ Επόμενα Βήματα

Η εφαρμογή μας έχει δομηθεί με τρόπο που είναι όπως εξηγήσαμε εύκολα διαχωρίσιμος, σπασμένος σε μικρά κομμάτια κώδικα για κάθε λειτουργία και εμφάνιση μέσα στην πλατφόρμα και αυτά τα στοιχεία παρέχουν μεγάλη επεκτασιμότητα. Αυτό μας βοηθάει διότι άμα κάτι πρέπει να αλλάξει, να σβηστεί ή να προστεθεί γίνεται με τέτοιο τρόπο που δεν επηρεάζει τα υπόλοιπα στοιχεία του κώδικα. Εκτός από αυτό μπορεί κάποιες ιδέες πολιτών ή κάποια άλλη εφαρμογή ολόκληρη να ενωθεί στην δικιά μας αν το απαιτήσουν οι συνθήκες με ελάχιστες αλλαγές. Επίσης αν μεγαλώσει η ομάδα προγραμματιστών μας θα μπορούν εύκολα να καταλάβουν τα κομμάτια κώδικα και να εργαστούν πάνω σε κάτι δικό τους χωρίς να επηρεάζουν τα ήδη υπάρχοντα δικά μας. Αυτό μας δίνει το περιθώριο πέρα από την διπλωματική εργασία να δουλέψουμε παραπάνω στην εφαρμογή και να βοηθήσουμε όσο μπορούμε με αυτή τον κόσμο κάνοντας τον καλύτερο με τον δικό μας τρόπο.

Επίσης είχαμε στο μυαλό μας να επεκτείνουμε το αντικείμενο της εφαρμογής πέρα από την υγεία. Σκεπτόμενοι το κοινό καλό έχουμε δημιουργήσει έτσι την πλατφόρμα που μπορεί ο κορμός της να μείνει ο ίδιος όμως να ειδικεύεται και να βοηθάει σε κάποιον άλλο τομέα, όπως για παράδειγμα την πυροσβεστική, την αστυνομία ή τα μέσα μαζικής μεταφοράς.

8. Αναφορές

1. <https://expressjs.com/>
2. <https://en.wikipedia.org/wiki/MongoDB>
3. <https://redis.io/topics/introduction>
4. <https://spring.io/projects/spring-cloud-gateway>
5. <https://router.vuejs.org>
6. <https://vuex.vuejs.org/>

9. Παράρτημα

Παράρτημα Α

JWT tokens: Το JWT token είναι ένα κρυπτογραφημένο αλφαριθμητικό που δημιουργείται από το server συγκεντρώνοντας τις διάφορες πληροφορίες (δομής Map, key=value) που αφορούν μια οντότητα(πχ. Χρήστης => userid, username, roles), ο server δίνει και ημερομηνία λήξης για το JWT και οι πληροφορίες αυτές κρυπτογραφούνται με βάση ενός κρυφού κλειδιού που γνωρίζει μόνο ο server. Το JWT αποστέλλεται και παραμένει στην συσκευή του χρήστη. Όταν ο χρήστης θέλει να εξυπηρετηθεί από μια υπηρεσία στέλνει το JWT token για να επαληθευτεί. Αποκρυπτογραφείται στον server το JWT με το κρυφό κλειδί και επαληθεύονται οι πληροφορίες του χρήστη και γίνεται έλεγχος εξουσιοδότησης για την εκτέλεση της υπηρεσίας.

Access Token: Το access token είναι ένα JWT token που αποθηκεύεται σε μεταβλητή της συσκευής του χρήστη και χρησιμοποιείται για την επαλήθευση και τον έλεγχο εξουσιοδότησης για πρόσβαση σε υπηρεσία στον server. Έχει πολύ μικρή διάρκεια ζωής, μόλις 20 λεπτά από την υπογραφή του, μετά το deadline του το access token θεωρείται μη έγκυρο.

Refresh Token: Το refresh token είναι ένα JWT token που αποθηκεύεται σε http-only cookie στην συσκευή του χρήστη, οπότε είναι μη προσβάσιμο από JavaScript κώδικα. Έχει πολύ μεγάλη διάρκεια ζωής και χρησιμοποιείται κάθε 20 λεπτά αυτόματα από την εφαρμογή του χρήστη με αποκλειστικό σκοπό να παραχθεί νέο access token για τον χρήστη.

Παράρτημα Β

Παρακάτω επισυνάπτονται τα link του κώδικα στο github.

Back-end: <https://github.com/Droukos/samosalive-cloud>

Front-end: <https://github.com/Droukos/SamosAliveClient>