



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

**Ανάπτυξη και αξιολόγηση εργαλείου για τη δημιουργία IoT
εφαρμογών με τη συμμετοχή τελικών χρηστών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

**Αλεξάκη Φίλιππου – Αποστόλου του Αιμιλίου
Παπαδόπουλου Αναστασίου του Ιωάννη**

Επιβλέπων : Γκουμόπουλος Χρήστος

Αναπληρωτής Καθηγητής του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών
Συστημάτων του Πανεπιστημίου Αιγαίου

Μέλη εξεταστικής επιτροπής:

Βουγιούκας Δημοσθένης

Καθηγητής του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του
Πανεπιστημίου Αιγαίου,

Μεσαριτάκης Χάρης

Αναπληρωτής Καθηγητής του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών
Συστημάτων του Πανεπιστημίου Αιγαίου

Σάμος, Οκτώβριος 2020

Πρόλογος και ευχαριστίες

Αρχικά, θα θέλαμε να ευχαριστήσουμε τον καθηγητή μας κ. Χρήστο Γκουμόπουλο που μας επέλεξε για την ανάπτυξη της συγκεκριμένης διπλωματικής εργασίας. Η παρουσία και η καθοδήγηση που μας παρείχε ήταν καθοριστική καθώς ήταν δίπλα μας σε όλα τα στάδια της. Η ολοκλήρωση της έρευνας είναι αποτέλεσμα της εξαιρετικής συνεργασίας και επικοινωνίας που δημιουργήθηκε μεταξύ μας. Οι δύσκολες συνθήκες που δημιουργήθηκαν λόγω της πανδημίας δεν επέτρεψαν την αυτοπροσώπως επικοινωνία. Το συγκεκριμένο εμπόδιο δεν επηρέασε την συνεργασία μας και την εκπόνηση διπλωματικής καθώς υπήρξε άμεση ανταπόκριση σε ότι δυσκολίες προέκυπταν. Τον ευχαριστούμε θερμά.

Επιπροσθέτως, ευχαριστούμε τις οικογένειες και τους φίλους μας καθώς αμφότεροι συνέβαλαν με τον τρόπο τους για την ολοκλήρωση των σπουδών και για την ευχάριστη διαμονή στη Σάμο.

Τέλος, θα θέλαμε να ευχαριστήσουμε τους συμμετέχοντες που έλαβαν μέρος στις δοκιμές που πραγματοποιήθηκαν με στόχο την αξιολόγηση και βελτίωση της εφαρμογής κινητού υπολογισμού μας και γενικότερα της διπλωματικής εργασίας.

© 2020

των

Αλεξάκη Φίλιππου – Απόστολου

Παπαδόπουλου Αναστασίου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πίνακας περιεχομένων

1	Εισαγωγή	1
1.1	Αντικείμενο διπλωματικής.....	1
1.2	Διάρθρωση διπλωματικής.....	2
2	Μελέτη Υφιστάμενης Κατάστασης.....	3
2.1	Εργαλεία για δημιουργία IoT εφαρμογών	3
2.1.1	Tasker	3
2.1.2	IFTTT.....	4
2.1.3	Atooma.....	5
2.1.4	AutomateIt.....	6
2.1.5	Bip.io.....	6
2.1.6	Elastic.io	6
2.1.7	itDuzzit.....	7
2.1.8	Node-RED.....	8
2.1.9	WigWag.....	8
2.1.10	Zapier.....	8
3	Ανάλυση Απαιτήσεων.....	10
3.1	Εισαγωγή.....	10
3.2	Λειτουργικές Απαιτήσεις	10
3.3	Μη λειτουργικές Απαιτήσεις	11
4	Σχεδιασμός Συστήματος.....	13
4.1	Προσέγγιση.....	13
4.2	Δημιουργία REST API και Βάσης Δεδομένων.....	13
4.2.1	Βάση Δεδομένων	14
4.2.2	Ασφάλεια Συστήματος	16
4.2.3	Κεντρικός Server	18
4.2.4	Αυθεντικοποίηση Χρηστών	21
4.2.5	Επικοινωνία με Έξυπνες Συσκευές.....	22
4.3	Εφαρμογή Android	24
4.3.1	Εισαγωγή.....	24
4.3.2	Editor Domino	26
4.3.3	Editor E - Wizard.....	37

4.3.4	Κάτοψη (Floorspan)	42
4.3.5	Αποθηκευμένοι Κανόνες.....	44
4.3.6	Εκπαίδευση στους Editors	44
4.3.7	Πληροφορίες Έξυπνων Συσκευών.....	46
4.4	Σχεδιασμός και Λειτουργίες Διεπαφής	51
4.4.1	Εισαγωγή.....	51
4.4.2	Εμπειρία Χρήσης Εφαρμογής	51
4.4.3	Χρήση Dominos ως metaphor για την απεικόνιση των έξυπνων συσκευών στον Editor Domino.....	52
4.4.4	Χρήση Icons ως metaphor για την απεικόνιση των έξυπνων συσκευών στον Editor E – Wizard.....	53
4.4.5	Μενού έξυπνων συσκευών στους δύο Editors	54
4.4.6	Χρώματα.....	54
4.4.7	Pop – Up Παράθυρα	55
4.4.8	Αναγραφή Ενεργειών Έξυπνων Συσκευών	55
4.4.9	Αναγραφή Αποθηκευμένου Κανόνα.....	56
4.4.10	Animation.....	57
4.4.11	Κάτοψη (Floorspan)	58
4.4.12	Emoji.....	58
4.5	Αναπαράσταση Δεδομένων σε Τρισδιάστατο Χώρο	59
4.5.1	Εισαγωγή.....	59
4.5.2	Περιβάλλον	59
4.5.3	Μοντελοποίηση τρισδιάστατων Domino και Πλέγματα Αντικειμένων.....	60
4.5.4	Δημιουργία του πρώτου τρισδιάστατου Domino	60
4.5.5	Δημιουργία Material για χρήση στο 2D Plane	61
4.5.6	Δημιουργία UV Σφαιρών για την χρήση αριθμητικών μεταβλητών στο κάτω μέρος του Domino.....	61
4.5.7	Δημιουργία διαχωριστικής γραμμής.....	62
4.5.8	Domino Group	62
4.5.9	Δημιουργία Animation.....	62
4.5.10	Τοποθέτηση κάμερας και φωτισμού.....	62
4.5.11	Γωνίες Λήψης.....	62
4.5.12	Μήνυμα Επιτυχίας.....	63
4.5.13	Τελική Μορφή Animation.....	63
5	Υλοποίηση Εφαρμογής Android	64
5.1	Εισαγωγή.....	64
5.2	Δομή Πακέτων	65
5.2.1	Javadoc	68

5.3	Editor Domino	69
5.3.1	Εισαγωγή.....	69
5.3.2	Προσθήκη Domino στο Grid.....	69
5.3.3	Διαγραφή Πρόσφατου Domino.....	71
5.3.4	Ανανέωση Ενεργειών Έξυπνων Συσκευών.....	72
5.3.5	Προσδιορισμός Λογικής Συσχέτισης.....	73
5.3.6	Αποθήκευση Κανόνα	73
5.3.7	Επανεκκίνηση Editor.....	73
5.4	Εκπαίδευση στους Editors	74
5.4.1	Δημιουργία Activity για χρήση του Προτύπου.....	75
5.5	Κάτοψη (Floorspan).....	76
5.5.1	Drag Activity.....	76
5.5.2	Floorspan Activity και Android Permissions.....	76
5.6	Editor E – Wizard	78
5.6.1	Εισαγωγή.....	78
5.6.2	Βασικές Μεταβλητές για την λειτουργία του E – Wizard	78
5.6.3	Άνοιγμα/Κλείσιμο του μενού έξυπνων συσκευών.....	79
5.6.4	Τοποθέτηση έξυπνης συσκευής στο μέρος του συμβάντος του κανόνα.....	80
5.6.5	Τοποθέτηση έξυπνης συσκευής στο μέρος της δράσης του κανόνα	82
5.6.6	Διαγραφή/Ανανέωση έξυπνης συσκευής	84
5.6.7	Προσθήκη λογικών τελεστών OR και AND στον κανόνα	86
5.6.8	Αποθήκευση/Επανεκκίνηση Editor	86
5.7	Αποθηκευμένοι Κανόνες	88
5.8	Πληροφορίες Έξυπνων Συσκευών.....	90
5.9	Κώδικας Αναγραφής κανόνα σε φυσική γλώσσα	91
5.10	Κώδικας Animation.....	91
5.11	Επαναχρησιμοποιήσιμα τμήματα κώδικα	92
5.12	Εξωτερικές Βιβλιοθήκες	92
6	Έλεγχος Συστήματος	94
6.1	Εισαγωγή.....	94
6.2	Σενάρια Δοκιμών	95
6.2.1	KANONAS 1	95
6.2.2	KANONAS 2	95
6.2.3	KANONAS 3	96
6.2.4	KANONAS 4	96

6.2.5	<i>ΚΑΝΟΝΑΣ 5</i>	97
6.3	Δοκιμή Χρηστών.....	97
6.3.1	<i>Πρώτη Δοκιμή</i>	97
6.3.2	<i>Δεύτερη Δοκιμή</i>	98
6.4	Αξιολόγηση.....	104
6.5	Ικανοποίηση Απαιτήσεων	107
6.5.1	<i>Ικανοποίηση λειτουργικών απαιτήσεων</i>	107
6.5.2	<i>Ικανοποίηση μη λειτουργικών απαιτήσεων</i>	108
7	<i>Αδυναμίες και Βελτίωση Συστήματος</i>	109
7.1	Αδυναμίες	109
7.1.1	<i>Εισαγωγή</i>	109
7.1.2	<i>Στατικό περιβάλλον τοποθέτησης έξυπνων συσκευών στο μέρος των συμβάντων στον Editor Domino</i> 109	
7.1.3	<i>Ορισμός μίας δράσης στον Editor Domino και διαγραφής της ήδη ορισμένης δράσης</i>	110
7.1.4	<i>Ανανέωση, επεξεργασία και μετονομασία αποθηκευμένου κανόνα</i>	110
7.2	Βελτίωση.....	110
7.2.1	<i>Editor Domino</i>	110
7.2.2	<i>Editor E – Wizard</i>	111
7.2.3	<i>Αποθηκευμένοι Κανόνες</i>	111
7.2.4	<i>Κάτοψη (Floorspan)</i>	112
7.2.5	<i>Animation</i>	112
7.2.6	<i>Διαχωρισμός έξυπνων συσκευών</i>	112
7.2.7	<i>Εκπαίδευση στους Editors</i>	113
7.2.8	<i>Επικοινωνία πραγματικού χρόνου εφαρμογής με έξυπνες συσκευές</i>	113
8	<i>Συμπεράσματα</i>	114
8.1	Πλατφόρμα IoT.....	114
8.2	Ανάδραση Χρηστών	115
8.3	Το Μέλλον των Έξυπνων Κατοικιών	115
	Βιβλιογραφία	116
	Παράρτημα I – System Usability Scale	118

Λίστα Σχημάτων

Εικόνα 1. Εφαρμογή "Tasker".....	4
Εικόνα 2. Δημιουργία κανόνα στο "IFTTT".	5
Εικόνα 3. Δημιουργία κανόνα στο "Atooma".	6
Εικόνα 4. Δημιουργία trigger στο "AutomateIt".	6
Εικόνα 5. Δημιουργία action στο "AutomateIt".	6
Εικόνα 6. Δημιουργία κανόνα στο "Elastic.io".	7
Εικόνα 7. Δημιουργία κανόνα στο "itDuzzit".	7
Εικόνα 8. Δημιουργία κανόνα στο "Node-RED".	8
Εικόνα 9. Δημιουργία κανόνα στο "Zapier".	9
Εικόνα 10. Συσχετίσεις βάσης δεδομένων.	14
Εικόνα 11. Πρωτόκολλο Επικοινωνίας HTTP/1.1.	16
Εικόνα 12. Πρωτόκολλο Επικοινωνίας HTTP/1.2.	17
Εικόνα 13. Έξυπνες συσκευές απεικονισμένες ως Dominos.	26
Εικόνα 14. Διαχωρισμός συμβάντων - δράσης.	27
Εικόνα 15. Διαχωρισμός έξυπνων συσκευών με βάση την θέση που βρίσκονται στην περιοχή την οποία βρίσκονται.	27
Εικόνα 16. Δημιουργία λογικής σχέσης OR.....	28
Εικόνα 17. Δημιουργία λογικής σχέσης AND.	28
Εικόνα 18. Δημιουργία λογικής σχέσης OR.....	28
Εικόνα 19. Δημιουργία λογικής σχέσης AND.	29
Εικόνα 20. Βοηθητικό domino "shadow" για ορισμό δράσης.....	30
Εικόνα 21. Εμφάνιση δυνατών ρυθμίσεων της έξυπνης συσκευής μέσω pop-up μηνύματος.	30
Εικόνα 22. Εμφάνιση επιλεγμένων ρυθμίσεων της έξυπνης συσκευής στα αριστερά.	30
Εικόνα 23. Εναρκτήριο βοηθητικό domino “shadow” στα συμβάντα.	31
Εικόνα 24. Εμφάνιση Pop Up μηνύματος για ρύθμιση της συσκευής.	31
Εικόνα 25. Εμφάνισης της επιλεγμένης έξυπνης συσκευής και της ρύθμισης που εισήγαγε ο χρήστης στο κάτω μέρος.	31
Εικόνα 26. Εμφάνιση δύο βοηθητικών dominos "shadow" για την προσθήκη του επόμενου domino.	32
Εικόνα 27. Κανόνας έπειτα από αρκετές προσθήκες έξυπνων συσκευών.	32
Εικόνα 28. Εμφάνιση μηνύματος σε περίπτωση που ο χρήστης δεν έχει ορίσει δράση.	33
Εικόνα 29. Εμφάνιση μηνύματος σε περίπτωση που ο χρήστης δεν έχει ορίσει, τουλάχιστον, ένα συμβάν.	33

Εικόνα 30. Κουμπί το οποίο πατώντας το θα μας οδηγήσει στις επιλογές :1) Αποθήκευση , 2) Επανεκκίνηση.....	34
Εικόνα 31. Εμφάνιση των επιλογών : 1)Αποθήκευση, 2)Ανανέωση.	34
Εικόνα 32. Επιλογή Αποθήκευσης και πληκτρολόγηση ονόματος του κανόνα.....	34
Εικόνα 33. Εμφάνιση του κανόνα στην λίστα με τους αποθηκευμένους κανόνες.	35
Εικόνα 34. Εμφάνιση του κανόνα σε απλό κείμενο και σε κινούμενο σχέδιο εφ' όσον πατήσει το κουμπί με όνομα "Animation".	35
Εικόνα 35. Επανεκκίνηση Editor Domino.	35
Εικόνα 36. Προηγούμενες ρυθμίσεις του έξυπνου ρολογιού.	36
Εικόνα 37. Αλλαγή της ρύθμισης από 21:00 σε 20:00.....	36
Εικόνα 38. Εμφάνισης της νέας λεζάντας έπειτα από την ενημέρωση που δέχθηκε το έξυπνο ρολόι.....	36
Εικόνα 39. Διαγραφή του domino που αναπαριστάται το "Phone".....	37
Εικόνα 40. Εμφάνιση pop-up μηνύματος για ολοκλήρωση της διαγραφής.	37
Εικόνα 41. Ο κανόνας της Εικόνας 26 μετά την διαγραφή του domino "Phone".	37
Εικόνα 42. Έξυπνες συσκευές απεικονισμένες ως εικονίδια.	37
Εικόνα 43. Διαχωρισμός συμβάντων – δράσεων Editor E - Wizard.....	38
Εικόνα 44. Εμφάνιση του κουμπιού που αφορά το λογικό OR και AND.....	39
Εικόνα 45. Εμφάνιση λογικού OR και AND για επιλογή λογικής σχέσης και για προσθήκη έξυπνης συσκευής.	39
Εικόνα 46. Προσθήκη έξυπνου κλιματιστικού στη δράση.....	40
Εικόνα 47. Εισαγωγή πρώτης έξυπνης συσκευής στο μέρος των συμβάντων.	40
Εικόνα 48. Προσθήκη 2ης έξυπνης συσκευής στα συμβάντα. Οι δύο έξυπνες συσκευές συσχετίζονται μεταξύ τους με το λογικό OR.	41
Εικόνα 49. Εμφάνιση pop-up μηνύματος για Διαγραφή ή Ανανέωση.....	42
Εικόνα 50. Περιβάλλον Κάτοψης (Floorspan).....	42
Εικόνα 51. Προσθήκη εικονιδίων έξυπνων συσκευών στην μεταφορτωμένη εικόνα κάτοψης.	43
Εικόνα 52. Λίστα αποθηκευμένων κανόνων.	44
Εικόνα 53. Πληροφορίες Έξυπνων Συσκευών στο μενού της εφαρμογής.....	47
Εικόνα 54. Πληροφορίες Έξυπνου Ρολογιού και Τηλεόρασης.....	48
Εικόνα 55. Πληροφορίες Έξυπνου Θερμομέτρου, Καφετιέρας και Ψυγείου.	49
Εικόνα 56. Πληροφορίες Έξυπνου Φούρνου και Κλιματιστικού.	50
Εικόνα 57. Πληροφορίες Έξυπνου Κινητού Τηλεφώνου και Φορητών Συσκευών.	51
Εικόνα 58. Domino Έξυπνο Ρολόι.	Error! Bookmark not defined.
Εικόνα 59. Domino Έξυπνης Καφετιέρας.....	Error! Bookmark not defined.
Εικόνα 60. Domino Έξυπνο Ψυγείου.....	Error! Bookmark not defined.
Εικόνα 61. Domino Έξυπνο Κλιματιστικό.....	Error! Bookmark not defined.

Εικόνα 62. Domino Έξυπνο Θερμόμετρο.	Error! Bookmark not defined.
Εικόνα 63. Domino Έξυπνων Φορητών Συσκευών.	Error! Bookmark not defined.
Εικόνα 64. Domino Έξυπνης Τηλεόρασης.....	Error! Bookmark not defined.
Εικόνα 65. Domino Έξυπνου Φούρνου.....	Error! Bookmark not defined.
Εικόνα 66. Domino Έξυπνου Κινητού Τηλεφώνου.	Error! Bookmark not defined.
Εικόνα 67. Domino Έξυπνο Ρολόι.	52
Εικόνα 68. Domino Έξυπνης Καφετιέρας.....	52
Εικόνα 69. Domino Έξυπνο Ψυγείου.....	52
Εικόνα 70. Domino Έξυπνο Κλιματιστικό.....	52
Εικόνα 71. Domino Έξυπνο Θερμόμετρο.	52
Εικόνα 72. Domino Έξυπνων Φορητών Συσκευών.	52
Εικόνα 73. Domino Έξυπνης Τηλεόρασης.....	53
Εικόνα 74. Domino Έξυπνου Φούρνου.....	53
Εικόνα 75. Domino Έξυπνου Κινητού Τηλεφώνου.	53
Εικόνα 76. Icon Έξυπνου Ρολογιού.	53
Εικόνα 77. Icon Έξυπνης Τηλεόρασης.....	53
Εικόνα 78. Icon Έξυπνου Θερμομέτρου.	53
Εικόνα 79. Icon Έξυπνης Καφετιέρας.....	53
Εικόνα 80. Icon Έξυπνου Ψυγείου.....	53
Εικόνα 81. Icon Έξυπνου Φούρνου.....	53
Εικόνα 82. Icon Έξυπνου Κινητού Τηλεφώνου	53
Εικόνα 83. Icon Έξυπνων Φορητών Συσκευών	53
Εικόνα 84. Οριζόντια κύλιση μενού στον Editor Domino.	54
Εικόνα 85. Οριζόντια κύλιση μενού στον Editor E-Wizard.....	54
Εικόνα 86. Αποχρώσεις κόκκινου χρώματος.	54
Εικόνα 87. Χρήση Gradient Color στην Εκπαίδευση των Editors.	55
Εικόνα 88. Παράδειγμα Pop – Up Παραθύρου.	55
Εικόνα 89. Αναγραφή ενέργειας έξυπνης συσκευής στον Editor Domino.	56
Εικόνα 90. Αναγραφή ενέργειας έξυπνης συσκευής στον Editor E-Wizard.....	56
Εικόνα 91. Αναγραφή αποθηκευμένου κανόνα που δημιουργήθηκε με τον Editor Domino.	57
Εικόνα 92. Αναγραφή αποθηκευμένου κανόνα που δημιουργήθηκε στον Editor E – Wizard.	57
Εικόνα 93. Animation αποθηκευμένου κανόνα.	57
Εικόνα 94. Επιλογή του Έξυπνου Θερμομέτρου για προσθήκη στην εικόνα της κάτοψης.	58
Εικόνα 95. Τωρινή εικόνα της κάτοψης μετά την προσθήκη του Έξυπνου Θερμομέτρου.....	58
Εικόνα 96. Emojis στην αναγραφή της δράσης στον Editor Domino.	59
Εικόνα 97. Emojis στην αναγραφή της δράσης στον Editor E-Wizard.....	59

Εικόνα 98. Emojis στις <<Πληροφορίες Έξυπνων Συσκευών>>.	59
Εικόνα 99. Διαφορά UV Σφαίρα (Sphere) και Σφαίρα (Sphere).	60
Εικόνα 100. RoolTile Material.	61
Εικόνα 101. Προβολή κανόνα σε animation μέσω της εφαρμογής.	63
Εικόνα 102. Πακέτο “Entities”.	65
Εικόνα 103. Πακέτο “PopUps”.	65
Εικόνα 104. Πακέτο “Utilities”.	66
Εικόνα 105. Πακέτο “intro”.	66
Εικόνα 106. Πακέτο “Activities”	67
Εικόνα 107. Όλες οι κλάσεις και δραστηριότητες της εφαρμογής.	68
Εικόνα 108. Όλα τα πακέτα της εφαρμογής.	68
Εικόνα 109. XML Κώδικας Grid Layout.	69
Εικόνα 110. Έξυπνες συσκευές ως ImageViews.	70
Εικόνα 111. Βασικές μεταβλητές Editor Domino.	70
Εικόνα 112. Listener αντικειμένων τύπου ImageView.	70
Εικόνα 113. Κώδικας διαγραφής domino.	71
Εικόνα 114. Υπογραφή μεθόδου διαγραφής domino.	71
Εικόνα 115. Pop - up παράθυρο συμβάντος έξυπνου φούρνου.	72
Εικόνα 116. Pop - up παράθυρο δράσης έξυπνου φούρνου.	72
Εικόνα 117. Κώδικας επανεκκίνησης Editor Domino.	74
Εικόνα 118. Κώδικας δημιουργίας προτύπου.	74
Εικόνα 119. Κώδικας για τη χρήση του προτύπου.	75
Εικόνα 120. Οπτικό αποτέλεσμα κώδικα "Εκπαίδευση στους Editors"	75
Εικόνα 121. Πρώτο μέρος κώδικα "Drag Activity".	76
Εικόνα 122. Δεύτερο μέρος κώδικα "Drag Activity".	76
Εικόνα 123. Δήλωση άδειας στο Android Manifest.	77
Εικόνα 124. Κώδικας αιτήματος άδειας χρήστη.	77
Εικόνα 125. Κώδικας διαγραφής άδειας χρήστη.	77
Εικόνα 126. Βασικές μεταβλητές Editor E-Wizard.	78
Εικόνα 127. Κώδικας ανοίγματος/κλεισίματος μενού έξυπνων συσκευών.	80
Εικόνα 128. Κώδικας τοποθέτησης έξυπνης συσκευής στο κομμάτι του συμβάντος στον Editor E-Wizard.	81
Εικόνα 129. Κώδικας τοποθέτησης έξυπνης συσκευής στο κομμάτι της δράσης στον Editor E-Wizard.	82
Εικόνα 130. Κώδικας μεθόδου Trigger(...).	84
Εικόνα 131. Κώδικας διαγραφής έξυπνης συσκευής που ανήκει στις δράσεις στον Editor E-Wizard.	85
Εικόνα 132. Κώδικας ανανέωσης έξυπνης συσκευής που βρίσκεται στις δράσεις στον Editor E-Wizard.	85

Εικόνα 133. Κώδικας ανανέωσης έξυπνης συσκευής που βρίσκεται στα συμβάντα στον Editor E-Wizard.	86
Εικόνα 134. Κώδικας αρχικοποίησης μεταβλητής operator στο κουμπί AND.	86
Εικόνα 135. Κώδικας αποθήκευσης συμβάντων στον Editor E-Wizard.	87
Εικόνα 136. Κώδικας επανεκκίνησης στον Editor E-Wizard.	88
Εικόνα 137. Βασικές μεταβλητές αποθήκευσης κανόνων.	88
Εικόνα 138. Κώδικας αποθήκευσης κανόνων.	89
Εικόνα 139. Κώδικας πληροφοριών έξυπνων συσκευών.	90
Εικόνα 140. Κώδικας αναγραφής κανόνα σε φυσική γλώσσα.	91
Εικόνα 141. Φάκελος "animation" στον κώδικα Android.	91
Εικόνα 142. Μέθοδοι Animation στον κώδικα Android.	92
Boxplot 1. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor Domino.	99
Boxplot 2. Αριθμός αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor Domino.	100
Boxplot 3. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor E - Wizard.	102
Boxplot 4. Αριθμού αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor E-Wizard.	103
Boxplot 5. Απαντήσεις των χρηστών στο SUS ερωτηματολόγιο.	106

Λίστα Πινάκων

Πίνακας 1. Λειτουργικές Απαιτήσεις	11
Πίνακας 2. Μη λειτουργικές Απαιτήσεις.	12
Πίνακας 3. Βασικά endpoints για: α)χρήστες, β)έξυπνες συσκευές και γ)κανόνες.	21
Πίνακας 4. Ανανεωμένα βασικά endpoints για: α)χρήστες ,β)έξυπνες συσκευές και γ)κανόνες.....	22
Πίνακας 5. Streaming endpoints του server.	23
Πίνακας 6. Τελικός Πίνακας Back-End Server.	24
Πίνακας 7. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor Domino.....	98
Πίνακας 8. Αριθμός αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor Domino.	99
Πίνακας 9. Αριθμός λαθών χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor Domino.....	100
Πίνακας 10. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor E - Wizard. ...	101
Πίνακας 11. Αριθμός αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor E - Wizard.	102
Πίνακας 12. Αριθμός Λαθών χρηστών κατά τη διαδικασία δημιουργίας κανόνων στον Editor E - Wizard	103
Πίνακας 13. Απαντήσεις χρηστών στο ερωτηματολόγιο SUS και τα αντίστοιχα SUS σκορ.	105
Πίνακας 14. Μέσος όρος και τυπική απόκλιση κάθε ερώτησης του SUS ερωτηματολογίου και των SUS σκορ.....	105
Πίνακας 15. Μικρότερος, μεγαλύτερος μέσος όρος SUS ερωτηματολογίου και μεγαλύτερο SUS σκορ χρήστη.	105
Πίνακας 16. Ικανοποίηση λειτουργικών απαιτήσεων εφαρμογής.....	108
Πίνακας 17. Ικανοποίηση μη λειτουργικών απαιτήσεων.	108

Ακρωνύμια

IoT	Internet of Things
EUD	End User Development
SUS	System Usability Scale
IFTTT	If This Then That
REST API	Representational State Transfer Application Programming Interface
LDAP	Lightweight Directory Access Protocol
B2B	Business to Business
TLS	Transport Layer Security
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
UI	User Interface
UX	User Experience
URL	Uniform Resource Locator
SSL	Secure Sockets Layer
RSA	Rivest-Shamier-Adleman
IP	Internet Protocol
XML	eXtensible Markup Language
JPA	Java Persistence API
JDBS	Java Database Connectivity
ID	Identification
JSON	JavaScript Object Notation
JWT	JSON Web Token
SDK	Software Development Kit
ΛΑ	Λειτουργική Απαίτηση
ΜΛΑ	Μη Λειτουργική Απαίτηση

Περίληψη

Η έρευνά μας εντάσσεται στο πεδίο του Διαδικτύου των Πραγμάτων και έχει ως σκοπό την ανάπτυξη ενός εργαλείου τελικού χρήστη για την δημιουργία και διαχείριση αυτοματοποιημένων διαδικασιών συνδυάζοντας υπηρεσίες που είναι διαθέσιμες μέσα σε ένα περιβάλλον διάχυτου υπολογισμού. Περιλαμβάνει την δημιουργία δύο περιβαλλόντων ικανών να υποστηρίξουν την έκφραση κανόνων και λογικών σχέσεων μεταξύ έξυπνων συσκευών, Web Services και λειτουργιών κινητών συσκευών.

Η βασική πρόκληση για τη δημιουργία ενός τέτοιου εργαλείου είναι να βρεθούν οι κατάλληλες γνωστικές ενδείξεις που θα ενσωματωθούν στο εργαλείο με τη μορφή μεταφορών (metaphors) ώστε να απεικονίζουν δομές και έννοιες που είναι γνωστές στο φυσικό κόσμο και έτσι οι χρήστες να μπορούν πιο εύκολα να τις μάθουν και να τις εφαρμόσουν.

Το πρώτο περιβάλλον αποτελεί μία δική μας προσέγγιση η οποία περιέχει αρκετά κοινά στοιχεία με το γνωστό παιχνίδι Domino. Θα παρουσιαστεί η κύρια ιδέα διαρρύθμισης του γραφικού περιβάλλοντος, ο τρόπος αλληλεπίδρασης και ένας τρόπος αναπαράστασης κανόνων σε τρισδιάστατο επίπεδο μέσω Domino.

Το δεύτερο περιβάλλον αποτελεί μέρος μίας υφιστάμενης ιδέας την οποία υλοποιήσαμε γραφικά με δικό μας τρόπο. Η γραφική απεικόνιση ομοιάζει με το πρότυπο σύνθεσης Wizard και δημιουργήθηκε για λόγους σύγκρισης.

Τα δύο περιβάλλοντα συμπίπτουν στον προγραμματιστικό σχηματισμό των κανόνων καθώς χρησιμοποιούν το κοινό προγραμματιστικό πρότυπο, πυροδότησης – δράσης (trigger – action). Επίσης, οι ενέργειες που λαμβάνουν χώρα ρυθμίζονται με όμοιο τρόπο. Διαφέρουν μεταξύ τους ως προς το πρότυπο σύνθεσης καθώς και στην διαμόρφωση και απεικόνιση των λογικών σχέσεων που λαμβάνουν χώρα κατά τη δημιουργία του κανόνα.

Σημαντικός παράγοντας αποτελεί ο οπτικός τρόπος αναπαράστασης λειτουργιών και δεδομένων στα δύο περιβάλλοντα έκφρασης κανόνων. Γι' αυτό το λόγο η διαμόρφωσή τους πραγματοποιήθηκε ώστε να είναι πιο φιλική προς τους χρήστες.

Τέλος, στην εργασία περιλαμβάνεται η αξιολόγηση του συστήματος από καθημερινούς χρήστες με στόχο την εύρεση αδυναμιών και την βελτίωση της ευχρηστίας. Η αξιολόγηση του συστήματος πραγματοποιήθηκε μέσω της χρήσης του ερωτηματολογίου System Usability Scale (SUS) υπολογίζοντας το SUS score του κάθε χρήστη.

Λέξεις Κλειδιά: Διαδίκτυο των Πραγμάτων, Περιβάλλον Δημιουργίας Κανόνων, Πυροδότηση, Δράση, Κανόνας, Λογικές Σχέσεις, Έξυπνες Συσκευές, Φιλική Διεπαφή, Αξιολόγηση

Abstract

This diploma presents a review in the Internet of Things area. The goal was to develop a software tool that will enable end-users to create trigger-action programming paradigms specified as rules. Each rule might consist of smart devices, device functions, Web Services as well as the logical relationship between them. In order to achieve this kind of environment we had to come up with a metaphor that users are already familiar with. The well-known Domino game was a great selection to be used as a metaphor. Users are able to visualize their smart devices and Web Services as Dominos in order to combine them and proceed in the creation of trigger-action paradigms. Also, a 3D illustration of rules created by Domino Editor will be put on display. In comparison with the Domino Editor we created a second Editor which implies a different graphical approach to the Wizard composition paradigm.

In order to achieve a user-centric development, a selection of potential users involved in the evaluation of the system. All the proposals were taken into consideration and applied to the system to achieve the best possible user experience.

Keywords: *Internet of Things, Editor, Trigger, Action, Rule, Logical Relationship, Smart Devices, Friendly Interface, Evaluation*

1

Εισαγωγή

1.1 Αντικείμενο διπλωματικής

Η διπλωματική μας εργασία στοχεύει στην δημιουργία ενός προγραμματιστικού περιβάλλοντος που βοηθά τους χρήστες να αυτοματοποιούν διαδικασίες συνδυάζοντας έξυπνες συσκευές, Web Services και λειτουργίες κινητών συσκευών. Αξίζει να αναφερθεί το γεγονός πως δεν υπάρχει κάποιο κοινό μοντέλο στην δημιουργία ενός τέτοιου περιβάλλοντος. Πολλές προσεγγίσεις προσπαθούν μέσω μεταφορών να δώσουν λύση στο πρόβλημα και να εξασφαλίσουν στον χρήστη μια εικόνα για το πως πρόκειται να χρησιμοποιήσει το περιβάλλον.

Η πρόκληση με την οποία ήρθαμε αρχικά αντιμέτωποι, ήταν να δημιουργήσουμε την δικιά μας ιδέα η οποία πρέπει να υφίσταται και ως μεταφορά. Η μεταφορά που χρησιμοποιήσαμε για να λύσουμε το πρόβλημα αυτό είναι βασισμένη σε μια παραδοχή του παιχνιδιού Domino. Στο Domino γνωρίζουμε πως υπάρχουν δύο είδη παιχνιδιού. Το πρώτο είδος παίζεται με βάση τους αριθμούς που αναγράφονται στα Domino, στο δεύτερο και πιο ευχάριστο, τοποθετούμε τα Domino σε σειρά ή παράλληλα το ένα μετά το άλλο. Στόχος σε αυτή τη περίπτωση είναι η ρίψη του τελευταίου Domino. Με την δεύτερη παραδοχή ως μεταφορά και λύση, συνεχίσαμε στην δημιουργία του περιβάλλοντος.

Η δημιουργία του περιβάλλοντος και ο τρόπος κατανόησης της μεταφοράς από τους χρήστες αποτελούν τεχνικές δυσκολίες που κληθήκαμε να αντιμετωπίσουμε κατά την διάρκεια της εργασίας. Βασιζόμενοι στο End-User Development (EUD) μοντέλο για την διεκπεραίωση της, χρειαστήκαμε τη συνεχή αλληλεπίδραση των χρηστών με το σύστημα ως προς την αξιολόγηση και βελτίωσή του [1], [2]. Προσθέσαμε και δικά μας επιπλέον χαρακτηριστικά που κρίναμε αναγκαία να υπάρχουν σε μια πλατφόρμα για τη δημιουργία IoT εφαρμογών τα οποία αναλύουμε αναλυτικά στα επόμενα κεφάλαια.

Με την βοήθεια των χρηστών καθώς και των δυο δοκιμών που έλαβαν χώρα, προέκυψαν δυνατότητες βελτίωσης του περιβάλλοντος και των λειτουργιών του. Όσον αφορά την μεταφορά που επιλέξαμε ήταν πλήρως κατανοητή κατά την επεξήγηση. Συμπεριλάβαμε την εκπαίδευση του χρήστη στην πλατφόρμα καθώς κρίνεται αναγκαία η κατανόηση του περιβάλλοντος και χωρίς την φυσική παρουσία των δημιουργών του.

Δημιουργήσαμε ακόμη ένα δεύτερο εργαλείο για την αυτοματοποίηση διαδικασιών βασισμένο σε υπάρχον πρότυπο με στόχο την σύγκριση των δύο μεταφορών ως προς την ευχρηστία τους και ενδεχομένως τη βελτίωση της μεταφοράς που επιλέξαμε.

Στην πρότασή μας συμπεριλάβαμε και back - end τεχνικά χαρακτηριστικά σε περίπτωση που θα έπρεπε η πλατφόρμα να γίνει διαθέσιμη σε ένα περιβάλλον παραγωγής λαμβάνοντας υπόψιν τις διαθέσιμες state-of-the-art τεχνολογίες και ιδιαίτερα θέματα ασφάλειας της πλατφόρμας.

1.2 Διάρθρωση διπλωματικής

Η διπλωματική εργασία περιλαμβάνει οκτώ (8) κεφάλαια.

Το πρώτο κεφάλαιο αφορά μια εισαγωγή σχετικά με το περιβάλλον και το πλαίσιο στο οποίο εντάσσεται η διπλωματική καθώς και τα προβλήματα που υπάρχουν και πως σκοπεύει να τα επιλύσει.

Στο δεύτερο κεφάλαιο θα παρουσιαστούν αντίστοιχα εργαλεία που βρέθηκαν ύστερα από έρευνα και θα αναφερθούν πληροφορίες σχετικά με το κάθε ένα.

Στο τρίτο κεφάλαιο θα αποτυπωθούν λεπτομερώς οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος μας.

Στο τέταρτο κεφάλαιο θα παρουσιαστεί μια θεωρητική προσέγγιση για τη δημιουργία REST API και βάσης δεδομένων, θα προβληθούν αναλυτικά τα κύρια μέρη της εφαρμογής κινητού υπολογισμού που υλοποιήσαμε καθώς και πληροφορίες σχετικά με τις λειτουργίες, τις δυνατότητες που προσφέρουν και την αλληλεπίδραση των χρηστών μ' αυτές. Επίσης, θα γίνει αναφορά σχετικά με τον σχεδιασμό της διεπαφής και των λειτουργιών της εφαρμογής.

Στο πέμπτο κεφάλαιο θα γίνει αναφορά στην υλοποίηση της εφαρμογής κινητού υπολογισμού, στα βασικά πακέτα που δημιουργήσαμε, στις κλάσεις και δραστηριότητες που είναι καθοριστικές για την ορθή λειτουργία του συστήματος.

Στο έκτο κεφάλαιο θα αναφερθούμε στον έλεγχο του συστήματος. Συγκεκριμένα, θα αποτυπωθούν οι δύο δοκιμές που πραγματοποιήσαμε με χρήστες, τα σενάρια και τα αποτελέσματα των δοκιμών καθώς και η αξιολόγηση των εργαλείων μέσω του ερωτηματολογίου SUS.

Στο έβδομο κεφάλαιο θα αναφερθούμε στις αδυναμίες του εργαλείου μας καθώς και στους τρόπους αντιμετώπισης αυτών αλλά και γενικότερα στη βελτίωση της εφαρμογής.

Τέλος, στο όγδοο κεφάλαιο θα μοιραστούμε τα συμπεράσματα στα οποία καταλήξαμε έπειτα από την εκπόνηση της διπλωματικής εργασίας.

2

Μελέτη Υφιστάμενης Κατάστασης

2.1 Εργαλεία για δημιουργία IoT εφαρμογών

Στον παρόν κεφάλαιο θα εξετάσουμε εφαρμογές που υπάρχουν στην αγορά και αφορούν την δημιουργία κανόνων μεταξύ έξυπνων συσκευών, Web Services και λειτουργιών που παρέχει το κινητό τηλέφωνο. Έπειτα από έρευνα που έγινε, διαπιστώθηκε ότι υπάρχουν ποικίλες εφαρμογές οι οποίες προσφέρονται για χρήση μέσω προγραμμάτων στον υπολογιστή, εφαρμογών κινητού υπολογισμού και διαδικτυακών εργαλείων. Κάθε εφαρμογή έχει το δικό της πλαίσιο για την δημιουργία κανόνων και ορίζει ποιες υπηρεσίες θα προσφέρονται. Υπάρχουν εφαρμογές που υποστηρίζουν υπηρεσίες που μπορούν να δημιουργήσουν κανόνες σε συνδυασμό των τριών παραπάνω συστατικών (έξυπνων συσκευών, Web Services και λειτουργιών κινητού τηλεφώνου). Από την άλλη μεριά, υπάρχουν εφαρμογές που εξυπηρετούν δύο ή μία εξ' αυτών.

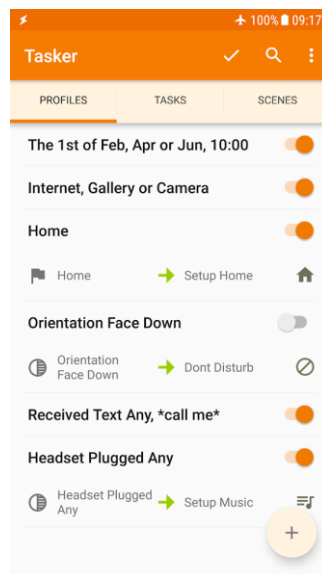
Κάθε εφαρμογή έχει τον δικό της τρόπο δημιουργίας, απεικόνισης και αναγραφής των κανόνων. Υπάρχουν κοινά σημεία αλλά κάθε μία έχει την δική της ιδέα και σκοπό τον οποίο εξυπηρετεί [3], [4].

2.1.1 Tasker

Πρόκειται για εφαρμογή κινητού υπολογισμού η οποία αποτελεί την εξέλιξη της εφαρμογής *Apt* η οποία προγραμματίστηκε για κινητά που είχαν λειτουργικό σύστημα *Palm* και η πρώτη έκδοση ήταν το 2007¹. Πλέον, προσφέρεται σε κινητά που έχουν λειτουργικό σύστημα *Android* (Εικόνα 1). Η βασική της λειτουργία είναι να

¹ <https://tasker.joaoapps.com/history.html>

εκτελεί εργασίες (*tasks*) σύμφωνα με συνθήκες (*contexts*) που έχουν οριστεί από τους χρήστες². Οι συνδυασμοί συνθηκών – εργασιών αποτελούν το προφίλ χρήστη (*profile*) και κάθε προφίλ χρήστη έχει τους δικούς του συνδυασμούς. Δεν παρέχει επικοινωνία με έξυπνες συσκευές και οι συνδυασμοί που δημιουργούνται αφορούν λειτουργίες που επιθυμούν να εκτελεστούν στο κινητό τηλέφωνο. Οι εργασίες μπορούν να πραγματοποιηθούν είτε χειροκίνητα μέσω της εφαρμογής είτε αυτόματα διότι η εφαρμογή παρακολουθεί τις συνθήκες και όταν ικανοποιηθούν εκτελούνται οι αντίστοιχες εργασίες. Οι συνθήκες μπορεί να είναι η ώρα, ημερομηνία, κίνηση, τοποθεσία, κάποιες ήδη εγκατεστημένες εφαρμογές κινητού υπολογισμού και όχι μόνο. Οι εργασίες αφορούν ενέργειες που μπορούν να εκτελεστούν στο κινητό τηλέφωνο όπως εμφάνιση ειδοποίησης, έναρξη της λίστας αναπαραγωγής μουσικής. Τέλος, αφορά κυρίως χρήστες που είναι εξοικειωμένοι με την τεχνολογία και διαθέτουν μια εμπειρία στον χειρισμό εφαρμογών κινητού υπολογισμού [4].



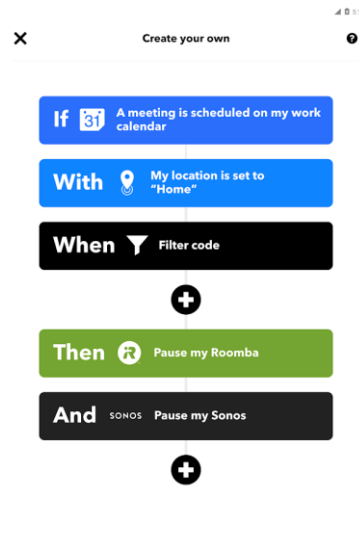
Εικόνα 1. Εφαρμογή "Tasker".

2.1.2 IFTTT

Το *IFTTT* πρόκειται για μία από της πιο δημοφιλείς εφαρμογές για την δημιουργία κανόνων. Προσφέρεται μέσω εφαρμογής κινητού υπολογισμού για κινητά τηλέφωνα που διαθέτουν λειτουργικό σύστημα *Android*, *IOS* και μέσω διαδικτυακού ιστότοπου. Οι κανόνες που δημιουργούνται ονομάζονται συνταγές (*recipes*) και αποτελούνται από ένα συμβάν και μία δράση (*action*). Για να πραγματοποιηθεί μία συνταγή, παρακολουθείται το συμβάν που έχει οριστεί και εφ' όσον ισχύσει τότε πυροδοτείται η αντίστοιχη δράση. Το γεγονός και η δράση μπορούν να αφορούν υπάρχουσες εφαρμογές κινητού υπολογισμού που είναι εγκατεστημένες στο κινητό τηλέφωνο, υπηρεσίες που προσφέρονται από το κινητό όπως ημερομηνία, ώρα, τοποθεσία καθώς και έξυπνες συσκευές. Ως μειονέκτημα μπορεί να αναφερθεί ότι στην δωρεάν έκδοση της εφαρμογής μία συνταγή δεν μπορεί να διαθέτει παραπάνω από ένα γεγονός και παραπάνω από μία δράση. Αντιθέτως, στην έκδοση επί πληρωμή δίνεται η δυνατότητα ορισμού περισσότερων εκ του ενός trigger και action (Εικόνα 2). Το ιδιαίτερο χαρακτηριστικό που το κάνει να ξεχωρίζει από τις υπόλοιπες εφαρμογές είναι ότι μπορεί να επικοινωνήσει με

² <https://tasker.joaoapps.com/index.html>

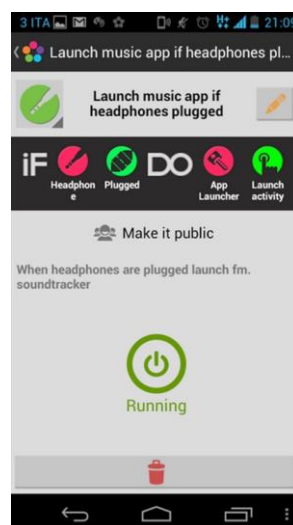
γνωστά Web Services όπως το Facebook, Instagram, eBay, Youtube. Λόγω του φιλικού περιβάλλοντος που διαθέτει και της απλής δομής δημιουργίας συνταγών απευθύνεται και σε χρήστες που δεν διαθέτουν ιδιαίτερη εξοικίωση με την τεχνολογία και προσφέρεται δωρεάν [4], [5], [6].



Εικόνα 2. Δημιουργία κανόνα στο "IFTTT".

2.1.3 Atooma

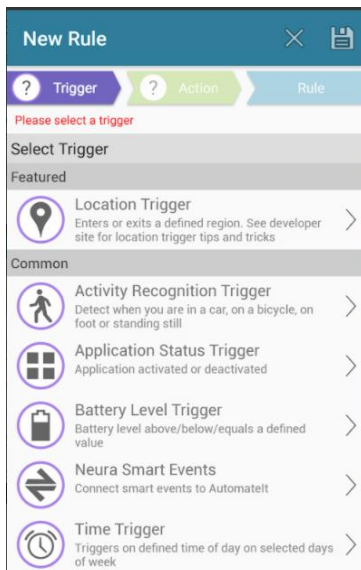
Η εφαρμογή *Atooma* είναι μία από τις λίγες εφαρμογές στον συγκεκριμένο τομέα η οποία μπορεί να συνδυάσει λειτουργίες κινητού τηλεφώνου, έξυπνες συσκευές και Web Services στην σύνθεση ενός κανόνα. Ο συνδυασμός των λειτουργιών κινητού τηλεφώνου, έξυπνων συσκευών και Web Services επιτυγχάνεται εφ' όσον όλα τα παραπάνω είναι συνδεδεμένα με το κινητό τηλέφωνο. Η εφαρμογή κινητού υπολογισμού που διαθέτει μπορεί να χρησιμοποιηθεί σε κινητά τηλέφωνα που διαθέτουν λειτουργικό σύστημα Android και IOS. Όσον αφορά τους κανόνες, μπορούν συμπεριληφθούν περισσότερα από ένα συμβάντα και δράσεις (Εικόνα 3). Στοχεύει σε χρήστες που δεν είναι εξοικειωμένοι με την τεχνολογία και διατίθεται στην αγορά επί πληρωμή αλλά και δωρεάν [4], [7].



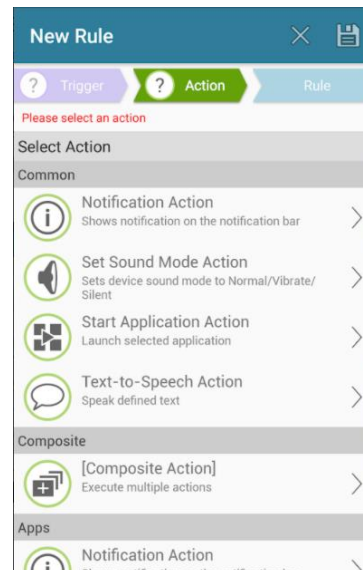
Εικόνα 3. Δημιουργία κανόνα στο "Atooma".

2.1.4 AutomateIt

Η εφαρμογή *AutomateIt* εκτελεί λειτουργίες που αφορούν το κινητό τηλέφωνο. Οι χρήστες ορίζουν τα συμβάντα (Εικόνα 4) και τις δράσεις (Εικόνα 5) που επιθυμούν έτσι ώστε να συνθέσουν τον κανόνα αλλά παρέχει κάποιους ήδη αποθηκευμένους κανόνες. Απευθύνεται σε χρήστες που είναι ανεξοικειωτοι με την τεχνολογία και προσφέρεται δωρεάν αλλά και επί πληρωμή [4].



Εικόνα 4. Δημιουργία trigger στο "AutomateIt".



Εικόνα 5. Δημιουργία action στο "AutomateIt".

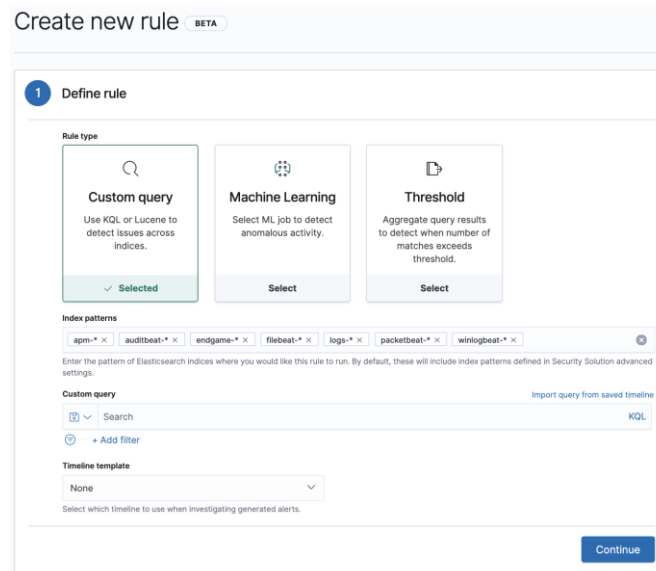
2.1.5 Bip.io

Το *Bip.io* αποτελεί μια εφαρμογή η οποία αφορά λειτουργίες σχετικά με Web Services. Η χρήση της γίνεται μέσω υπολογιστή και ως metaphor χρησιμοποιείται ένας Γράφος. Τα Web Services παρουσιάζονται ως κόμβοι (*nodes*) τους οποίους οι χρήστες ενώνουν μεταξύ τους με βέλη (*arrows*). Εφ' όσον ενωθούν δύο ή περισσότεροι κόμβοι-Web Services τότε οι χρήστες πρέπει να καθορίσουν το συμβάν και την δράση. Μέσω της κατάλληλης καθοδήγησης που τους παρέχει η εφαρμογή καθορίζουν το συμβάν και την δράση καθώς και τις ιδιότητες ενεργοποίησης αυτών. Προσφέρεται στην αγορά δωρεάν αλλά και επί πληρωμή. Απευθύνεται σε χρήστες που δεν έχουν ιδιαίτερη γνώση της τεχνολογίας [4].

2.1.6 Elastic.io

Οι υπηρεσίες που προσφέρει η εφαρμογή *Elastic.io* εξυπηρετούν λειτουργίες με Web Services και διατίθεται για χρήση μέσω υπολογιστή (Εικόνα 6). Έχει σχεδιαστεί για να χρησιμοποιείται κυρίως για επιχειρησιακούς

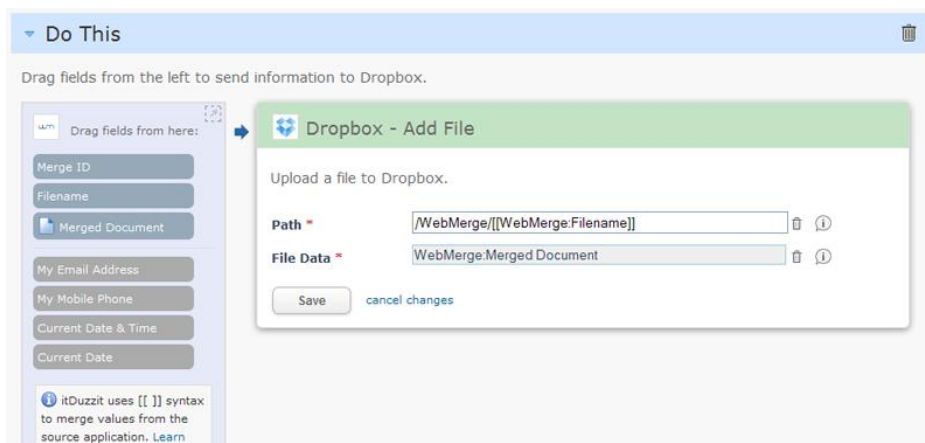
σκοπούς³. Διατίθεται στην αγορά επί πληρωμή και μπορεί να χρησιμοποιηθεί από χρήστες που δεν είναι εξοικειωμένοι με την τεχνολογία [4].



Εικόνα 6. Δημιουργία κανόνα στο "Elastic.io".

2.1.7 itDuzzit

Πρόκειται για ένα διαδικτυακό εργαλείο το οποίο εξυπηρετεί λειτουργίες Web Service. Όπως το IFTTT έτσι και το itDuzzit στη σύνθεση του κανόνα μπορεί να περιέχει ένα συμβάν και μία δράση (Εικόνα 7). Διατίθεται στην αγορά επί πληρωμή αλλά και δωρεάν. Απευθύνεται σε χρήστες που είναι ανεξοικειωτοι με την τεχνολογία [4].



Εικόνα 7. Δημιουργία κανόνα στο "itDuzzit".

³ <https://www.elastic.io/team/>

Node-RED

☒ input

☐ output

☐ function

☐ social

☐ storage

☐ analysis

☐ advanced

inject

catch

mqtt

http

websocket

tcp

udp

serial

Home Energy

Node-RED GitHub Hooks

Filter dups

msg.payload

Node-RED GitHub Hooks

/home/kno1eary/github_hooks.json

timestamp

msg.payload

Info

debug

Node

Name: SlackHook

Type: http in

ID: 40c91d4d.b36e4

Properties

Provides an input node for http requests, allowing the creation of simple web services.

The resulting message has the following properties:

- msg.req : [http request](#)
- msg.res : [http response](#)

For POST/PUT requests, the body is available under `msg.req.body`. This uses the [Express bodyParser middleware](#) to parse the content to a JSON object.

By default, this expects the body of the request to be url encoded:

```
foo=bar&this=that
```

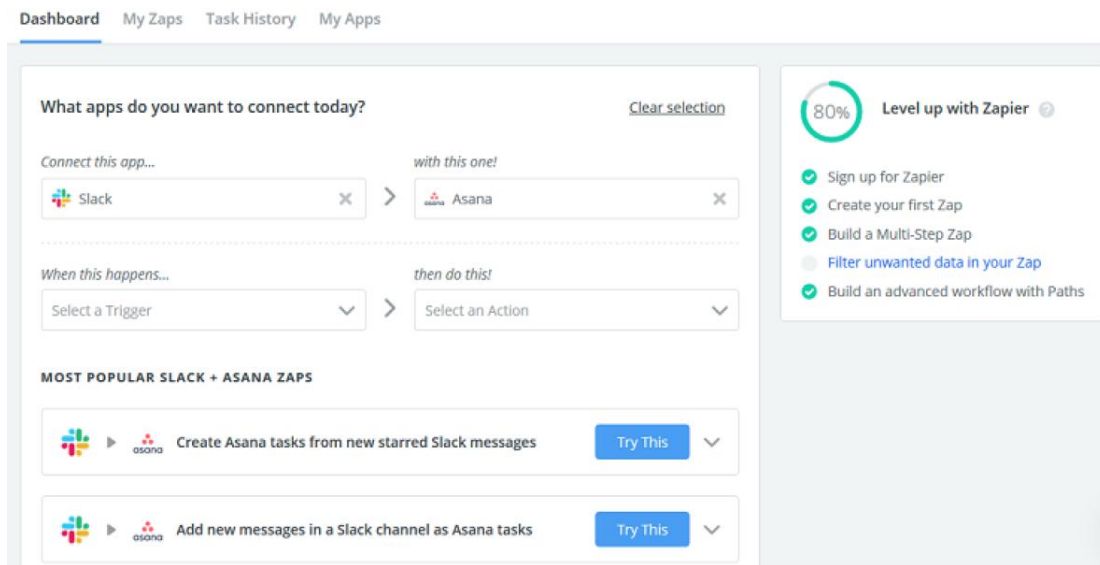
To send JSON encoded data to the node, the content-type header of the request must be set to `application/json`.

Note: This node does not send any response to the http request. This should be done with a subsequent HTTP Response node.

2.1.9 WigWag

2.1.10 Zapier

⁴ <https://nodered.org/>



Εικόνα 9. Δημιουργία κανόνα στο "Zapier".

3

Ανάλυση Απαιτήσεων

3.1 Εισαγωγή

Η εφαρμογή κινητού υπολογισμού που κληθήκαμε να αναπτύξουμε έχει ως κύριο στόχο την δημιουργία ενός περιβάλλοντος φιλικού προς τους χρήστες για την δημιουργία κανόνων σ' ένα έξυπνο περιβάλλον. Απευθύνεται σε χρήστες οι οποίοι κατέχουν έξυπνες συσκευές και επιθυμούν τον έλεγχο και τον συνδυασμό τους αλλά και σε χρήστες που επιθυμούν να αυτοματοποιήσουν κάποιες διαδικασίες τους. Στόχος της ενότητας είναι οι χρήστες να αντιληφθούν τις δυνατότητες της εφαρμογής κινητού υπολογισμού καθώς θα αναφερθούν λεπτομερώς οι λειτουργικές και μη λειτουργικές απαιτήσεις.

3.2 Λειτουργικές Απαιτήσεις

Παρακάτω, απαριθμούνται οι λειτουργίες που θα πρέπει να παρέχει η εφαρμογή και πως θα πρέπει αντιδρά σε συγκεκριμένες εισόδους και καταστάσεις (Πίνακας 1).

ΛΑ1.	Υποστήριξη περιβάλλοντος για την δημιουργία κανόνων με τη χρήση έξυπνων συσκευών υπό την μορφή domino.
ΛΑ2.	Υποστήριξη περιβάλλοντος για την δημιουργία κανόνων με τη χρήση έξυπνων συσκευών μέσω κατευθυντήριου οδηγού.
ΛΑ3.	Παροχή εκπαίδευσης για την χρήση των περιβαλλόντων που αφορούν την δημιουργία κανόνων.
ΛΑ4.	Υποστήριξη διαχωρισμού έξυπνων συσκευών σε συμβάντα (events) και δράσεις (actions).
ΛΑ5.	Υποστήριξη σύνδεσης συμβάντων σύμφωνα με τις λογικές πράξεις «AND» και «OR».
ΛΑ6.	Παροχή αναγραφής ρύθμισης έξυπνης συσκευής υπό τη μορφή απλού κειμένου

	κατά τη διάρκεια δημιουργίας του κανόνα.
ΛΑ7.	Παροχή διαγραφής έξυπνης συσκευής κατά τη διάρκεια δημιουργίας κανόνα στο κομμάτι του συμβάντος και της δράσης.
ΛΑ8.	Παροχή ανανέωσης έξυπνης συσκευής κατά τη διάρκεια δημιουργίας κανόνα στο κομμάτι του συμβάντος και της δράσης.
ΛΑ9.	Υποστήριξη επανεκκίνησης περιβάλλοντος δημιουργίας κανόνων με τη χρήση έξυπνων συσκευών υπό την μορφή domino.
ΛΑ10.	Υποστήριξη επανεκκίνησης περιβάλλοντος δημιουργίας κανόνων με τη χρήση έξυπνων συσκευών μέσω κατευθυντήριου οδηγού.
ΛΑ11.	Παροχή αποθήκευσης κανόνα στο περιβάλλον δημιουργίας κανόνων με τη χρήση έξυπνων συσκευών υπό την μορφή domino.
ΛΑ12.	Παροχή αποθήκευσης κανόνα με όνομα εξ' ορισμού από χρήστες στο περιβάλλον δημιουργίας κανόνων με τη χρήση έξυπνων συσκευών μέσω κατευθυντήριου οδηγού.
ΛΑ13.	Υποστήριξη λίστας αποθηκευμένων κανόνων.
ΛΑ14.	Παροχή διαγραφής αποθηκευμένων κανόνων.
ΛΑ15.	Παροχή προβολής αποθηκευμένου κανόνα υπό τη μορφή απλού κειμένου.
ΛΑ16.	Υποστήριξη προβολής αποθηκευμένου κανόνα σε τρισδιάστατο χώρο υπό την μορφή βίντεο εφ' όσον οι αποθηκευμένοι κανόνες έχουν δημιουργηθεί στον Editor Domino.
ΛΑ17.	Παροχή μεταφόρτωσης εικόνας κάτοψης του χώρου.
ΛΑ18.	Παροχή επεξεργασίας της εικόνας κάτοψης που μεταφορτώθηκε.
ΛΑ19.	Υποστήριξη τοποθέτησης έξυπνων συσκευών στην εικόνα κάτοψης.
ΛΑ20.	Υποστήριξη αποθήκευσης εικόνας κάτοψης μετά την επεξεργασία της.
ΛΑ21.	Παροχή πληροφοριών σχετικά με τις ρυθμίσεις των έξυπνων συσκευών.

Πίνακας 1. Λειτουργικές Απαιτήσεις.

3.3 Μη λειτουργικές Απαιτήσεις

Όσον αφορά τις μη λειτουργικές απαιτήσεις αναφέρονται σε περιορισμούς στις υπηρεσίες ή λειτουργίες της εφαρμογής κινητού υπολογισμού μας. Επίσης, στόχος των μη λειτουργικών απαιτήσεων είναι να διαμορφωθούν έτσι ώστε να βελτιώσουν την αλληλεπίδραση του χρήστη με την εφαρμογή μας. Παρακάτω απαριθμούνται οι μη λειτουργικές απαιτήσεις (Πίνακας 2).

ΜΛΑ1.	Η εφαρμογή κινητού υπολογισμού θα υλοποιηθεί σε γλώσσα προγραμματισμού Java και για την μεταγλώττιση του κώδικα θα χρησιμοποιηθεί το Android Studio.
ΜΛΑ2.	Η μικρότερη έκδοση λειτουργικού συστήματος που απαιτεί η εφαρμογή είναι Android 8.1 (Oreo API level 27).
ΜΛΑ3.	Το σύστημα της εφαρμογής θα πρέπει να διαθέτει όσο το δυνατόν πιο φιλικό περιβάλλον γίνεται στη χρήση.
ΜΛΑ4.	Θα πρέπει να υπάρχει προσαρμοστικότητα του γραφικού περιβάλλοντος της εφαρμογής ανάλογα με την οθόνη που διαθέτει η συσκευή που χρησιμοποιεί την εφαρμογή.
ΜΛΑ5.	Οι χρήστες για την ολοκλήρωση της δημιουργίας κανόνα θα πρέπει να έχουν προσθέσει έξυπνες συσκευές και στο κομμάτι του συμβάντος (event) και στο κομμάτι της δράσης (action).
ΜΛΑ6.	Οι χρήστες για την επεξεργασία μεταφορτωμένης εικόνας κάτοψης θα πρέπει να έχουν μεταφορτώσει την εικόνα της κάτοψης που επιθυμούν.

- | | |
|---------------|--|
| ΜΛΑ7. | Η εφαρμογή θα πρέπει να μεταφράζει την λογική σύνδεση των έξυπνων συσκευών στα περιβάλλοντα δημιουργίας κανόνα. |
| ΜΛΑ8. | Για την καλύτερη κατανόηση του χρήστη, η εφαρμογή θα πρέπει να αντικατοπτρίζει τις έξυπνες συσκευές ως “dominos” στο περιβάλλον δημιουργίας κανόνα υπό την μορφή domino. |
| ΜΛΑ9. | Για την καλύτερη εμπειρία του χρήστη, η εφαρμογή θα πρέπει να αντικατοπτρίζει τις έξυπνες συσκευές ως εικονίδια που παραπέμπουν στην πραγματική τους μορφή στο περιβάλλον δημιουργίας κανόνα μέσω κατευθυντήριου οδηγού. |
| ΜΛΑ10. | Για την καλύτερη εμπειρία του χρήστη, η εφαρμογή θα πρέπει να αναπαριστά, στο κομμάτι της δράσης (action), τις ενέργειες της/των έξυπνης/ων συσκευής/ων με “emoji (s)”. |
| ΜΛΑ11. | Η εφαρμογή θα πρέπει να μην επιτρέπει την δημοσιοποίηση των προσωπικών δεδομένων του χρήστη. |

Πίνακας 2. Μη λειτουργικές Απαιτήσεις.

4

Σχεδιασμός Συστήματος

4.1 Προσέγγιση

Στην ενότητα αυτή θα παρουσιάσουμε και θα περιγράψουμε τους τρόπους σχεδίασης μιας φιλικής διεπαφής για μια πλατφόρμα Internet of Things και τη θεωρητική αρχιτεκτονική προσέγγιση με την οποία σχεδιάστηκε η εφαρμογή.

Η θεωρητική αρχιτεκτονική προσέγγιση μας περιλαμβάνει τις απαραίτητες απαιτήσεις και προϋποθέσεις για την δημιουργία REST API με χρήση Web Service (s) και Βάσης Δεδομένων σε ένα περιβάλλον παραγωγής (production environment). Η δημιουργία του Web Service καθώς και η σύνδεση με την Βάση Δεδομένων μπορεί να επιτευχθεί με την χρήση του SpringBoot framework σε συνδυασμό με το Hibernate framework, το οποία θα καλύψουμε αναλυτικά στα επόμενα κεφάλαια.

Η φιλική διεπαφή σε μια τέτοιου είδους πλατφόρμα θα πρέπει να έχει σχεδιαστεί με τη χρήση κάποιου UI/UX εργαλείου, στην εφαρμογή μας σχεδιάστηκε με το JustInMind Prototype και υλοποιήθηκε σε Android Studio με τη χρήση native γλώσσας προγραμματισμού Java.

Επιπλέον, έγινε η χρήση του Open-Source Game-Engine Unity σε γλώσσα προγραμματισμού C# για τη δημιουργία animations που συμβάλουν στην καλύτερη εμπειρία των χρηστών.

4.2 Δημιουργία REST API και Βάσης Δεδομένων

Σε αυτό το κεφάλαιο θα δούμε με ποιο τρόπο μπορούμε να επιτύχουμε την επικοινωνία των Έξυπνων Συσκευών με έναν κεντρικό Server καθώς θα συμπεριλάβουμε πληροφορίες σχετικά με την βάση δεδομένων καθώς και την αυθεντικοποίηση χρηστών. Σε ένα τέτοιο σύστημα έχει μεγάλη σημασία και η

ασφάλειά του, έτσι θα παρουσιάσουμε τρόπους με τους οποίους μπορεί να προστατευτεί ο κεντρικός server, οι χρήστες του και οι έξυπνες συσκευές [9].

4.2.1 Βάση Δεδομένων

Η βάση δεδομένων αποτελεί υποσύνολο των Directories (Ευρετήρια) που χρησιμοποιούνται σχετικά με την αποθήκευση δεδομένων. Η διαφορά που υπάρχει μεταξύ των ευρετηρίων είναι στον τρόπο, στον σκοπό και στη μορφή αποθήκευσης δεδομένων.

Παραδείγματα ευρετηρίων:

1. LDAP για αυθεντικοποίηση.
2. Real Time Database (No - SQL) , για προσπέλαση δεδομένων σε πραγματικό χρόνο, συνήθως τα δεδομένα είναι υπό την μορφή JSON.
3. Relational Database για συσχετίσεις μεταξύ οντοτήτων υπό την μορφή πινάκων

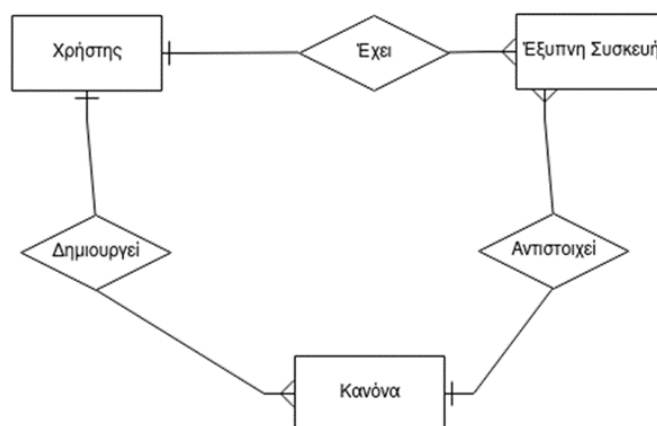
Σε ένα περιβάλλον παραγωγής και ειδικά σε ένα περιβάλλον B2B, ενδέχεται να χρειαστεί ένας συνδυασμός των παραπάνω, παρόλα αυτά, η χρήση μιας σχεσιακής βάσης δεδομένων καθώς, ενός πολυ-νηματικού (Multi Threaded) server και η προσπέλαση /αποστολή δεδομένων μέσω διαδικτυακών ροών (Web-Sockets) αρκεί για να εξυπηρετεί εκατομμύρια χρήστες.

Ξεκινώντας την δημιουργία της βάσης δεδομένων θα πρέπει να γνωρίζουμε τις οντότητες που θα συμπεριλάβουμε καθώς και τις συσχετίσεις μεταξύ αυτών (Εικόνα 10).

Οντότητες:

- Χρήστης
- Έξυπνη συσκευή
- Κανόνας

Συσχετίσεις:



Εικόνα 10. Συσχετίσεις βάσης δεδομένων.

Ιδιότητες Χρήστη:

- Id (unique, primary key)
- Username

- Password
- Rule (foreign key)
- Smart Devices (foreign key)
- Age
- Gender
- Employment

Ιδιότητες Έξυπνης Συσκευής:

- Id (unique, primary key)
- Settings
- Position
- Connected

Ιδιότητες Κανόνων:

- Id (unique, primary key)
- Name
- Smart Devices
- Event
- Action

4.2.1.1 Τύποι μεταβλητών των ιδιοτήτων στην σχεσιακή βάση δεδομένων

Πίνακας Χρηστών:

- Id (unique, primary key) - integer
- Username - varchar
- Password - varchar
- Rule (foreign key) - integer
- Smart Devices (foreign key) - integer
- Age - integer
- Gender - varchar
- Employment – varchar

Πίνακας Έξυπνης Συσκευής:

- Id (unique, primary key) - integer
- Settings - varchar, π.χ. “TV: Masterchef,GameOfThrones”, “Clock: 00:00-23:00”, δηλαδή, “Name : setting1,setting2,setting3”
- Position (Θέση στο σπίτι) - varchar
- Connected (Συνδεδεμένη στον server για επικοινωνία) - boolean

Πίνακας Κανόνων:

- Id (unique, primary key) - integer
- Name - varchar
- Smart Devices (foreign key) - integer
- Event - varchar
- Action – varchar

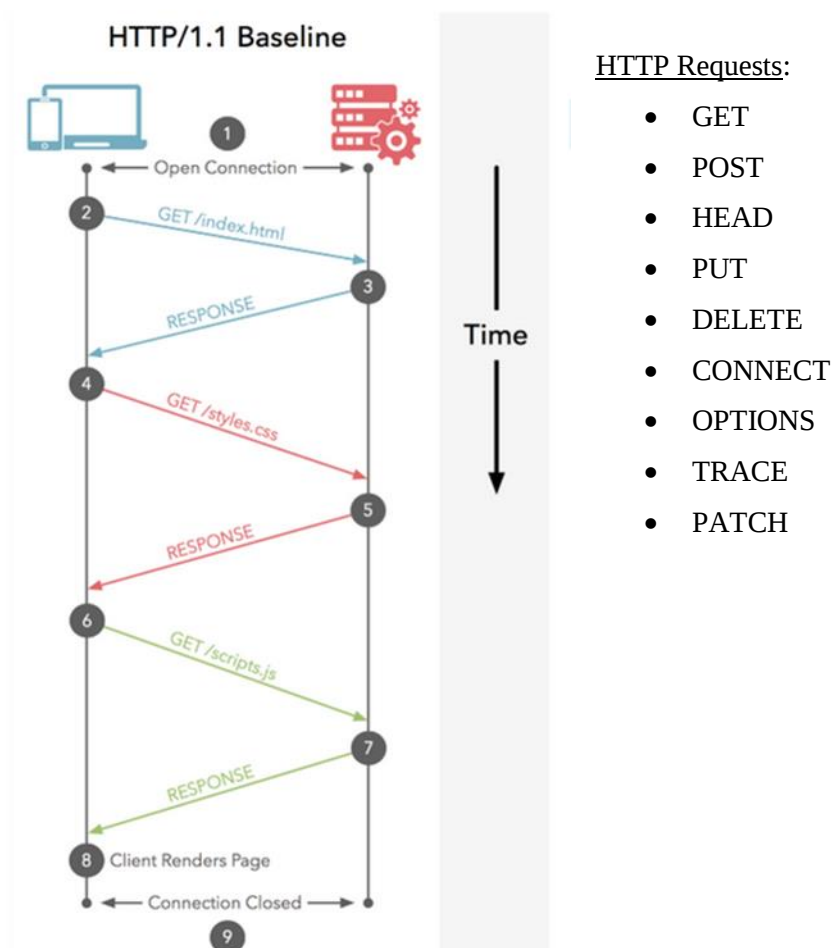
4.2.2 Ασφάλεια Συστήματος

Στην συγκεκριμένη ενότητα θα καλύψουμε τις βασικές τεχνολογικές υποδομές και πρωτόκολλα επικοινωνίας τα οποία χρησιμοποιούνται σε συστήματα στις μέρες μας καθώς αποτελούν σημαντικό παράγοντα για το σύστημα μας.

4.2.2.1 Επιλογή Πρωτοκόλλου Επικοινωνίας

HTTP/1.1

Πρόκειται για το πρωτόκολλο που χρησιμοποιείται στις μέρες μας για την επικοινωνία Client – Server (Εικόνα 11). Χρησιμοποιείται από την μεριά του Client για την αποστολή (request) δεδομένων μέσω του Παγκόσμιου Ιστού. Οι Server απαντούν με δεδομένα και τύπους καταστάσεων γνωστά ως Responses και Status Codes. Είναι γνωστό ως μοντέλο ερώτησης-απάντησης.



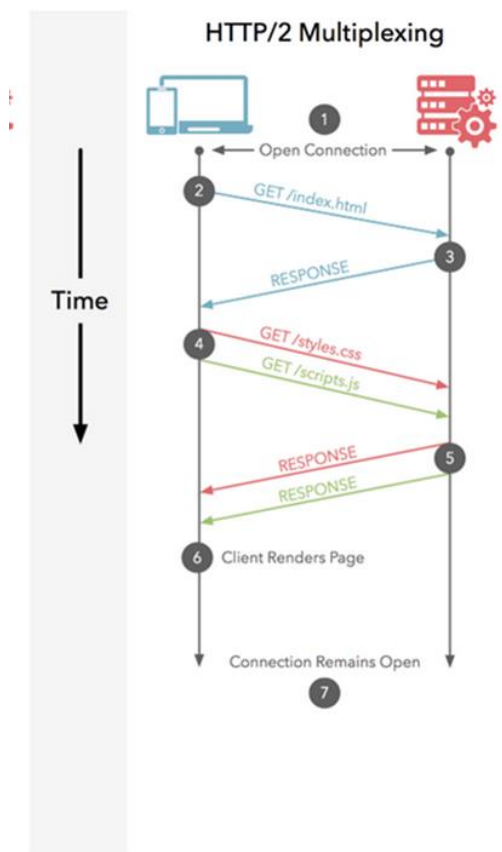
Εικόνα 11. Πρωτόκολλο Επικοινωνίας HTTP/1.1.

HTTP Responses Codes:

1. **1xx** χρησιμοποιούνται για την ενημέρωση του χρήστη, όμως χρησιμοποιούνται σπάνιο, για παράδειγμα, ο διακομιστής συμφωνεί να χειριστεί την αίτηση του πελάτη (Status Code: **100**).
2. **2xx** σημαίνουν ότι ο χειρισμός της αίτησης έγινε με επιτυχία και ότι επιστρέφεται το περιεχόμενο αν υπήρχε.
3. **3xx** πληροφορούν στον πελάτη πως πρέπει να ψάξει αλλού, είτε χρησιμοποιώντας μια διαφορετική διεύθυνση URL, είτε εξετάζοντας την κρυφή μνήμη, για παράδειγμα, η σελίδα μετακινήθηκε (Status Code: **301**).
4. **4xx** σημαίνουν ότι η αίτηση απέτυχε λόγω κάποιου σφάλματος του πελάτη, όπως άκυρη αίτηση ή ανύπαρκτη σελίδα. Παράδειγμα: απαγορευμένη σελίδα (**403**), η σελίδα δεν βρέθηκε (**404**).

HTTP/1.2

Το πρωτόκολλο 1.2 χρησιμοποιείται για λόγους επιδόσεων καθώς εμπλουτίζεται το μοντέλο ερώτηση-απάντηση με μία συνεχής ροή δεδομένων και χρησιμοποιείται το μοντέλο πολλαπλές ερωτήσεις – πολλαπλές απαντήσεις, έτσι έχουμε και λιγότερα αιτήματα, βέβαια λόγω του ανοιχτού δίαυλου επικοινωνίας μπορεί να γίνει στόχος επιθέσεων. Για αυτό τον λόγο μπορούμε να προσθέσουμε μια πληροφορία αυθεντικοποίησης στην κεφαλίδα του αιτήματος (header) (Εικόνα 12).



Εικόνα 12. Πρωτόκολλο Επικοινωνίας HTTP/1.2.

HTTPS

Το HTTPS πρωτόκολλο δεν είναι τίποτα παραπάνω παρά ένα ασφαλές HTTP, έχει χρησιμοποιηθεί το TLS (1.1/1.2) για να δημιουργηθεί το HTTPS. Το οποίο επιτυγχάνεται μέσω της απόκτησης του SSL

Certificate από αρκετές online υπηρεσίες φιλοξενίας. Στην περίπτωση του τοπικού server (Σε επίπεδο Lan) το HTTPS δεν χρησιμοποιείται.

.

4.2.2.2 Χρήστες και Έξυπνες Συσκευές

Για την ασφαλή μεταφορά πληροφοριών από τους χρήστες στον server αρκεί το HTTPS πρωτόκολλο καθώς χρησιμοποιεί ασύμμετρη και συμμετρική κρυπτογράφηση των πακέτων.

Στην ασύμμετρη κρυπτογραφία, κρυπτογραφούνται τα πακέτα με χρήση του ιδιωτικού κλειδιού και η αποκρυπτογράφηση του από τον server με χρήση του δημόσιου κλειδιού του χρήστη. Ένα ισχυρό μήκος RSA κλειδιού είναι **3017-bit**.

Στην συμμετρική κρυπτογραφούνται και αποκρυπτογραφούνται με τη χρήση ενός κοινού κλειδιού.

Επιλογή δυνατού κωδικού για είσοδο στο σύστημα, ένας τέτοιος κωδικός μπορεί να περιέχει συνδυασμό από:

- ο Σύμβολα: @#!\$%^&*
- ο Αριθμούς: 0-9
- ο Κεφαλαία και πεζά γράμματα: A-Z και a-z
- ο Όμοιους χαρακτήρες: i, l, 1, L, o, 0, O

Μια συμβολοσειρά μήκους 16, θεωρείται ισχυρός κωδικός πρόσβασης, σε κάθε περίπτωση θα πρέπει να υπάρχει κρυπτογραφημένος στην βάση δεδομένων.

Spring Security

Η επιλογή του framework για την δημιουργία του server / back-end θα είναι το Spring Framework, όπως θα δούμε και παρακάτω. Χρησιμοποιώντας το Spring Security πακέτο έχουμε πρόσβαση σε χειρισμούς αυθεντικοποίησης χρηστών, δημιουργία απαιτήσεων των requests που λαμβάνουμε καθώς υπάρχει και η δυνατότητα ανάθεσης ρόλων στους χρήστες. Επίσης μπορούμε να χρησιμοποιήσουμε την cache του server για να μετράμε τις αποτυχημένες προσπάθειες των χρηστών για είσοδο στην εφαρμογή και έπειτα μπορούμε να προβούμε σε μπλοκάρισμα της διεύθυνσης IP για ένα χρονικό διάστημα. Ακόμα θα δημιουργήσουμε ένα token αυθεντικοποίησης για να έχουν οι χρήστες πρόσβαση στον server και να ανταλλάζουν δεδομένα.

4.2.3 Κεντρικός Server

Η δημιουργία του Server θα γίνει με τη χρήση του Spring Framework δημιουργώντας ένα Maven project για τη προσθήκη των πακέτων / βιβλιοθηκών μέσω XML καθώς θα χρησιμοποιηθεί η γλώσσα προγραμματισμού Java , για τον προγραμματισμό του. Θα γίνει η χρήση αναδραστικού ιστού (Reactive Web - Sockets) για την ζωντανή (live) επικοινωνία των έξυπνων συσκευών και ικανοποίησης των κανόνων.

Βιβλιοθήκες που θα χρειαστούμε:

- Spring Security
- Spring Web (REST API σε μορφή JSON)
- Spring Reactive – Web (Χρήση Web Flux)
- JDBC – API (Σύνδεση με την βάση)
- Spring Data JPA (Δημιουργία instances των tables)
- Json Web Token (Token αυθεντικοποίησης με συνδυασμό του Spring Security)

- Spring Boot Actuator (Παρακολούθηση server)

Το πρώτο βήμα είναι να δημιουργήσουμε όλες τις Οντότητες (Entities) υπό την ακριβή μορφή που περιεγράφηκαν στην ενότητα της βάσης δεδομένων έτσι ώστε να υπάρχει αντιστοίχιση στην δημιουργία του τοπικού repository. Έτσι με την χρήση του **@RestController** annotation όταν θα επιστρέφονται οι Οντότητες στο endpoint θα γίνονται αυτόματα υπό την μορφή **JSON Object**.

4.2.3.1 Δημιουργία Spring Boot Annotations

Δημιουργία Spring Boot Server:

- @SpringBootApplication

Δημιουργία API endpoints:

- @RestController
- @Controller
- @RequestMapping

Αντιστοίχιση Entity με Table στη βάση δεδομένων:

- @Entity
- @Table (name= "users")
- @Id
- @Column (name= "username")
- @ManyToOne
- @JoinColumn (name= "smart_device_foreign_key")

Δημιουργία Repositories:

- @EnableJpaRepositories

Δημιουργία και παραμετροποίηση Spring Security:

- @EnableWebSecurity
- @Order (σε περίπτωση πολλαπλών layers αυθεντικοποίησης, εμείς θα έχουμε ένα layer)
- @Configuration

4.2.3.2 Δημιουργία Οντοτήτων

Κλάση Χρήστης:

- @id integer id;
- @column (name= "username") String username;
- @column (name= "password") String password;
- @ManyToOne @JoinColumn (name="rule_id_fk") Rule[] rules;
- @ManyToOne @JoinColumn (name="smart_device_id_fk") Smart Device[] smartDevices;
- @column (name= "age") integer age;
- @column (name= "gender") String gender;
- @column (name= "employment") String employment;

Κλάση Έξυπνης Συσκευής:

- @id integer id;
- @column (name= "settings") String settings;
- @column (name= "position") String position;
- @column (name= "connected") boolean connected;

Κλάση Κανόνων:

- @id integer id;
- @column (name= "name") String name;
- @ManyToOne @JoinColumn (name="smart_device_id_fk") Smart Device[] smartDevices;
- @column (name= "event") String event;
- @column (name= "action") String action;

4.2.3.3 Δημιουργία JPA Repository

Κάθε πίνακας δεδομένων στην βάση, θεωρείται για το Spring ένα Repository, με τη χρήση και την παραμετροποίηση του Repository μπορούμε να φτιάξουμε queries με την βάση και να πάρουμε τα δεδομένα που χρειαζόμαστε.

Default JPA Queries:

- count ()
- delete (Entity)
- deleteAll ()
- deleteById (Integer)
- exists (Entity)
- existsById (Integer)
- findAll ()
- findAllById (Iterable)
- findOne (Entity)
- flush ()
- save (Entity)
- saveAll (Iterable)
- saveAndFlush (Entity)

Τα παραπάνω queries είναι αρκετά για την χρήση των 3 βασικών requests, GET, POST, και DELETE.

Με τη δημιουργία μίας διεπαφής (Interface) και κληρονομώντας το JpaRepository Interface μπορούμε να δημιουργήσουμε τα τρία Repositories:

1. User Repository
2. Smart Device Repository
3. Rules Repository

4.2.3.4 Δημιουργία REST – API και expose δεδομένων σε endpoints

Έχουμε 3 συνολικά Entities , επομένως θα χρειαστούμε τα βασικά operations για το κάθε ένα:

1. GET /users - HTTP_STATUS_OK = 200
2. POST /users/add - HTTP_STATUS_OK = 200
3. DELETE /user/delete - HTTP_STATUS_OK = 200

Επίσης μπορούμε να χρησιμοποιήσουμε δικό μας Entity για τη χρήση custom HTTP απαντήσεων, οι προκαθορισμένες που μας δίνονται από το HTTP πρωτόκολλο είναι αρκετές.

Με τον παρακάτω πίνακα καλύπτουμε τις βασικές ανάγκες του back-end ώστε να μπορούμε να αποθηκεύουμε / ανανεώνουμε / προσθέτουμε / διαγράφουμε δεδομένα στην βάση (Πίνακας 3).

Type	Endpoint	Payload Required	JSON - Response	HTTP Success Status	HTTP Failed Status
GET	/users	-	User	200	500
POST	users/add	User	String	200	500
DELETE	users/delete	User	String	200	500/405
GET	/devices	-	SmartDevice	200	500
POST	devices/add	SmartDevice	String	200	500
DELETE	devices/delete	SmartDevice	String	200	500/405
GET	/rules	-	Rule	200	500
POST	rules/add	Rule	String	200	500
DELETE	rules/delete	Rule	String	200	500/405

Πίνακας 3. Βασικά endpoints για: α)χρήστες, β)έξυπνες συσκευές και γ)κανόνες.

Σημείωση: Εξ ορισμού, σε περίπτωση που στείλουμε χρήστη με το ίδιο username στο /users/add, το JPA Repository, ενημερώνει τον χρήστη με τα καινούργια στοιχεία στην βάση. Το ίδιο συμβαίνει εάν συμπεριλάβουμε τα ID του κανόνα (Rule) και έξυπνης συσκευής (SmartDevice).

Παραμετροποίηση του **application.properties** αρχείου

Σε αυτό το αρχείο μπορούμε να συμπεριλάβουμε αρκετές πληροφορίες κατά την εκκίνηση του server, πρόκειται για ένα configuration αρχείο που πρέπει να έχουμε συμπληρωμένο.

- spring.port - το port number που θα τρέχει ο server (**Προσοχή:** σε περιβάλλον παραγωγής το καθορίζει ο host)
- spring.datasource.url - url βάσης δεδομένων & σχήματος
- spring.datasource.username - όνομα χρήστη στην βάση
- spring.datasource.password - κωδικός χρήστη στη βάση
- loggin.file - αρχείο με τα logs του server

4.2.4 Αυθεντικοποίηση Χρηστών

Για την αυθεντικοποίηση χρηστών θα χρησιμοποιήσουμε το JWT Token. Έχοντας οι χρήστες το συγκεκριμένο token μπορούν να αλληλοεπιδράσουν με το σύστημα και να κάνουν αλλαγές εκεί που τους επιτρέπουμε εμείς.

Το Json-Web-Token περιλαμβάνεται στην επικεφαλίδα (header) των αιτήσεων προς στον server , καθώς ο ίδιος ο server είναι υπεύθυνος για την δημιουργία του και την αποθήκευσή του στην cache για ένα χρονικό διάστημα που έχουμε προγραμματίσει. Αξίζει να σημειωθεί ότι δεν πρόκειται για μια εξ ορισμού λειτουργία του Spring Security αλλά για ένα layer που φτιάχνουμε εμείς για την προστασία των

δεδομένων. Χρησιμοποιείται στα περισσότερα Production Environments υπό την μορφή key – value στην κεφαλίδα.

Παράδειγμα JWT στην κεφαλίδα

Key: **Authorization**

Value: **Bearer**

2ow7xKjp3x6fJy4cJcQQi2BYy_eGzcRjCjVetGHJuQ_vMWmpOwQz2XkHbvh0cr36C2HBNe0KcaBM7uQ4XDzz2qZnDgEERcws5_HwdaPJxRMQ39SlwJCXyiJ5wt8OJ_f9TqVWVK5zL458wZJoBRbN_C5VyA8dYAMLBjZQDwWTDG0

Μπορούμε να καθορίσουμε εμείς το **key** καθώς και το **value**, θα ήταν καλό να αλλάζαμε αυτές τις παραμέτρους σε περίπτωση που κάποιος επιτιθέμενος λάβει το πακέτο. Στην περίπτωση που γνωρίζει το Authorization key και το payload που στέλνουμε μπορεί να κάνει σοβαρή ζημιά στην βάση μέσω του server , για αυτό θα ήταν καλό να έχουμε μια λίστα με IP διευθύνσεις που μπορούν να επικοινωνήσουν με τον server (trusted IP addresses).

Ο τρόπος απόκτησης και ανάθεσης του token γίνεται κατά το login στην εφαρμογή. Στην περίπτωση που έχει γίνει σωστή αυθεντικοποίηση (σωστά username , password) τότε γίνεται η εξουσιοδότηση στον χρήστη να χρησιμοποιήσει το API μας.

Άλλοι τρόποι αυθεντικοποίησης είναι ο OAuth και OAuth2.0 που σε αυτές τις περιπτώσεις χρησιμοποιείται η επικοινωνία μεταξύ 2 servers για την εξουσιοδότηση του χρήστη.

Με τη χρήση του JWT ο πίνακας 3 που είχαμε παραπάνω ανανεώνεται καθώς πλέον περιμένουμε το token στην κεφαλίδα και προστίθεται και το **403 Status Code** (access denied) (Πίνακας 4).

Type	Endpoint	Payload Required	JSON - Response	Header	HTTP Success Status	HTTP Failed Status
GET	/users	-	User	JWT	200	500/403
POST	users/add	User	String	JWT	200	500/403
DELETE	users/delete	User	String	JWT	200	500/403/405
POST	users/login	Authentication Request	String (JWT)	-	200	500/405
GET	/devices	-	SmartDevice	JWT	200	500/403
POST	devices/add	SmartDevice	String	JWT	200	500/403
DELETE	devices/delete	SmartDevice	String	JWT	200	500/403/405
GET	/rules	-	Rule	JWT	200	500/403
POST	rules/add	Rule	String	JWT	200	500/403
DELETE	rules/delete	Rule	String	JWT	200	500/403/405

Πίνακας 4. Ανανεωμένα βασικά endpoints για: α)χρήστες ,β)έξυπνες συσκευές και γ)κανόνες.

Σημείωση: Προκύπτει και ένα νέο endpoint που είναι υπεύθυνο για την αυθεντικοποίηση και εξουσιοδότηση του χρήστη, το οποίο δεν περιμένει JWT στα headers. Ο χρήστης πρέπει να στείλει ένα AuthenticationRequest αντικείμενο το οποίο αποτελείται από το username και το password.

4.2.5 Επικοινωνία με Έξυπνες Συσκευές

Για την επικοινωνία των έξυπνων συσκευών αλλά και για τον έλεγχο της δράσης του κανόνα θα πρέπει να δημιουργηθούν ροές στον server. Κάνοντας subscribe σε αυτά τα endpoints μπορούμε να παίρνουμε τα

δεδομένα που στέλνουν οι έξυπνες συσκευές στην ροή καθώς και να ελέγχουμε ανά χρονικά διαστήματα την εγκυρότητα του κανόνα.

4.2.5.1 Αναδραστικός Ιστός (Reactive Web)

Το Spring Framework έχει την βιβλιοθήκη Spring Reactive με την οποία μας δίνεται η δυνατότητα να δημιουργούμε ροές. Θα χρειαστούμε ζευγάρια από endpoints για να στείλουμε και να λάβουμε δεδομένα. Το ένα endpoint θα το ονομάσουμε **send** και το άλλο **receive**. Το **send** είναι υπεύθυνο για να την επεξεργασία των δεδομένων και την αποστολή στην ροή καθώς το **receive** είναι υπεύθυνο για να δώσει στον χρήστη την δυνατότητα να λάβει δεδομένα.

Για να πετύχουμε την παραπάνω συσχέτιση των endpoints θα πρέπει με κάποιο τρόπο να συμπεριλάβουμε κάτι κοινό. Για αυτό τον σκοπό χρησιμοποιείται ο **MessageProcessor** (επεξεργαστής-καταχωρητής δεδομένων). Είναι υπεύθυνος για την προσπέλαση, καταγραφή και επιστροφή των δεδομένων προσωρινά. Χρησιμοποιεί διεπαφές τύπου Consumer<Object> όπου έχει δημιουργηθεί με τη χρήση του “Λειτουργικού Προγραμματισμού” (Functional Programming) καθώς ο Consumer λειτουργεί μόνο όταν υπάρχει μια ενέργεια.

Με τον συνδυασμό της κλάσης Flux<T> η οποία είναι υπεύθυνη για την δημιουργία ροών (sink) μπορούμε να διαθέσουμε τα δεδομένα μέσω μιας προσπέλασης της λίστας των consumers που περιέχει ο processor.

Θέτοντας το τύπο του endpoint σε **MediaType.TEXT_EVENT_STREAM_VALUE** ώστε να γνωρίζει ο server ότι πρόκειται για ροή και επιστρέφοντας αντικείμενα τύπου Flux<T> επιτυγχάνουμε την αποστολή και προσπέλαση δεδομένων μέσω της ροής. Όπου <T> είναι ένα δικό μας template, θα μπορούσε να είναι είτε Object είτε List είτε οποιοδήποτε αντικείμενο στην Java (Πίνακας 5).

Θα χρειαστούμε:

- Send & Receive endpoints για τις έξυπνες συσκευές (1 κοινός επεξεργαστής (processor))
- Send & Receive endpoints για τους κανόνες (1 κοινός επεξεργαστής (processor))

Type	Endpoint	Payload Required	JSON -Response	Header	HTTP Success Status	HTTP Failed Status
Post	devices/send	SmartDevice	String	JWT	200	500/403/405
Get	devices/receive	-	Flux<SmartDevice>	JWT	200	500/403/405
Post	rules/send	SmartDevice	String	JWT	200	500/403/405
Get	rules/receive	-	Flux<Rule>	JWT	200	500/403/405

Πίνακας 5. Streaming endpoints του server.

Για την επεξεργασία του κανόνα και την εγκυρότητα του action θα πρέπει να φτιαχτεί μια κλάση με μια μέθοδο η οποία να συμπεριλαμβάνει το annotation @Bean. Τα @Beans ενεργοποιούνται αυτόματα στην εκκίνηση του sever. Εκεί θα πρέπει να γίνεται ο έλεγχος των αποθηκευμένων κανόνων και σε περίπτωση εγκυρότητας του event να σταλεί στον processor που είναι υπεύθυνος για τους κανόνες. Έτσι ο χρήστης θα λάβει τον κανόνα μόλις ισχύσει το event.

4.2.5.2 REST Template – Εξωτερικές κλήσεις API

Σε ένα περιβάλλον παραγωγής η @Bean μέθοδος για τον έλεγχο εγκυρότητας του event θα πρέπει να κάνει εξωτερικές κλήσεις σε όλα τα αναγκαία API για την συλλογή πληροφοριών. Επομένως ο server μας, για

κάποιες διεργασίες όπως αυτή θα πρέπει να γίνει ο Client (πελάτης) για να επικοινωνήσει με εξωτερικούς server. Το Spring Framework, έχει καλύψει και αυτή την ανάγκη καθώς μας δίνει πρόσβαση σε μια κλάση με το όνομα RestTemplate. Με τη χρήση του Rest Template μπορούμε να φτιάξουμε όλα τα είδη των requests για την εξωτερική επικοινωνία (Πίνακας 6). Η εξωτερική επικοινωνία είναι αναγκαία καθώς χρειαζόμαστε ζωντανή και έγκυρη πληροφορία για την ενημέρωση των έξυπνων συσκευών.

Ένα παράδειγμα είναι η παροχή όλων των καναλιών από την έξυπνη τηλεόραση. Θα πρέπει ο server να είναι σε θέση να παρέχει όλες τις λίστες που υπάρχουν με όλα τα διαθέσιμα κρατικά / ιδιωτικά κανάλια, πράγμα το οποίο μπορεί να καθοριστεί από πολύ απλό έως πολύ δύσκολο καθώς εμπλέκονται και παράγοντες που δεν αφορούν αποκλειστικά το προγραμματιστικό κομμάτι [10].

Type	Endpoint	Payload Required	JSON -Response	Header	HTTP Success Status	HTTP Failed Status
GET	/users	-	User	JWT	200	500/403
POST	users/add	User	String	JWT	200	500/403
DELETE	users/delete	User	String	JWT	200	500/403/405
POST	users/login	Authentication Request	String (JWT)	-	200	500/405
GET	/devices	-	SmartDevice	JWT	200	500/403
POST	devices/add	SmartDevice	String	JWT	200	500/403
DELETE	devices/delete	SmartDevice	String	JWT	200	500/403/405
POST	devices/send	SmartDevice	String	JWT	200	500/403/405
GET	devices/receive	-	Flux<SmartDevice>	JWT	200	500/403/405
GET	/rules	-	Rule	JWT	200	500/403
POST	rules/add	Rule	String	JWT	200	500/403
DELETE	rules/delete	Rule	String	JWT	200	500/403/405
POST	rules/send	SmartDevice	String	JWT	200	500/403/405
GET	rules/receive	-	Flux<Rule>	JWT	200	500/403/405

Πίνακας 6. Τελικός Πίνακας Back-End Server.

- **Κόκκινο:** αποστολή σε ροή ή παραλαβή δεδομένων ροής
- **Πράσινο:** παραλαβή δεδομένων
- **Μωβ:** διαγραφή δεδομένων
- **Μπλε:** αποστολή δεδομένων

4.3 Εφαρμογή Android

4.3.1 Εισαγωγή

Η εφαρμογή που κληθήκαμε να υλοποιήσουμε έγινε με την χρήση του Android Studio (API 29 : Android 10.0 , min SDK version : 27, target SDK version : 29) και η κύρια ιδέα είναι η δημιουργία μιας όσο το δυνατόν πιο φιλικής διεπαφής προς τους χρήστες για την δημιουργία και επεξεργασία κανόνων στο τομέα του Internet of Things. Οι κανόνες περιλαμβάνουν τις παραμετροποιήσεις των έξυπνων συσκευών που έχουν ορίσει οι χρήστες σ' ένα σύνολο έξυπνων συσκευών και τις λογικές συσχετίσεις αυτών που έχουν ως

αποτέλεσμα μία ή περισσότερες δράσεις. Χωρίζονται σε δύο κύρια μέρη : α) *Συμβάν (Event)*, β) *Δράση (Action)*.

- **Event:** Το σύνολο των ενεργειών που μπορούν να ικανοποιηθούν από τις έξυπνες συσκευές καθώς και η λογική συσχέτισή τους.
- **Action:** Το σύνολο των αποτελεσμάτων ικανοποίησης του Event.

Κύριο μέλημα της συγκεκριμένης εφαρμογής είναι η δημιουργία των κανόνων να προσφέρεται μέσω μιας φιλικής διεπαφής.

Όπως αναφέρεται και παραπάνω στο κομμάτι του συμβάντος (event) υπάρχουν λογικές συσχετίσεις, αναλόγως το πλήθος των έξυπνων συσκευών του κανόνα στο κομμάτι του συμβάντος. Σε περίπτωση που υπάρχει **μόνο** μια έξυπνη συσκευή τότε **δεν** υπάρχει λογική συσχέτιση. Όμως, σε αντίθετη περίπτωση που υπάρχουν δύο ή περισσότερες έξυπνες συσκευές τότε σ' αυτές μεταξύ θα πρέπει να οριστεί λογική συσχέτιση από τους χρήστες. Η εφαρμογή μας έχει δύο εκ των βασικών λογικών συσχετίσεων, αυτές είναι:

- **AND:** Δύο ή περισσότερες έξυπνες συσκευές που συνδέονται μεταξύ τους με το λογικό «AND» θα πρέπει οι ενέργειες αυτών να ισχύουν, οπωσδήποτε, ώστε να πραγματοποιηθεί η επιθυμητή δράση. Εστω και μία ενεργεία μιας έξυπνης συσκευής να μην ισχύσει τότε η δράση δεν πραγματοποιείται.
- **OR:** Δύο ή περισσότερες έξυπνες συσκευές που συνδέονται μεταξύ τους με το λογικό «OR» θα πρέπει έστω και μία ενέργεια από μία έξυπνη συσκευή να ισχύσει ώστε να πραγματοποιηθεί η επιθυμητή δράση. Για να μην πραγματοποιηθεί η δράση θα πρέπει όλες οι ενέργειες κάθε έξυπνης συσκευής να μην ισχύουν.

Επιπροσθέτως, στο κομμάτι της δράσης (action) **δεν** υπάρχουν λογικές συσχετίσεις.

Υπάρχουν έξι (6) κύριες επιλογές με τις οποίες οι χρήστες μπορούν να αλληλεπιδράσουν με την εφαρμογή.

- i. **Editor Domino:** Οι έξυπνες συσκευές αναπαρίστανται ως dominos για την δημιουργία κανόνων. Οι χρήστες μπορούν να οραματιστούν τις θέσεις των domino για την σύνθεση ενός κανόνα.
- ii. **Editor E-Wizard:** Οι έξυπνες συσκευές αναπαρίστανται ως εικονίδια που παραπέμπουν στην πραγματική τους μορφή. Ακολουθώντας τα κατάλληλα βήματα οι χρήστες μπορούν να συνθέσουν κανόνες.
- iii. **Κάτοψη (Floorspan):** Ορίζοντας την κάτοψη της περιοχής που υπάρχουν οι έξυπνες συσκευές του χρήστη δίνεται η δυνατότητα μέσω του συγκεκριμένου εργαλείου να προστεθούν οι έξυπνες συσκευές (αναπαρίστανται ως domino) πάνω σ' αυτή την περιοχή.
- iv. **Αποθηκευμένοι Κανόνες:** Απεικονίζονται όλοι οι κανόνες που έχουν αποθηκεύσει οι χρήστες σύμφωνα με τα ονόματα που έχουν δώσει. Με βάσει τον editor δημιουργίας του κανόνα απεικονίζεται και το αντίστοιχο εικονίδιο αναφερόμενο είτε στον editor Domino είτε στον editor E-Wizard.
- v. **Εκπαίδευση στους Editors:** Αναλυτική καθοδήγηση των χρηστών για την λειτουργία των δύο (2) editor.
- vi. **Πληροφορίες έξυπνων συσκευών:** Παρουσίαση των έξυπνων συσκευών και των παραμετροποιήσεων που μπορούν να δεχθούν.

Ολοκληρώνοντας την εισαγωγή, θα πρέπει να αναφερθεί ότι, για να προσεγγίσουμε όσο τον δυνατόν περισσότερο την προσαρμοστικότητα του γραφικού περιβάλλοντος της εφαρμογής ανάλογα με την οθόνη που διαθέτει η συσκευή που χρησιμοποιεί την εφαρμογή χρησιμοποιήσαμε layouts που μας δίνονται από το Android γι' αυτό το σκοπό. Έγινε κατά κύριο λόγο χρήση του *Relative Layout* και του *Constraint Layout*. Στα *Relative Layouts* μπορούμε να χρησιμοποιήσουμε κάποια tags έτσι ώστε τα γραφικά στοιχεία της εφαρμογής να έχουν συγκεκριμένη σειρά εμφάνισης. Επιπροσθέτως, έγινε χρήση των margins για να τηρούνται οι αποστάσεις των στοιχείων κατά τον ίδιο συντελεστή. Στο *Constraint Layout* ορίζονται το πολύ τέσσερα βασικά constraints (περιορισμοί) τα οποία έχουν την ιδιότητα να προσδιορίζουν την θέση δυναμικά στο γραφικό περιβάλλον. Για την αποφυγή διαρροών των γραφικών πληροφοριών σε συσκευές χρησιμοποιήσαμε το οριζόντιο και κάθετο γραφικό στοιχείο που ονομάζεται *ScrollView*.

4.3.2 Editor Domino

Ο συγκεκριμένος editor προήλθε από την ιδέα του ομώνυμου παιχνιδιού Domino. Ένας τρόπος του συγκεκριμένου παιχνιδιού είναι η τοποθέτηση των dominos σε σειρά ή/και παράλληλα με στόχο την πτώση όλων των dominos, από το πρώτο μέχρι το τελευταίο και των ενδιάμεσων, αλλά κυρίως του τελευταίου domino.

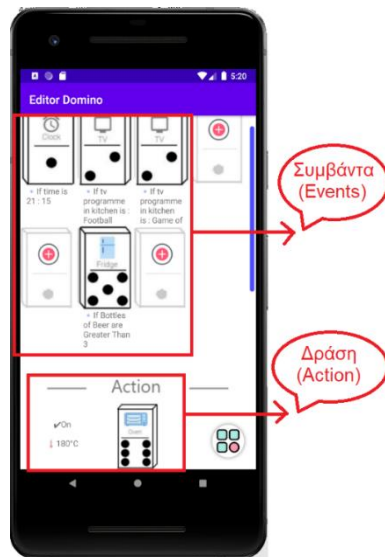
Λαμβάνοντας υπόψιν τα παραπάνω και σκεπτόμενοι ότι οι έξυπνες συσκευές μπορούν να απεικονιστούν ως dominos (Εικόνα 13), χρήση των dominos ως *metaphor*, μπορεί να γίνει ο διαχωρισμός μεταξύ συμβάντος (event) και δράσης (action). Ως αποτέλεσμα μπορούν να δημιουργηθούν κανόνες γι' αυτό το λόγο καταλήξαμε στην δημιουργία του συγκεκριμένου editor.



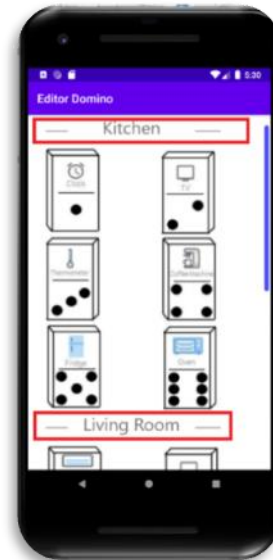
Εικόνα 13. Έξυπνες συσκευές απεικονισμένες ως Dominos.

Το domino που βρίσκεται στο τέλος ορίζεται να είναι η δράση και όλα τα προηγούμενα dominos ορίζονται να είναι τα συμβάντα (Εικόνα 14). Επίσης, υπάρχει διαχωρισμός των έξυπνων συσκευών με βάση το που

βρίσκονται στο εσωτερικό της περιοχής. Για παράδειγμα, αν οι έξυπνες συσκευές βρίσκονται σε μια κατοικία τότε ο έξυπνος φούρνος εμφανίζεται στην εφαρμογή ότι βρίσκεται στην κουζίνα. Αντίστοιχα και για τις υπόλοιπες συσκευές (Εικόνα 15).

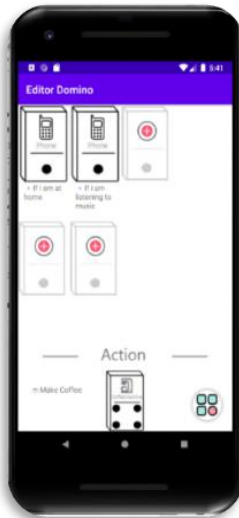


Εικόνα 14. Διαχωρισμός συμβάντων - δράσης.

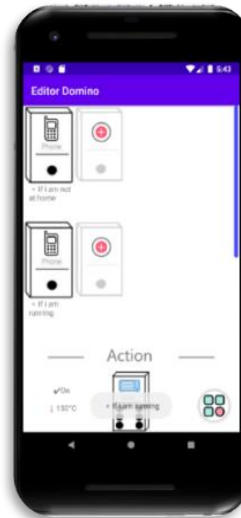


Εικόνα 15. Διαχωρισμός έξυπνων συσκευών με βάση την θέση που βρίσκονται στην περιοχή την οποία βρίσκονται.

Σύμφωνα με την τοποθέτηση των dominos στο κομμάτι των συμβάντων γίνεται αντιληπτό αν πρόκειται για λογική σχέση «OR» ή/και «AND». Σε περίπτωση που υπάρχει ένα μόνο domino στο κομμάτι των συμβάντων τότε είναι προφανές ότι δεν υπάρχει λογική σχέση. Όσον αφορά τη λογική σχέση «OR», ο χρήστης έχει τη δυνατότητα να την εφαρμόσει, μεταξύ δύο dominos ή περισσότερων, τοποθετώντας τα οριζόντια πάνω στο πλέγμα του editor (Εικόνα 16). Από την άλλη μεριά, η λογική σχέση «AND» μπορεί να πραγματοποιηθεί κατά την ίδια λογική με παραπάνω αλλά τοποθετώντας τα dominos κάθετα πάνω στο πλέγμα (Εικόνα 17).



Εικόνα 16. Δημιουργία λογικής σχέσης OR.

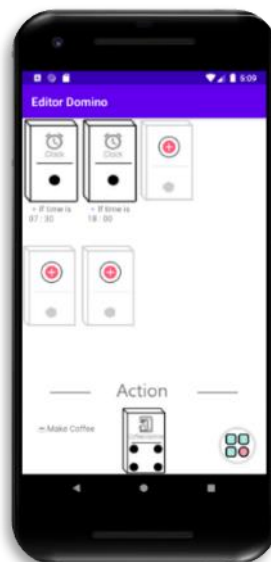


Εικόνα 17. Δημιουργία λογικής σχέσης

Λογικό OR και AND

4.3.2.1 Λογικό OR

Είναι γνωστό ότι στην λογική σχέση «OR» δεν είναι απαραίτητο να ισχύουν όλες οι συνθήκες που συνδέονται με το λογικό «OR», αρκεί και η μια να είναι αληθής ώστε να ισχύσει. Έστω ότι, έχουμε τον παρακάτω απλό κανόνα όπου στο κομμάτι του συμβάντος είναι το έξυπνο ρολόι το οποίο έχει τοποθετηθεί δύο φορές πάνω στο πλέγμα. Η πρώτη του τιμή είναι 7:30 και η δεύτερη 18:00. Στο κομμάτι της δράσης είναι η έξυπνη καφετιέρα όπου έχει ρυθμιστεί έτσι ώστε να φτιάξει καφέ (Εικόνα 18).



Εικόνα 18. Δημιουργία λογικής σχέσης OR.

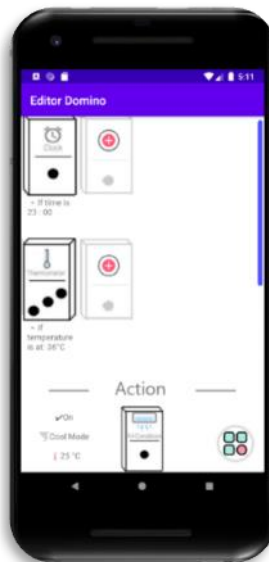
Σύμφωνα με την τοποθέτηση των δύο (2) dominos στο κομμάτι των συμβάντων, τα οποία είναι οριζόντια-παράλληλα, γίνεται αντιληπτό ότι υπάρχουν τρεις (3) περιπτώσεις ώστε να πραγματοποιηθεί η δράση :

1. Να ισχύσει μόνο το πρώτο domino που έχει ρυθμιστεί να γίνει η ρίψη του στις 7:30.
2. Να ισχύσει μόνο το δεύτερο domino που έχει ρυθμιστεί να γίνει η ρίψη του στις 18:00.
3. Να ισχύσουν και τα δύο.

Δηλαδή, είτε πέσει το πρώτο domino ρολόι είτε το δεύτερο είτε και τα δύο η δράση θα πραγματοποιηθεί. Για να μην πραγματοποιηθεί η δράση θα πρέπει να μην ισχύουν και οι δύο ρυθμίσεις από τις συσκευές. Κατ' αυτόν τον τρόπο επιτυγχάνουμε να έχουμε τη λογική σχέση «OR» μεταξύ δύο ή/και περισσότερων έξυπνων συσκευών.

4.3.2.2 Λογικό AND

Είναι γνωστό στην λογική σχέση «AND» είναι απαραίτητο να ισχύουν όλες οι συνθήκες που συνδέονται με το λογικό «AND» ώστε να είναι αληθής. Έστω ότι, δημιουργείται κανόνας ο οποίος περιλαμβάνει στο κομμάτι των συμβάντων δύο έξυπνες συσκευές: α) έξυπνο ρολόι, β) έξυπνο θερμόμετρο. Η πρώτη έξυπνη συσκευή ρυθμισμένη έτσι ώστε να «πέσει» στις 23:00 και η δεύτερη ρυθμισμένη να «πέσει» όταν η θερμοκρασία φτάσει τους 36 °C. Οι παραπάνω δύο συσκευές συνδεδεμένες μεταξύ τους με λογικό «AND». Ως δράση έχει οριστεί το έξυπνο κλιματιστικό ρυθμισμένο έτσι ώστε να ανοίξει στους 25 °C και στον τρόπο λειτουργίας κρύο (Εικόνα 19).



Εικόνα 19. Δημιουργία λογικής σχέσης AND.

Σύμφωνα με την τοποθέτηση των δύο (2) dominos στο κομμάτι των συμβάντων, τα οποία είναι κάθετα, γίνεται αντιληπτό ότι υπάρχει μόνο μία περίπτωση ώστε να πραγματοποιηθεί η δράση:

Να ισχύσει το πρώτο domino που έχει ρυθμιστεί να «πέσει» στις 23:00 και εκείνη την ώρα η θερμοκρασία να έχει φτάσει τους 36 °C, δηλαδή να ισχύσει και το δεύτερο domino. Σε περίπτωση που ισχύει η συνθήκη (ρύθμιση) ενός μόνο εκ των δύο η δράση δεν πραγματοποιείται.

Κατ' αυτόν τον τρόπο επιτυγχάνουμε να έχουμε την λογική σχέση «AND» μεταξύ δύο ή/και περισσότερων έξυπνων συσκευών.

4.3.2.3 Δημιουργία Κανόνα στον Editor Domino

Το πρώτο βήμα για την δημιουργία κανόνα στον editor Domino είναι ο ορισμός της δράσης (action). Για τον ορισμό της δράσης, οι χρήστες πρέπει να επιλέξουν το βοηθητικό domino που ονομάζεται “shadow” (Εικόνα 20) το οποίο θα τους οδηγήσει στο μενού όπου υπάρχουν όλες οι έξυπνες συσκευές οι οποίες είναι διαθέσιμες. Επιλέγοντας, αρχικά, την έξυπνη συσκευή και, έπειτα, ορίζοντας το επιθυμητό αποτέλεσμα, δηλαδή ποια θα είναι η τελική ενέργεια της έξυπνης συσκευής (Εικόνα 21). Αφού την ορίσουν, αριστερά από την έξυπνη συσκευή θα εμφανιστούν οι επιλεγμένες ρυθμίσεις σε απλό κείμενο (Εικόνα 22).



Εικόνα 20. Βοηθητικό domino "shadow" για ορισμό δράσης.



Εικόνα 21. Εμφάνιση δυνατών ρυθμίσεων της έξυπνης συσκευής μέσω pop-up μηνύματος.



Εικόνα 22. Εμφάνιση επιλεγμένων ρυθμίσεων της έξυπνης συσκευής στα αριστερά.

Κατά αυτόν τον τρόπο επιτυγχάνεται ο ορισμός της δράσης (action) στον Editor Domino.

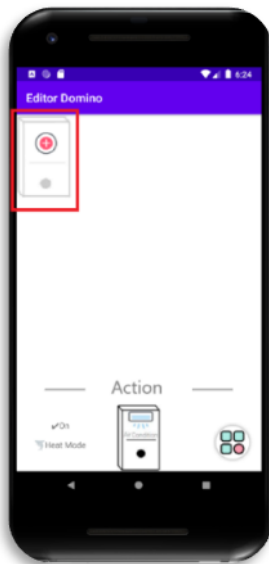
Το δεύτερο βήμα είναι η επιλογή της/των έξυπνης/ων συσκευής/ων που θα αποτελέσουν το/τα συμβάν/τα κομμάτι του κανόνα. Αρχικά, οι χρήστες τοποθετούν και ρυθμίζουν το εναρκτήριο domino

που εμφανίζεται στο πάνω αριστερό σημείο του πλέγματος μέσω του βοηθητικού domino “shadow” (Εικόνα 23, Εικόνα 24). Αφού προσθέσουν το πρώτο domino στο κάτω μέρος του θα εμφανιστεί η ρύθμιση που πραγματοποίησαν σε απλό κείμενο (Εικόνα 25). Στην συνέχεια, εάν και εφ’ όσον το επιθυμούν, έχουν δύο επιλογές για να τοποθετήσουν το επόμενο domino :

- Κάθεται
- Οριζόντια

μέσω των αντίστοιχων βοηθητικών dominos “shadow” (Εικόνα 26).

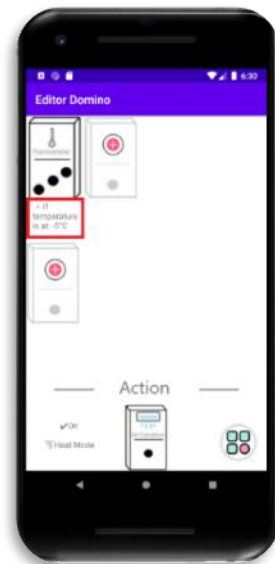
Όπως αναφέραμε και προηγουμένως, αν το domino τοποθετηθεί κάθετα μεταφράζεται ως λογική σχέση «AND» με το ακριβώς από πάνω του και από κάτω του, εάν και εφ’ όσον υπάρχει domino πάνω ή/και κάτω του. Αν το domino τοποθετηθεί οριζόντια μεταφράζεται ως λογική σχέση «OR» με το ακριβώς δεξιά και αριστερά του, εάν και εφ’ όσον υπάρχει domino δεξιά ή/και αριστερά του.



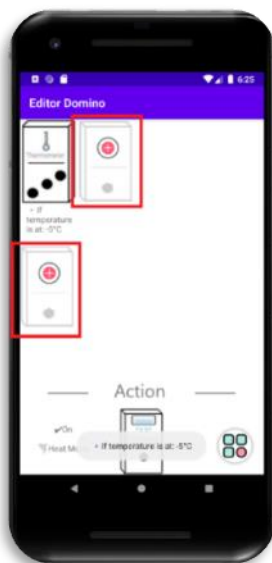
Εικόνα 23. Εναρκτήριο βοηθητικό domino “shadow” στα συμβάντα.



Εικόνα 24. Εμφάνιση Pop Up μηνύματος για ρύθμιση της συσκευής.

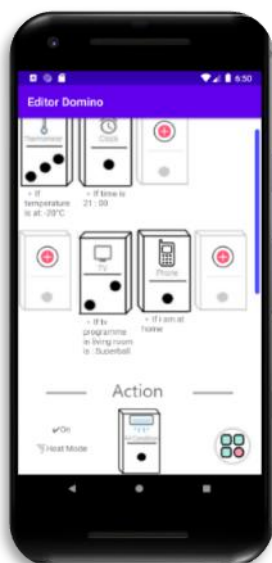


Εικόνα 25. Εμφάνιση της επιλεγμένης έξυπνης συσκευής και της ρύθμισης που εισήγαγε ο χρήστης στο κάτω μέρος.



Εικόνα 26. Εμφάνιση δύο βοηθητικών dominos "shadow" για την προσθήκη του επόμενου domino.

Για να ολοκληρωθεί ο κανόνας οι χρήστες είναι αυτοί που θα καθορίσουν το πότε θα τον θεωρήσουν «ολοκληρωμένο» σύμφωνα με τις έξυπνες συσκευές που έχουν προσθέσει, τον τρόπο που τις έχουν τοποθετήσει αλλά και την επιθυμία που θα ήθελαν να εκπληρώσουν μέσω του κανόνα. Κάθε φορά που προστίθεται ένα domino, τόσο δεξιά αυτού όσο και κάτω του θα εισέρχεται αυτόματα βοηθητικό domino “shadow” (Εικόνα 27).



Εικόνα 27. Κανόνας έπειτα από αρκετές προσθήκες έξυπνων συσκευών.

Βοηθητικό Domino “Shadow”

Το domino “shadow” παίζει καθοριστικό ρόλο για την δημιουργία του κανόνα. Σύμφωνα μ’ αυτό οι χρήστες προχωράνε την διαδικασία ολοκλήρωσης δημιουργίας του κανόνα. Δεν πρόκειται για τίποτα άλλο πέρα από ένα «κενό» domino το οποίο δεν αναπαριστά καμία έξυπνη συσκευή αλλά πρόκειται να αναπαραστήσει είτε στο κομμάτι του συμβάντος είτε στο κομμάτι της δράσης.

4.3.2.4 Διαχείριση Κανόνα

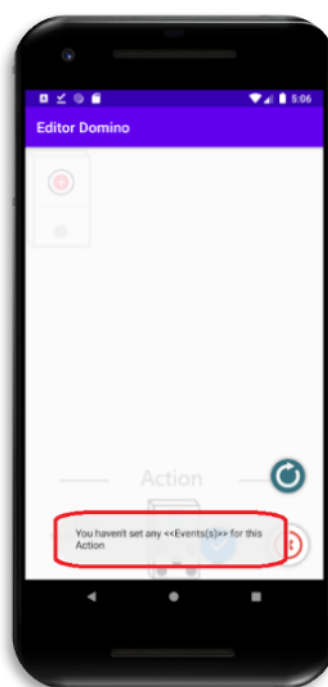
Όταν ο χρήστης βρίσκεται στη διαδικασία ανάπτυξης του κανόνα πραγματοποιείται έλεγχος αν τηρούνται κάποιες προϋποθέσεις ώστε ο κανόνας να θεωρείται έγκυρος προς αποθήκευση. Οι προϋποθέσεις είναι οι εξής :

1. Έχει οριστεί η δράση (action).
2. Έχει οριστεί, τουλάχιστον, μία έξυπνη συσκευή στο κομμάτι του συμβάντος (event).

Σε περίπτωση που δεν ισχύουν τα δύο παραπάνω τότε δεν μπορεί να γίνει αποθήκευση του κανόνα και οι χρήστες ενημερώνονται με κατάλληλο μήνυμα (Εικόνα 28, Εικόνα 29).



Εικόνα 28. Εμφάνιση μηνύματος σε περίπτωση που ο χρήστης δεν έχει ορίσει δράση.



Εικόνα 29. Εμφάνιση μηνύματος σε περίπτωση που ο χρήστης δεν έχει ορίσει, τουλάχιστον, ένα συμβάν.

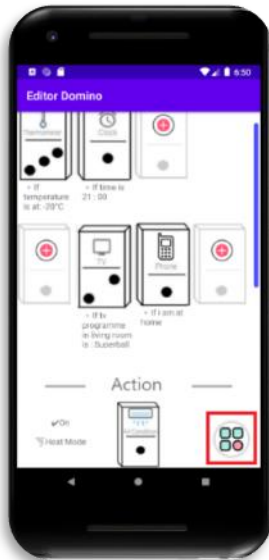
Με τον ίδιο τρόπο γίνεται και η διαχείριση των κανόνων στον Editor E-Wizard.

4.3.2.5 Αποθήκευση και Επανεκκίνηση

Αποθήκευση

Για να ολοκληρωθεί επιτυχώς η διαδικασία της δημιουργίας του κανόνα οι χρήστες θα πρέπει να τον αποθηκεύσουν. Η αποθήκευση επιτυγχάνεται μέσω του κουμπιού που βρίσκεται στο κάτω δεξιά μέρος της

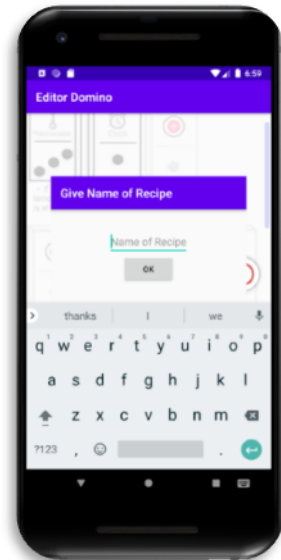
οθόνης και έπειτα επιλέγοντας το αριστερό κουμπί από τα δύο που εμφανίζονται (Εικόνα 30, Εικόνα 31). Οι χρήστες πληκτρολογούν το όνομα του κανόνα και έπειτα ο κανόνας αποθηκεύεται στη λίστα και διατηρείται μέχρι ο χρήστης να αποφασίσει να τον διαγράψει (Εικόνα 32, Εικόνα 33). Πατώντας πάνω στον αποθηκευμένο, πλέον, κανόνα οι χρήστες έχουν την δυνατότητα να δουν σε απλό κείμενο τον κανόνα καθώς και σε κινούμενο σχέδιο (Εικόνα 34).



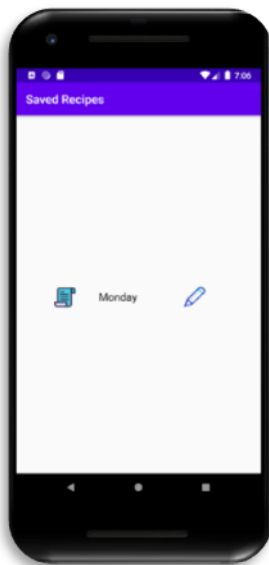
Εικόνα 30. Κουμπί το οποίο πατώντας το θα μας οδηγήσει στις επιλογές :1) Αποθήκευση , 2) Επανεκκίνηση.



**Εικόνα 31. Εμφάνιση των επιλογών :
1)Αποθήκευση,
2)Ανανέωση.**



Εικόνα 32. Επιλογή Αποθήκευσης και πληκτρολόγηση ονόματος του κανόνα.



Εικόνα 33. Εμφάνιση του κανόνα στην λίστα με τους αποθηκευμένους κανόνες.



Εικόνα 34. Εμφάνιση του κανόνα σε απλό κείμενο και σε κινούμενο σχέδιο εφ' όσον πατήσει το κουμπί με όνομα "Animation".

Επανεκκίνηση

Πέρα απ' τη δυνατότητα που δίνεται μέσω του πατήματος του κουμπιού που βρίσκεται κάτω δεξιά στην οθόνη στον editor domino (Εικόνα 21) δίνεται και η δυνατότητα της επανεκκίνησης του editor. Επανεκκίνηση σημαίνει η πλήρης διαγραφή οτιδήποτε υπάρχει τόσο στο κομμάτι συμβάντος όσο και στο κομμάτι της δράσης. Ουσιαστικά είναι μια επανεκκίνηση του editor Domino (Εικόνα 35).



Εικόνα 35. Επανεκκίνηση Editor Domino.

Με τον ακριβώς ίδιο τρόπο γίνονται οι λειτουργίες της αποθήκευσης και επανεκκίνησης στον Editor E-Wizard.

4.3.2.6 Ανανέωση και Διαγραφή Έξυπνης Συσκευής

Ανανέωση

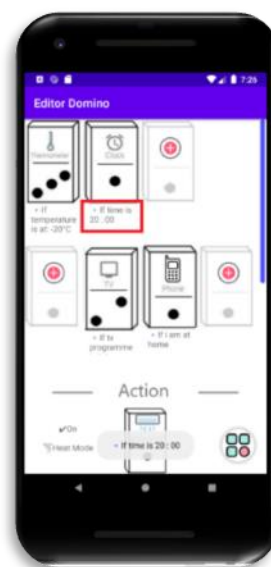
Σε περίπτωση που οι χρήστες, εφ' όσον έχουν προσθέσει δράση και ένα ή περισσότερα domino στο κομμάτι του συμβάντος, επιθυμούν να αλλάξουν την ρύθμιση σε κάποιο απ' αυτά ή ακόμα και σ' όλα τότε αρκεί ένα «κλικ» πάνω στη domino-συσκευή (Εικόνα 36). Έπειτα, θα τους εμφανιστεί το κατάλληλο pop-up παράθυρο (με τις προηγούμενες αποθηκευμένες ρυθμίσεις της συγκεκριμένης έξυπνης συσκευής) και οι χρήστες έχουν την δυνατότητα να τις αλλάξουν με τον ίδιο τρόπο που τις όρισαν την πρώτη φορά. Έστω ότι έχουμε τον κανόνα της εικόνας 18 και επιθυμούμε να αλλάξουμε την ρύθμιση του έξυπνου ρολογιού από 21:00 σε 20:00. Κάνοντας «κλικ» πάνω του θα μας εμφανιστούν οι προηγούμενες αποθηκευμένες ρυθμίσεις, δηλαδή 21:00 (Εικόνα 37). Ύστερα, γίνεται ενημέρωση της ρύθμισης και ορίζεται στις 20:00 (Εικόνα 38). Κατ' αυτό τον τρόπο επιτυγχάνεται η ενημέρωση μιας έξυπνης συσκευής στον editor Domino.



Εικόνα 36. Προηγούμενες ρυθμίσεις του έξυπνου ρολογιού.



Εικόνα 37. Αλλαγή της ρύθμισης από 21:00 σε 20:00.

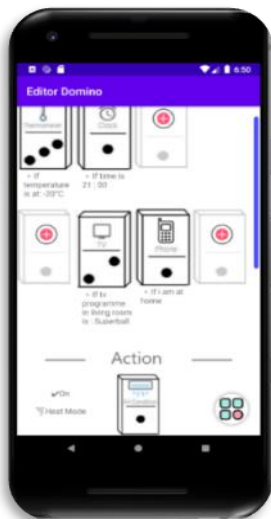


Εικόνα 38. Εμφάνισης της νέας λεξάντας έπειτα από την ενημέρωση που δέχθηκε το έξυπνο ρολόι.

Διαγραφή

Πέρα απ' την δυνατότητα της ενημέρωσης, δίνεται και η δυνατότητα της διαγραφής στο κομμάτι του συμβάντος. Προσφέρεται η δυνατότητα στους χρήστες εφ' όσον επιλέξουν μια έξυπνη συσκευή απ' το

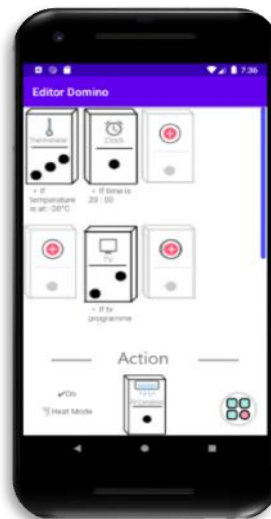
μενού, να την διαγράψουν απ' το κομμάτι του συμβάντος. Η δυνατότητα αυτή δίνεται κάνοντας παρατεταμένα «κλικ» στην έξυπνη συσκευή η οποία προστέθηκε. Η διαγραφή πραγματοποιείται μόνο για την τελευταία συσκευή που προστέθηκε στο κομμάτι του συμβάντος (Εικόνα 39, Εικόνα 40, Εικόνα 41).



Εικόνα 39. Διαγραφή του domino που αναπαριστάνει το "Phone".



Εικόνα 40. Εμφάνιση pop-up μηνύματος για ολοκλήρωση της διαγραφής.



Εικόνα 41. Ο κανόνας της Εικόνας 26 μετά την διαγραφή του domino "Phone".

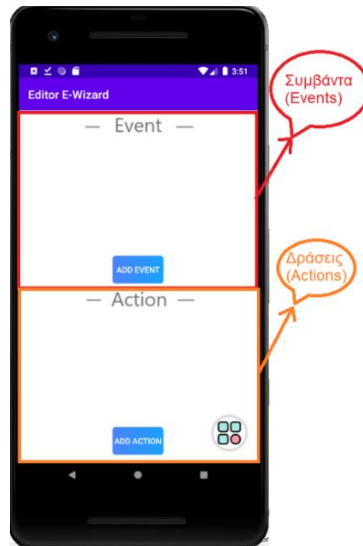
4.3.3 Editor E - Wizard

Ο Editor E-Wizard πρόκειται για έναν κατευθυντήριο οδηγό για τη διεκπεραίωση βημάτων με σκοπό την δημιουργία κανόνα. Παρακάτω αναπαρίστανται οι έξυπνες συσκευές ως εικονίδια που τις αντιπροσωπεύουν (Εικόνα 42).



Εικόνα 42. Έξυπνες συσκευές απεικονισμένες ως εικονίδια.

Έτσι η κάθε συσκευή είναι κοντά με την πραγματική της μορφή ώστε ο χρήστης να μπορεί να αντιληφθεί εύκολα τις συσκευές. Στην δική μας προσέγγιση η ακολουθία βημάτων για την δημιουργία ενός κανόνα έχει χωριστεί σε δυο μέρη: α) Συμβάν (Event) και β) Δράση (Action). Όλες οι προσθήκες στην λίστα με τις δράσεις συνδέονται με το λογικό AND καθώς δεν γίνεται να υπάρξει κανόνας που ως αποτέλεσμα να υπάρξει μια κατάσταση ασάφειας (π.χ. Αν $\diamond$ Τότε $\diamond$ ή $\diamond$). Γραφικά ο E-Wizard έχει απεικονιστεί-σχεδιαστεί με μια πολύ απλή μορφή ώστε να είναι εύκολη η χρήση του (Εικόνα 43).



Εικόνα 43. Διαχωρισμός συμβάντων – δράσεων Editor E - Wizard.

4.3.3.1 Λογικό OR και AND

Στον Editor Domino ο τρόπος ορισμού του λογικού «OR» γινόταν μέσω της τοποθέτησης μίας έξυπνης συσκευής στην ίδια γραμμή του πλέγματος με την άλλη. Επίσης, ο ορισμός του λογικού «AND» πραγματοποιούνταν μέσω της τοποθέτησης μίας έξυπνης συσκευής στην ίδια στήλη του πλέγματος με την άλλη. Στον Editor E-Wizard ο τρόπος ορισμού του λογικού «OR» και «AND» διαφέρει. Και οι δύο αυτοί λογικοί τελεστές προστίθενται στο κομμάτι του συμβάντος μέσω ενός κουμπιού, το οποίο αρχικά δεν είναι εμφανές. Για να γίνει εμφανές πρέπει πρώτα οι χρήστες να έχουν τοποθετήσει μία έξυπνη συσκευή στο κομμάτι του συμβάντος. Μόλις οι χρήστες προσθέσουν την πρώτη έξυπνη συσκευή τότε θα γίνει ορατό το κουμπί που αφορά το λογικό «OR» και «AND» στο κάτω δεξιά μέρος της οθόνης που αφορά το κομμάτι του συμβάντος (Εικόνα 44). Έτσι, οι χρήστες εάν επιθυμούν να προσθέσουν δεύτερη ή και περισσότερες έξυπνες συσκευές στο κομμάτι του συμβάντος πλέον θα πρέπει πρώτα να επιλέξουν ανάμεσα στο λογικό «AND» και «OR» (Εικόνα 45) και έπειτα, αυτόματα, θα ανοίξει το μενού των έξυπνων συσκευών για να επιλέξουν την έξυπνη συσκευή που επιθυμούν. Το αρχικό κουμπί «ADD EVENT» πλέον έχει γίνει μη ορατό για τους χρήστες.



Εικόνα 44. Εμφάνιση του κουμπιού που αφορά το λογικό OR και AND.



Εικόνα 45. Εμφάνιση λογικού OR και AND για επιλογή λογικής σχέσης και για προσθήκη έξυπνης συσκευής.

4.3.3.2 Δημιουργία Κανόνα στον Editor E - Wizard

Αρχικά, ο χρήστης καλείται να επιλέξει μεταξύ του συμβάντος (event) και της δράσης (action) σε κάθε προσθήκη έξυπνης συσκευής. Με τη χρήση του αντίστοιχου κουμπιού ανοίγει η λίστα με τις έξυπνες – διαθέσιμες συσκευές για την εισαγωγή της στον Editor.

Επιλογή «ADD ACTION»:

Η επιλεγμένη και ρυθμισμένη συμφώνα με τις επιλογές του χρήστη έξυπνη συσκευή προστίθεται στη λίστα με τα τις δράσεις. Στην προκειμένη περίπτωση έχει προστεθεί, μόνο, το έξυπνο κλιματιστικό στο κομμάτι της δράσης ρυθμισμένο έτσι ώστε να ανοίξει (on), στους 22 °C και σε τρόπο λειτουργίας υγρασίας (dry mode) (Εικόνα 46). Σε περίπτωση εισαγωγής δεύτερης συσκευής, αυτόματα, συσχετίζονται με το λογικό AND.



Εικόνα 46. Προσθήκη έξυπνου κλιματιστικού στη δράση.

Επιλογή «ADD EVENT»:

Για την πρώτη προσθήκη, ο χρήστης επιλέγει την συσκευή που επιθυμεί , έπειτα και ύστερα από την κατανόηση των λογικών συσχετίσεων, καλείτε να πατήσει σε ένα από δυο καινούργια κουμπιά καθώς η επιλογή «ADD EVENT» έχει αφαιρεθεί (Εικόνα 47).



Εικόνα 47. Εισαγωγή πρώτης έξυπνης συσκευής στο μέρος των συμβάντων.

Αναλόγως με την επιλογή , λαμβάνει χώρα και η αντίστοιχη συσχέτιση, με την προηγούμενη έξυπνη συσκευή. Αν για παράδειγμα ο χρήστης έχει βάλει ένα ρολόι με παραμετροποίηση “Αν η ώρα είναι 06:00” και αμέσως επόμενη συσκευή που θέλει να βάλει είναι ένα θερμόμετρο τότε θα πρέπει να επιλέξει τη λογική συσχέτιση με την χρήση του AND ή OR κουμπιού και έπειτα να επιλέξει από το μενού τη συσκευή αυτή (Εικόνα 48).

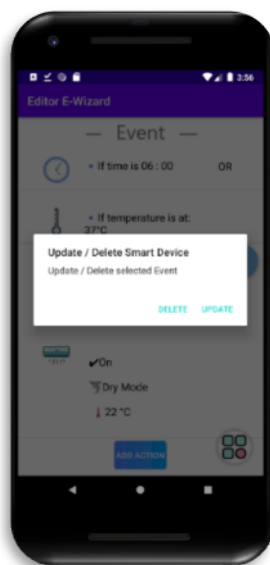


Εικόνα 48. Προσθήκη 2ης έξυπνης συσκευής στα συμβάντα. Οι δύο έξυπνες συσκευές συσχετίζονται μεταξύ τους με το λογικό OR.

Ο κανόνας της Εικόνα 48 μπορεί να θεωρηθεί χρήσιμος και να τεθεί προς αποθήκευση. Η αποθήκευση του κανόνα πραγματοποιείται με τον ίδιο τρόπο όπως και στον Editor Domino καθώς και η επανεκκίνηση.

4.3.3.3 Ανανέωση και Διαγραφή Έξυπνης Συσκευής

Όπως στις περισσότερες εφαρμογές έτσι και στην δική μας είναι αναγκαίο να περιέχεται η επιλογή του χρήστη να διαγράψει ή να αλλάξει τις παραμετροποιήσεις των έξυπνων συσκευών. Πατώντας παρατεταμένα σε κάποια συσκευή, που βρίσκεται στην περιοχή του συμβάντος (event), του εμφανίζεται το ακόλουθο μήνυμα με δύο (2) επιλογές, Διαγραφής η Ανανέωσης (Εικόνα 49). Αντίστοιχα και στη περίπτωση της δράσης (action) με μήνυμα: “Update / Delete Selected Action”.

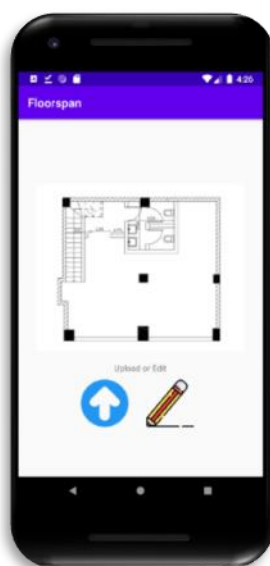


Εικόνα 49. Εμφάνιση pop-up μηνύματος για Διαγραφή ή Ανανέωση.

4.3.4 Κάτοψη (Floorspan)

Η συγκεκριμένη λειτουργία δίνει την δυνατότητα στον χρήστη να επιλέξει κάποια υπάρχουσα κάτοψη του χώρου στον οποίο βρίσκονται οι έξυπνες συσκευές και αφού την μεταφορτώσει στην εφαρμογή ως εικόνα να την επεξεργαστεί (Εικόνα 50).

Ο τρόπος επεξεργασίας της κάτοψης γίνεται με μια λειτουργία «Drag and Drop» των εικονιδίων που βρίσκονται στο πάνω μέρος της οθόνης. Ο χρήστης μπορεί να βάλει ως Domino τις έξυπνες συσκευές πάνω στην εικόνα και να την αποθηκεύσει. Με αυτό τον τρόπο μπορεί μέσω του floorspan να βλέπει τις έξυπνες συσκευές τοποθετημένες στην κάτοψή του και να έχει μια εικόνα για το που βρίσκονται μέσα στο χώρο.



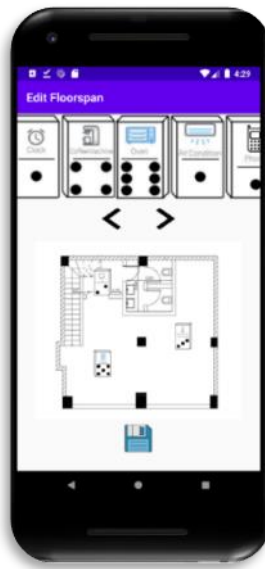
Εικόνα 50. Περιβάλλον Κάτοψης (Floorspan).

4.3.4.1 Μεταφόρτωση και Επεξεργασία

Με τη χρήση του μπλε κουμπιού μεταφόρτωση (upload) μπορεί ο χρήστης αφού περιηγηθεί στις φωτογραφίες του κινητού να επιλέξει την κάτοψη και να την ανεβάσει. Έπειτα, με το εικονίδιο του μολυβιού μπορεί να την επεξεργαστεί (edit).

4.3.4.2 Τοποθέτηση Έξυπνων Συσκευών

Κάνοντας drag and drop πάνω στην φωτογραφία τα Domino τα τοποθετεί στις κατάλληλες θέσεις (Εικόνα 51).



Εικόνα 51. Προσθήκη εικονιδίων έξυπνων συσκευών στην μεταφορτωμένη εικόνα κάτοψης.

Παρατηρούμε πως στο κάτω μέρος έχουν τοποθετηθεί τρεις έξυπνες συσκευές :

- Έξυπνο Ψυγείο
- Έξυπνο Θερμόμετρο
- Έξυπνη τηλεόραση

Επιπλέον υπάρχει έλεγχος σε περίπτωση που ο χρήστης τοποθετήσει στην ίδια θέση δύο (2) αντικείμενα, εμφανίζεται μήνυμα πως μπορεί να τοποθετήσει μόνο μία. Σε περίπτωση αντικατάστασης θα πρέπει να κάνει ξανά edit την φωτογραφία.

Τα δύο (2) βελάκια (arrows) βοηθούν τον χρήστη να κάνει scroll δεξιά και αριστερά την λίστα με τις έξυπνες συσκευές.

4.3.4.3 Αποθήκευση Επεξεργασμένης Εικόνας

Τέλος, οι χρήστες έχουν την δυνατότητα να αποθηκεύσουν την νέα εικόνα κάτοψης που έχουν δημιουργήσει έπειτα από την επεξεργασία της πατώντας την εικόνα της δισκέτας η οποία βρίσκεται κάτω

ακριβώς από την επεξεργασμένη εικόνα, η εικόνα θα αποθηκευτεί στον αποθηκευτικό χώρο του κινητού του χρήστη.

4.3.5 Αποθηκευμένοι Κανόνες

Αφού οι χρήστες έχουν ολοκληρώσει την διαδικασία ανάπτυξης κανόνα σ' έναν απ' του δύο Editors (Domino ή E-Wizard) τότε το επόμενο βήμα είναι να τον αποθηκεύσουν. Η αποθήκευση των κανόνων γίνεται με τον ίδιο τρόπο που αναφέρθηκε σε προηγούμενη ενότητα για κάθε Editor (βλέπε Ενότητα «4.3.2.5 Αποθήκευση και Επανεκκίνηση»). Αφού πληκτρολογήσουν το όνομα του κανόνα και πατήσουν το αντίστοιχο κουμπί για να ολοκληρωθεί η διαδικασία τότε μεταβαίνουν σε μια νέα δραστηριότητα που αφορά τους αποθηκευμένους κανόνες (Εικόνα 52).

Η εμφάνιση των αποθηκευμένων κανόνων γίνεται μέσω μίας λίστας όπου σε κάθε γραμμή της λίστα υπάρχει :

1. Μία εικόνα που παραπέμπει σε μορφή που θα μπορούσε να απεικονίζεται μία συνταγή (recipe) ώστε να είναι φιλική προς το χρήστη.
2. Όνομα του κανόνα.
3. Εικονίδιο του Editor στον οποίο φτιάχτηκε ο κανόνας.



Εικόνα 52. Λίστα αποθηκευμένων κανόνων.

4.3.6 Εκπαίδευση στους Editors

Όσον αφορά την επεξήγηση των δυνατοτήτων των δύο Editor δημιουργήσαμε μια επιλογή την οποία ονομάσαμε « Tutorial », δηλαδή Οδηγό Χρήσης. Οι χρήστες μπορούν να επιλέξουν την επιλογή αυτή και

έπειτα με βάσει τον Editor της επιλογής τους να διαβάσουν το Tutorial όσες φορές χρειάζονται. Η επεξήγηση έχει γίνει με αναλυτικά παραδείγματα και καθώς θεωρήσαμε αναγκαία την χρήση εικόνων.

Πιο αναλυτικά υπάρχει η επιλογή μεταξύ δυο κουμπιών του «Domino Tutorial» και του «E-Wizard Tutorial» .

4.3.6.1 Editor Domino

Έχουν συμπεριληφθεί συνολικά πέντε επεξηγηματικές διαφάνειες (slides) ξεκινώντας με την εισαγωγή και την περιγραφή του Editor Domino.

- (1) Editor Domino: “Οι έξυπνες συσκευές αναπαρίστανται από Domino. Για να δημιουργηθεί κάποιος κανόνας πρέπει να τοποθετηθούν τα Domino πάνω στο δοθέν πλαίσιο με έναν συγκεκριμένο τρόπο (Κάθετα ή Οριζόντια). Μπορείς να αποθηκεύσεις τον κανόνα ως Recipe και έπειτα να περιμένεις μέχρι να ενεργοποιηθεί το αποτέλεσμα που έχεις ορίσει”. Πρόκειται για μια μικρή εισαγωγή και περιγραφή της γενικής χρήσης του Editor Domino.
- (2) Επιλογή Αποτελέσματος (Action): “Το πρώτο βήμα είναι να ορισθεί το αποτέλεσμα επιλέγοντας την έξυπνη συσκευή που το εγείρει, Για παράδειγμα, θα ήθελα η έξυπνη καφετιέρα να μου φτιάξει καφέ.”. Σε αυτό το στάδιο μέσω του παραδείγματος δίνουμε στον χρήστη να καταλάβει την λειτουργία του Action.
- (3) Επιλογή Συμβάντων (Events): “Το δεύτερο βήμα είναι να ορισθεί το συμβάν επιλέγοντας την έξυπνη συσκευή που το εγείρει και να τοποθετηθεί στο πλαίσιο. Ένα παράδειγμα Συμβάντος είναι: Αν η ώρα του έξυπνου ρολογιού είναι 7:30 π.μ. Σε περίπτωση που χρειαστεί να τοποθετηθεί παραπάνω από ένα πρέπει να επιλεγεί μια Κάθετη ή Οριζόντια τοποθέτηση.” Γίνεται επεξήγηση του Event καθώς και μια αναφορά στον τρόπο τοποθέτησης
- (4) Κάθετη τοποθέτηση: “Τα Domino που έχουν τοποθετηθεί κάθετα θα πρέπει πάντοτε να ικανοποιούν τις συνθήκες των Συμβάντων για να εγείρει το αποτέλεσμα. Για παράδειγμα για να εγείρει το αποτέλεσμα πρέπει η ώρα να είναι 7:30 π.μ. και ταυτόχρονα η Τηλεόραση να παίζει τα νέα. Σε περίπτωση που κάποιο από τα δυο Events δεν ικανοποιείται δεν θα εγείρει το Αποτέλεσμα.”. Σε αυτό το παράδειγμα εξηγούμε τον όρο λογικό AND χωρίς να γίνεται αντιληπτό από τον χρήστη. Επίσης περιέχεται και μία εικόνα που δείχνει τις 2 συνθήκες που περιγράφονται.
- (5) Οριζόντια τοποθέτηση: “Τα Dominos που τοποθετούνται οριζόντια μπορούν να εγείρουν το Αποτέλεσμα δεδομένου ότι υπάρχει τουλάχιστον ένα Event που ικανοποιείται. Για παράδειγμα , Αν η ώρα είναι 7:30 π.μ. ή Τηλεόραση παίζει τα νέα τότε θα εγείρει το Αποτέλεσμα. Επίσης, αν κανένα από τα δυο Events δεν ικανοποιούνται τότε το Αποτέλεσμα δεν θα εγείρει.”. Πρόκειται για την επεξήγηση του λογικού OR με παράδειγμα ικανοποίησης των Συνθηκών.

4.3.6.2 Editor E - Wizard

Έχουν συμπεριληφθεί συνολικά έξι επεξηγηματικές διαφάνειες ξεκινώντας με την εισαγωγή και την περιγραφή του Editor E – Wizard.

- (1) E-Wizard Editor: “Σε αυτόν τον Editor οι έξυπνες συσκευές αναπαρίστανται από εικονίδια, μπορείς να δημιουργήσεις έναν κανόνα προσθέτοντας Συμβάντα και Αποτελέσματα. Έπειτα μπορείς να το αποθηκεύσεις σαν Recipe και να περιμένεις να εγείρει το Αποτέλεσμα”. Υπάρχει η εισαγωγή και η περιγραφή του Editor, θεωρούμε πλέον ότι οι έννοιες Συμβάν και Αποτέλεσμα

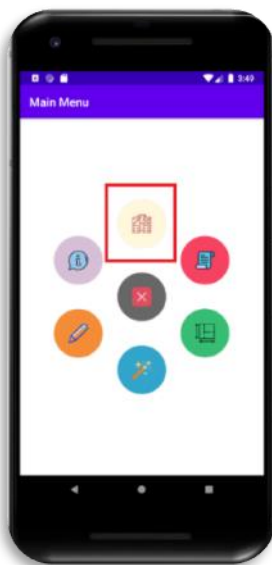
είναι κατανοητές καθώς έχουν πλήρως αναλυθεί στον Editor Domino. Σε περίπτωση που δεν είναι κατανοητές πλήρως, υπάρχει η δυνατότητα να ανατρέξουν στο Tutorial του Domino για επανάληψη.

- (2) Προσθήκη Συμβάντων (1/2): “ Η πρώτη επιλογή Συμβάντος γίνεται με τη χρήση του κουμπιού « Προσθήκη Συμβάντος » και έπειτα τοποθετείται στην λίστα με τα Συμβάντα “
- (3) Προσθήκη Αποτελεσμάτων: “ Μπορείς να τοποθετήσεις πολλαπλά Αποτελέσματα πατώντας στο κουμπί « Προσθήκη Αποτελέσματος » και έπειτα προστίθεται στη λίστα με τα Αποτελέσματα.”. Αξίζει να αναφερθεί σε αυτό το σημείο πως υπάρχει ένα σύνολο Αποτελεσμάτων σε σχέση με τον Domino Editor καθώς εκεί υπάρχει ένα Αποτέλεσμα (το τελευταίο Domino προς ρίψη) για αυτό τον λόγο χρησιμοποιήθηκε εδώ ο πληθυντικός.
- (4) Προσθήκη Συμβάντων (2/2): “Από την δεύτερη συσκευή προς τοποθέτηση και μετά πρέπει να γίνει μία επιλογή μεταξύ δυο κουμπιών, του «AND» κουμπιού και του «OR» κουμπιού για να συνεχιστεί η διαδικασία τοποθέτησης Συμβάντων στην λίστα.” Υπάρχει η αναφορά του λογικού ΚΑΙ του λογικού Ή , η επεξήγηση θα γίνει μέσω παραδειγμάτων στην επόμενη διαφάνεια.
- (5) Το Πρώτο Κουμπί: “Αν δεις την επιλογή «AND» και «OR» σημαίνει πως η προηγούμενη συνθήκη θα συνδεθεί με την νέα συνθήκη της καινούργιας έξυπνης συσκευής που θα προσθέσεις. Σε περίπτωση που επιλέξεις το πρώτο κουμπί «AND» θα πρέπει οι συνδεδεμένες Συνθήκες να ικανοποιούνται ταυτόχρονα”.
- (6) Το Δεύτερο Κουμπί: “Από την άλλη αν επιλέξεις το δεύτερο κουμπί «OR» τουλάχιστον ένα από τα συνδεδεμένα Συμβάντα χρειάζεται να ικανοποιείται έτσι ώστε να εγείρει το Αποτέλεσμα”.

Στις 2 περιπτώσεις των Οδηγών Χρήσης είδαμε πως οι επεξηγήσεις έγιναν με μέσω την χρήση των παραδειγμάτων και απλών σεναρίων. Το σενάριο που χρησιμοποιήσαμε ήταν “Αν η ώρα είναι 7:30π.μ OR / AND το τηλεοπτικό πρόγραμμα είναι τα νέα TOTE να φτιάξει η καφετιέρα καφέ.”. Το θεωρήσαμε αρκετά απλό καθώς και οικείο ως πρώτη επαφή με τον Editor Domino, δίνοντας το έναυσμα για μια εμπλουτίσει του σεναρίου αυτού από τους χρήστες.

4.3.7 Πληροφορίες Έξυπνων Συσκευών

Το τελευταίο κομμάτι της ενότητας Εφαρμογή Android αφορά την επιλογή που έχουν οι χρήστες μέσω της εφαρμογής να πληροφορηθούν σχετικά με τις δυνατότητες των έξυπνων συσκευών τόσο στο μέρος του συμβάντος (event) όσο και στο μέρος της δράσης (action). Έχουν την δυνατότητα να ενημερωθούν για τις ακριβείς ρυθμίσεις που μπορεί να εφαρμόσει κάθε έξυπνη συσκευή. Για την εφαρμογή της συγκεκριμένης επιλογής οι χρήστες, αρχικά, θα πρέπει να επιλέξουν το κατάλληλο εικονίδιο από το μενού της εφαρμογής (Εικόνα 53).



Εικόνα 53. Πληροφορίες Έξυπνων Συσκευών στο μενού της εφαρμογής.

Έπειτα, θα παρουσιαστούν όλες οι έξυπνες συσκευές απεικονισμένες ως dominos καθώς και οι πληροφορίες αυτών. Οι έξυπνες συσκευές και οι αναγραφόμενες ρυθμίσεις αυτών απεικονίζονται εναλλάξ. Δηλαδή, αρχικά το εικονίδιο του ρολογιού ως domino εμφανίζεται αριστερά και στα δεξιά αυτού οι ρυθμίσεις. Ακριβώς από κάτω του βρίσκεται η επόμενη έξυπνη συσκευή που είναι η τηλεόραση. Παρατηρείται ότι οι ρυθμίσεις της τηλεόρασης αναγράφονται στα αριστερά (κάτω από το εικονίδιο του ρολογιού) και το εικονίδιο της τηλεόρασης εμφανίζεται στα δεξιά (κάτω από το τις ρυθμίσεις του ρολογιού). Ο λόγος είναι ότι κατ' αυτό τον τρόπο γίνεται πιο εύκολα αντιληπτό από τους χρήστες οι διαχωρισμοί μεταξύ των ρυθμίσεων και του εικονιδίου ίδιας έξυπνης συσκευής αλλά και οι διαχωρισμοί μεταξύ και των υπολοίπων έξυπνων συσκευών και των ρυθμίσεων αυτών. Παρακάτω, τα στιγμιότυπα οθόνης για την επεξήγηση των ρυθμίσεων της κάθε έξυπνης συσκευής.

4.3.7.1 Έξυπνο Ρολόι

Συμβάν: Η μία και μοναδική ρύθμιση που μπορεί να εφαρμοστεί στο έξυπνο ρολόι είναι ο χρόνος. Οι χρήστες μέσω της συγκεκριμένης έξυπνης συσκευής έχουν την δυνατότητα να ορίσουν την ώρα και το/τα λεπτό/ά που θα επιθυμούσαν μία δράση να πραγματοποιηθεί.

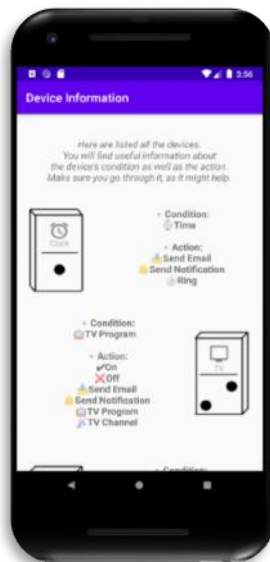
Δράση: Συνολικά περιλαμβάνονται τρεις (3) δυνατές δράσεις. Η πρώτη αφορά την αποστολή ηλεκτρονικού μηνύματος, η δεύτερη την αποστολή ειδοποίησης και η τελευταία του κουδουνίσματος του έξυπνου ρολογιού (Εικόνα 54).

4.3.7.2 Έξυπνη Τηλεόραση

Συμβάν: Η μία και μοναδική ρύθμιση που μπορεί να εφαρμοστεί στην έξυπνη τηλεόραση είναι η επιλογή του τηλεοπτικού προγράμματος. Επιλέγοντας το τηλεοπτικό πρόγραμμα στο κομμάτι του συμβάντος

σημαίνει ότι για να ισχύσει η έξυπνη τηλεόραση θα πρέπει να απεικονίζει το συγκεκριμένο τηλεοπτικό πρόγραμμα που έχει επιλεγεί από το χρήστη.

Δράση: Δίνονται στο σύνολο έξι (6) δυνατές δράσεις. Το άνοιγμα και κλείσιμο της έξυπνης τηλεόρασης, η αποστολή ηλεκτρονικού μηνύματος, η αποστολή ειδοποίησης η επιλογή τηλεοπτικού προγράμματος και καναλιού εφ' όσον έχει επιλεγεί πιο πριν το άνοιγμα της έξυπνης τηλεόρασης (Εικόνα 54).



Εικόνα 54. Πληροφορίες Έξυπνου Ρολογιού και Τηλεόρασης.

4.3.7.3 Έξυπνο Θερμόμετρο

Συμβάν: Το έξυπνο θερμόμετρο περιλαμβάνει μία μοναδική ρύθμιση στο κομμάτι του συμβάντος και αυτή είναι ο ορισμός της θερμοκρασίας. Βάζοντας την θερμοκρασία που επιθυμούν οι χρήστες θα πρέπει, για να πραγματοποιηθεί η δράση, η θερμοκρασία που έδωσαν οι χρήστες να συμπίπτει με την θερμοκρασία του έξυπνου θερμομέτρου.

Δράση: Δύο δυνατές επιλογές δράσεις. Η πρώτη επιλογή είναι αυτή της αποστολής ηλεκτρονικού μηνύματος και η δεύτερη της αποστολής ειδοποίησης (Εικόνα 55).

4.3.7.4 Έξυπνη Καφετιέρα

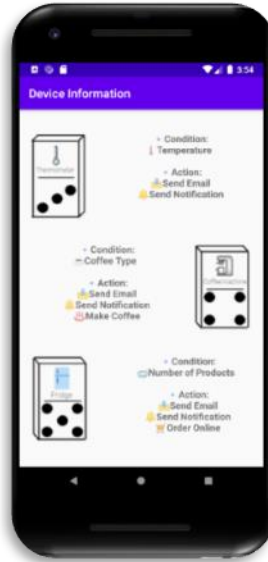
Συμβάν: Αναλόγως την θερμοκρασία του καφέ που πίνουν οι χρήστες έτσι ορίζεται και στο κομμάτι του συμβάντος η έξυπνη καφετιέρα. Δηλαδή, οι χρήστες έχουν την επιλογή να επιλέξουν ανάμεσα σε «κρύο» και «ζεστό» καφέ που πίνουν εκείνη την στιγμή ή σε ανεξάρτητο χρονικό όριο.

Δράση: Τρεις (3) δυνατές επιλογές δράσεις. Η πρώτη επιλογή είναι αυτή της αποστολής ηλεκτρονικού μηνύματος, η δεύτερη της αποστολής ειδοποίησης και η τρίτη της παραγωγής καφέ (Εικόνα 55).

4.3.7.5 Έξυπνο Ψυγείο

Συμβάν: Έχοντας μια λίστα με προτεινόμενα προϊόντα που μπορεί να περιλαμβάνει ένα ψυγείο οι χρήστες μπορούν βάσει του αριθμού αυτών να ορίσουν ένα συμβάν για το έξυπνο ψυγείο.

Δράση: Τρεις δυνατές επιλογές δράσεις. Η πρώτη επιλογή είναι αυτή της αποστολής ηλεκτρονικού μηνύματος, η δεύτερη της αποστολής ειδοποίησης και η τρίτη της ηλεκτρονικής παραγγελίας (Εικόνα 55).



Εικόνα 55. Πληροφορίες Έξυπνου Θερμομέτρου, Καφετιέρας και Ψυγείου.

4.3.7.6 Έξυπνος Φούρνος

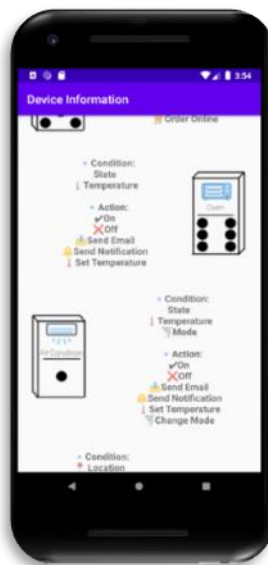
Συμβάν: Η πρώτη επιλογή ρύθμισης του έξυπνου φούρνου είναι αυτή της κατάστασής του (state) δηλαδή, είτε ανοιχτός (on) είτε κλειστός (off) και η δεύτερη επιλογή είναι η θερμοκρασία.

Δράση: Στο σύνολο προσφέρονται πέντε (5) δυνατές επιλογές δράσεις. Το άνοιγμα και το κλείσιμο του έξυπνου φούρνου, η αποστολή ηλεκτρονικού μηνύματος και ειδοποίησης και η προσθήκη θερμοκρασίας, εφ' όσον έχει επιλεγεί πιο πριν το άνοιγμα του έξυπνου φούρνου (Εικόνα 56).

4.3.7.7 Έξυπνο Κλιματιστικό

Συμβάν: Οι επιλογές των χρηστών στο κομμάτι του συμβάντος για το κλιματιστικό είναι τρεις (3). Η πρώτη αφορά την κατάσταση του κλιματιστικού δηλαδή, είτε είναι ανοιχτό (on) είτε κλειστό (off), η δεύτερη αφορά την θερμοκρασία και η τρίτη τον τρόπο λειτουργίας του κλιματιστικού.

Δράση: Έξι (6) δυνατές επιλογές για τον ορισμό δράσης. Το άνοιγμα και το κλείσιμο του έξυπνου κλιματιστικού, η αποστολή ηλεκτρονικού μηνύματος και ειδοποίησης και η προσθήκη θερμοκρασίας και τρόπου λειτουργίας εφ' όσον έχει επιλεγεί πιο πριν το άνοιγμα του έξυπνου κλιματιστικού (Εικόνα 56).



Εικόνα 56. Πληροφορίες Έξυπνου Φούρνου και Κλιματιστικού.

4.3.7.8 Έξυπνο Κινητό Τηλέφωνο

Συμβάν: Συνολικά έξι (6) ξεχωριστές κατηγορίες ρυθμίσεων υπάρχουν στο κομμάτι του συμβάντος για το έξυπνο τηλέφωνο. Οι κατηγορίες είναι οι εξής :

1. *Τοποθεσία* : Αφορά την τοποθεσία των χρηστών.
2. *Υγεία* : Δυνατότητα προσθήκης άθλησης στον κανόνα.
3. *Μουσική* : Έλεγχος εάν οι χρήστες ακούνε μουσική.
4. *Μπαταρία* : Οι χρήστες μπορούν να ορίσουν με ακρίβεια το ποσοστό μπαταρίας.
5. *Ακουστικά* : Έλεγχος εάν είναι συνδεδεμένα ακουστικά.
6. *Φορητές συσκευές* : Εξέταση εάν το έξυπνο κινητό είναι συνδεδεμένο με την έξυπνη φορητή συσκευή.

Δράση: Όπως και στο κομμάτι του συμβάντος έτσι κι εδώ υπάρχουν έξι (6) ξεχωριστές κατηγορίες ρυθμίσεων(Εικόνα 57). Οι κατηγορίες είναι εξής :

1. *Τοποθεσία* : Παρακολούθηση τοποθεσίας και αποστολή ειδοποίησης που αφορά τοποθεσία.
2. *Υγεία* : Αποστολή ειδοποιήσεων είτε για τρέξιμο είτε για περπάτημα.
3. *Μουσική* : Έναρξη ή κλείσιμο της μουσικής λίστας αναπαραγωγής.
4. *Μπαταρία* : Αποστολή ειδοποίησης για φόρτιση έξυπνου κινητού και ορισμός του τρόπου φόρτισης σε γρήγορη.
5. *Ακουστικά* : Αυξομείωση του ήχου.
6. *Φορητές συσκευές* : Άνοιγμα ή κλείσιμο της έξυπνης φορητής συσκευής και αποστολή ειδοποίησης σε αυτή.

4.3.7.9 Έξυπνες Φορητές Συσκευές

Συμβάν: Η πρώτη ρύθμιση στο κομμάτι του συμβάντος για τις έξυπνες φορητές συσκευές είναι ο ορισμός για το εάν την φοράω. Έπειτα η κατάσταση που βρίσκομαι, δηλαδή εάν περπατάω ή τρέχω καθώς και

διάφορες πληροφορίες σχετικές με την υγεία όπως ο παλμός της καρδιάς, πίεση αίματος και επίπεδα σακχάρου στο αίμα.

Δράση: Συνολικά περιλαμβάνονται τρεις (3) δυνατές δράσεις. Η πρώτη αφορά την αποστολή ηλεκτρονικού μηνύματος, η δεύτερη την αποστολή ειδοποίησης και η τελευταία του κουδουνίσματος της έξυπνης φορητής συσκευής(Εικόνα 57).



Εικόνα 57. Πληροφορίες Έξυπνου Κινητού Τηλεφώνου και Φορητών Συσκευών.

4.4 Σχεδιασμός και Λειτουργίες Διεπαφής

4.4.1 Εισαγωγή

Ο σχεδιασμός μιας φιλικής διεπαφής συχνά προκύπτει να είναι μια αρκετά προκλητική διαδικασία για τους προγραμματιστές καθώς το σύνηθες είναι οι προγραμματιστές να λαμβάνουν σχεδιασμένες τις γραφικές διαπαφές. Στην εφαρμογή μας προσπαθήσαμε όσο το περισσότερο να προσεγγίσουμε ένα απλό, λυτό αλλά και όμορφο UI με τη χρήση του JustinMind Prototype και του Adobe XD.

Με τη χρήση του JustInMind Prototype σχεδιάσαμε όλες τις έξυπνες συσκευές υπό την μορφή Domino και έπειτα της κάναμε εξαγωγή με την μορφή εικόνας (jpg ή png) για τη χρήση στην εφαρμογή Android.

Έγινε χρήση του εργαλείου Adobe XD για τον σχεδιασμό των φράσεων που περιέχονται στους δύο Editors οι οποίες οριοθετούν τα πλαίσια τοποθέτησης των έξυπνων συσκευών

4.4.2 Εμπειρία Χρήσης Εφαρμογής

Για να πετύχουμε υψηλής εμπειρία χρήσης [11], χρησιμοποιήσαμε τα εξής:

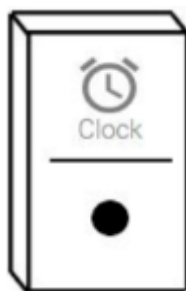
- Έξυπνες συσκευές υπό την μορφή Domino (Editor Domino).
- Έξυπνες συσκευές υπό την μορφή Icons (Editor E – Wizard).
- Μενού (επιλογή συσκευής) που εμφανίζεται με κύλιση από αριστερά.

- Κατάλληλα χρώματα για ξεκούραση των χρηστών.
- Χρήση Pop-Up παραθύρων για την εύκολη πρόσβαση στις ρυθμίσεις των συσκευών.
- Αναγραφή των κανόνων υπό την μορφή κειμένου.
- Γραφικός διαχωρισμός μεταξύ Συμβάντος (Event) – Δράσης (Action) στις πληροφορίες του κανόνα.
- Animations για τον τρισδιάστατο οραματισμό του κανόνα.
- Drag n Drop για την εύκολη τοποθέτηση των έξυπνων συσκευών στην κάτοψη (floorspan).
- Χρήση Emoji.

4.4.3 Χρήση Dominos ως metaphor για την απεικόνιση των έξυπνων συσκευών στον Editor

Domino

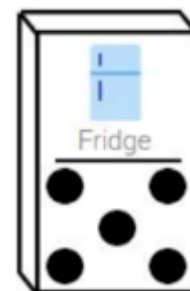
Η επιλογή της εμφάνισης των έξυπνων συσκευών υπό την μορφή domino πραγματοποιήθηκε για να φέρει την αίσθηση ότι πρόκειται για παιχνίδι. Ως συνέπεια, οι χρήστες θα νιώσουν πιο οικεία κατά την πρώτη τους επαφή με τον Editor γιατί πρόκειται για ένα διάσημο παιχνίδι που όλοι γνωρίζουν.



Εικόνα 58. Domino
Έξυπνο Ρολόι.



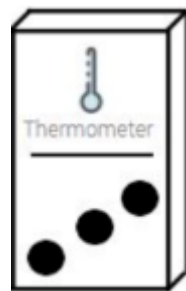
Εικόνα 59. Domino
Έξυπνης Καφετιέρας.



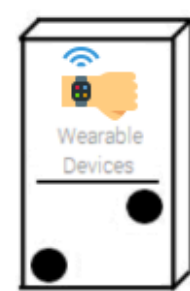
Εικόνα 60. Domino
Έξυπνο Ψυγείου.



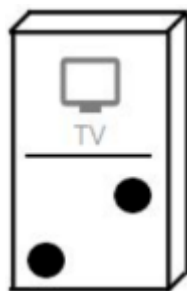
Εικόνα 61. Domino
Έξυπνο Κλιματιστικό.



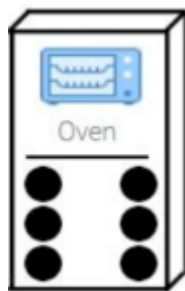
Εικόνα 62. Domino
Έξυπνο Θερμόμετρο.



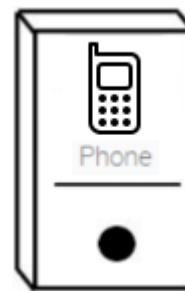
Εικόνα 63. Domino Έξυπνων
Φορητών Συσκευών.



Εικόνα 64. Domino Έξυπνης Τηλεόρασης.



Εικόνα 65. Domino Έξυπνου Φούρνου.



Εικόνα 66. Domino Έξυπνου Κινητού Τηλεφώνου.

4.4.4 Χρήση Icons ως metaphor για την απεικόνιση των έξυπνων συσκευών στον Editor E – Wizard

Η επιλογή των εικόνων στον Editor E-Wizard είναι εξίσου σημαντική καθώς θα πρέπει η εικόνα να αντιπροσωπεύει και την έξυπνη συσκευή. Επιλέξαμε τη χρήση icons (μικρών εικονιδίων) για την αναπαράσταση των έξυπνων συσκευών.



Εικόνα 67. Icon Έξυπνου Ρολογιού.



Εικόνα 68. Icon Έξυπνης Τηλεόρασης.



Εικόνα 69. Icon Έξυπνου Θερμομέτρου.



Εικόνα 70. Icon Έξυπνης Καφετιέρας.



Εικόνα 71. Icon Έξυπνου Ψυγείου.



Εικόνα 72. Icon Έξυπνου Φούρνου.



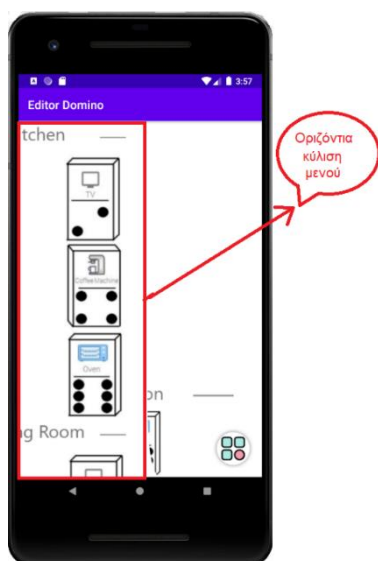
Εικόνα 73. Icon Έξυπνου Κινητού Τηλεφώνου



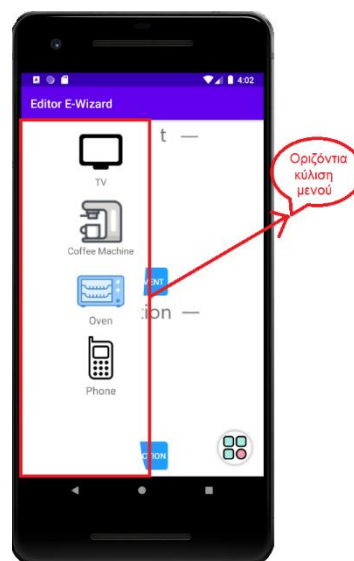
Εικόνα 74. Icon Έξυπνων Φορητών Συσκευών

4.4.5 Μενού έξυπνων συσκευών στους δύο Editors

Το μενού εμφανίζεται αυτόματα στην εφαρμογή (και στους δύο Editors) με κύλιση από αριστερά προς τα δεξιά. Στον Editor Domino το μενού εμφανίζεται έπειτα από το πάτημα του βοηθητικού “shadow” domino είτε στο κομμάτι του συμβάντος είτε στο κομμάτι της δράσης. Στον Editor E-Wizard το μενού εμφανίζεται έπειτα από το πάτημα του αντίστοιχου κουμπιού είτε στο κομμάτι του συμβάντος με το αντίστοιχο κουμπί «ADD EVENT» αλλά και με τα κουμπιά «AND» και «OR» είτε στο κομμάτι της δράσης με το αντίστοιχο κουμπί «ADD ACTION». Με αυτό τον τρόπο γίνονται εύκολα προσβάσιμες οι έξυπνες συσκευές κατά τη δημιουργία κανόνων και για τους δύο Editors στους χρήστες (Εικόνα 84, Εικόνα 85).



Εικόνα 75. Οριζόντια κύλιση μενού στον Editor Domino.

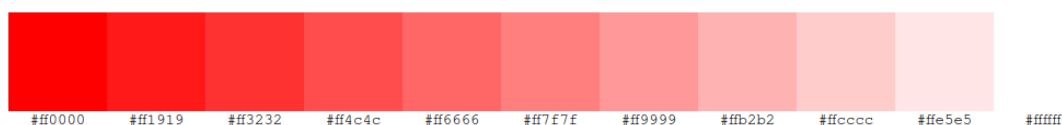


Εικόνα 76. Οριζόντια κύλιση μενού στον Editor E-Wizard.

4.4.6 Χρώματα

Η σωστή επιλογή χρωματικών τόνων είναι εξίσου σημαντική για την εμπειρία των χρηστών αλλά και για την ξεκούραση τους. Έχουμε συμπεριλάβει αποχρώσεις κοντά στο γκρι για τα περισσότερα χρώματα καθώς στοχεύουμε με αυτές τις αποχρώσεις στην ξεκούραση των χρηστών. Δηλαδή για αποχρώσεις (πχ. Κόκκινου (Εικόνα 86) στις επιλογές [#ff9999](#) - [#ffcccc](#).

Tints of #ff0000



Εικόνα 77. Αποχρώσεις κόκκινου χρώματος.

Επίσης, έγιναν και επιλογές χρωμάτων στον χώρο των “semi – transparent” καθώς και στις χρωματικές βαθμίδες (Gradient Colors) για να βελτιώσουμε την εμπειρία και να δώσουμε έμφαση έτσι ώστε να ξεχωρίζουν (Εικόνα 87).



Εικόνα 78. Χρήση Gradient Color στην Εκπαίδευση των Editors.

4.4.7 Pop – Up Παράθυρα

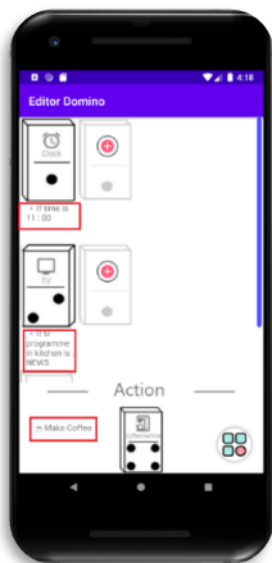
Θεωρήσαμε καθοριστική την χρήση “Pop-Up” παραθύρων για την παραμετροποίηση των έξυπνων συσκευών καθώς θεωρούμε ότι είναι ο πιο οικείος τρόπος για την προβολή πληροφορίας στο υπάρχον περιβάλλον χωρίς ανακατεύθυνση (Εικόνα 88).



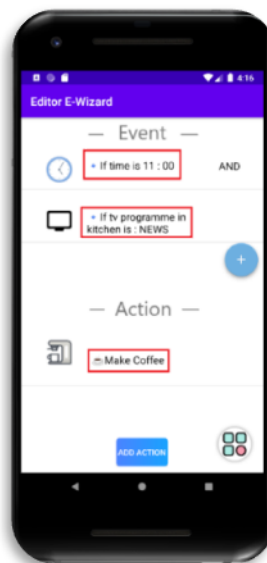
Εικόνα 79. Παράδειγμα Pop – Up Παραθύρου.

4.4.8 Αναγραφή Ενέργειών Έξυπνων Συσκευών

Οι ενέργειες των έξυπνων συσκευών αναγράφονται ως κείμενο στους Editors τόσο στο κομμάτι του συμβάντος όσο και στο κομμάτι της δράσης κατά τη διάρκεια δημιουργίας του κανόνα έτσι ώστε ο χρήστης να είναι σε θέση να δει την παραμετροποίηση του χωρίς να χρειαστεί κάποια περαιτέρω ενέργεια αλλά και να γνωρίζει την ρύθμιση που έχει ορίσει για κάθε συγκεκριμένη έξυπνη συσκευή κατά τη διάρκεια δημιουργίας του κανόνα (Εικόνα 89, Εικόνα 90).



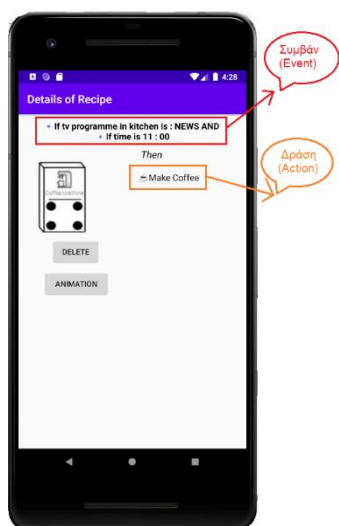
Εικόνα 80. Αναγραφή ενέργειας έξυπνης συσκευής στον Editor Domino.



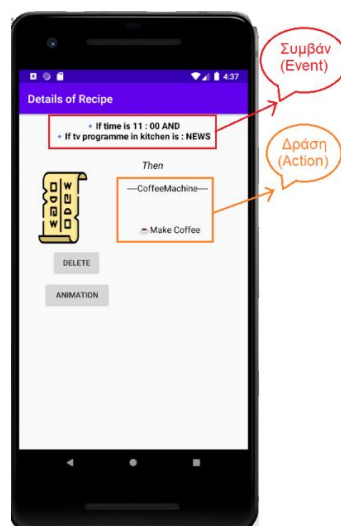
Εικόνα 81. Αναγραφή ενέργειας έξυπνης συσκευής στον Editor E-Wizard.

4.4.9 Αναγραφή Αποθηκευμένου Κανόνα

Επιπλέον, στο κομμάτι της προβολής του αποθηκευμένου κανόνα έχουμε διαχωρίσει το Συμβάν και την Δράση. Το Συμβάν εμφανίζεται με **bold** γραμματοσειρά ενώ η Δράση εμφανίζεται *Italic* με τα δεδομένα στοιχισμένα στο κέντρο και το ένα κάτω από το άλλο. Το κομμάτι του συμβάντος απεικονίζεται με τον ίδιο τρόπο και στους δύο Editors. Αντίθετα, το κομμάτι της δράσης διαφέρει. Στον Editor Domino προστίθεται αριστερά από το κείμενο που αναγράφει την δράση και η έξυπνη συσκευή που πραγματοποιεί την ενέργεια της δράσης υπό την μορφή domino. Στον Editor E-Wizard προστίθεται στο ίδιο σημείο με τον Editor Domino μία εικόνα που απεικονίζει έναν πάπυρο και παραπέμπει σε «συνταγή», δηλαδή recipe (Εικόνα 91, Εικόνα 92) καθώς επίσης επειδή στον editor E-Wizard στο κομμάτι της δράσης μπορούν να προστεθούν παραπάνω από μία έξυπνες συσκευές γι' αυτό το λόγο πρώτα αναγράφεται σε απλό κείμενο η έξυπνη συσκευή και έπειτα η αντίστοιχη ενέργεια που πραγματοποιεί.



Εικόνα 82. Αναγραφή αποθηκευμένου κανόνα που δημιουργήθηκε με τον Editor Domino.



Εικόνα 83. Αναγραφή αποθηκευμένου κανόνα που δημιουργήθηκε στον Editor E – Wizard.

4.4.10 Animation

Το Animation αποσκοπεί στον οραματισμό της πραγματοποίησης του κανόνα για τον Editor Domino. Έτσι, ο χρήστης μπορεί να δει στον χώρο τον κανόνα που έχει αποθηκεύσει και να είναι το έναυσμα για τον οραματισμό περίπλοκων κανόνων στον τρισδιάστατο χώρο (Εικόνα 93).

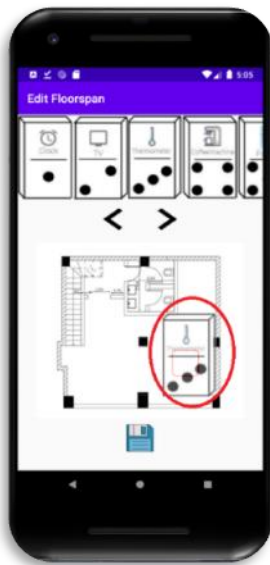


Εικόνα 84. Animation αποθηκευμένου κανόνα.

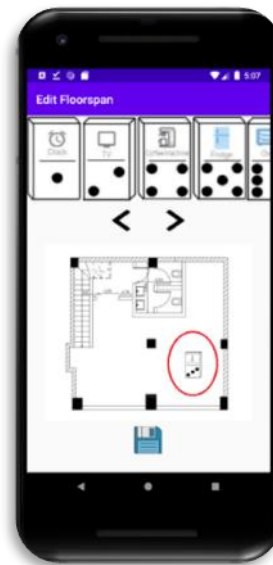
Στο Animation αναγράφεται το Domino της Δράσης καθώς και όλα τα Συμβάντα που συμβάλουν. Όλες οι παραμετροποιήσεις των έξυπνων συσκευών εμφανίζονται στο κάτω μέρος του εικονιδίου του 3D Domino και στο τέλος εμφανίζεται το μήνυμα επιτυχίας «Success».

4.4.11 Κάτοψη (Floorspan)

Το Drag n Drop είναι ένας πολύ εύκολος τρόπος και πολλές φορές οικείος στους περισσότερους χρήστες για την αλλαγή κατάστασης / μεταφορά κάποιου αντικειμένου σε έναν χώρο. Ένα απλό παράδειγμα είναι το Drag n Drop που κάνουμε πολλές φορές στην μεταφορά αρχείων μεταξύ υπολογιστή και εξωτερικού δίσκου, έτσι θεωρήσαμε πως θα είναι γνωστή σε ένα ικανοποιητικό ποσοστό. Όταν επιλεγεί η έξυπνη συσκευή που προτίθεται ο χρήστης να προσθέσει στην εικόνα της κάτοψης τότε αυτόματα εμφανίζεται ένα κόκκινο χώρος στο πίσω μέρος ο οποίος αντιπροσωπεύει το σημείο στο οποίο θα προστεθεί η έξυπνη συσκευή πάνω στην εικόνα της κάτοψης. Η λειτουργία αυτή εφαρμόζεται στο Floorspan για την τοποθέτηση των Dominos στο δισδιάστατο επίπεδο (Εικόνα 94, Εικόνα 95).



Εικόνα 85. Επιλογή του Έξυπνου Θερμομέτρου για προσθήκη στην εικόνα της κάτοψης.



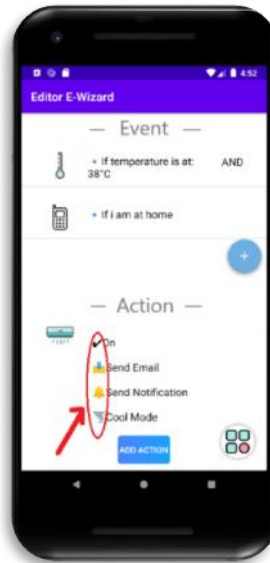
Εικόνα 86. Τωρινή εικόνα της κάτοψης μετά την προσθήκη του Έξυπνου Θερμομέτρου.

4.4.12 Emoji

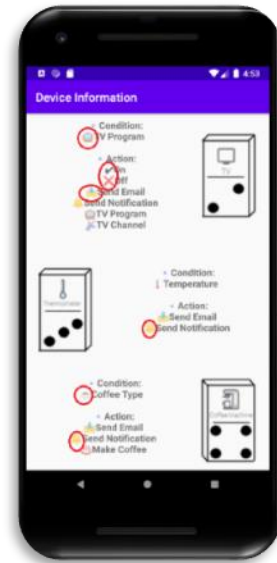
Η χρήση Emoji αποσκοπεί στο να τραβήξει το ενδιαφέρον του χρήστη και να εμπλουτίσει το κείμενο δίνοντας ποικίλα χαρακτηριστικά. Έχουμε χρησιμοποιήσει αρκετά καθώς μας άρεσε η ιδέα και δίνει έναν πιο ευχάριστο τόνο στην ανάγνωση των κανόνων. Emoji εμφανίζονται κατά τη διάρκεια δημιουργίας του κανόνα και στους δύο Editors στο κομμάτι της δράσης (Εικόνα 96, Εικόνα 97) καθώς και στην επιλογή του μενού «Πληροφορίες Έξυπνων Συσκευών» (Εικόνα 98).



Εικόνα 87. Εmojis στην αναγραφή της δράσης στον Editor Domino.



Εικόνα 88. Εmojis στην αναγραφή της δράσης στον Editor E-Wizard.



Εικόνα 89. Εmojis στις <<Πληροφορίες Έξυπνων Συσκευών>>.

4.5 Αναπαράσταση Δεδομένων σε Τρισδιάστατο Χώρο

4.5.1 Εισαγωγή

Όπως έχει αναφερθεί και σε προηγούμενη Ενότητα «Σχεδιασμός Φιλικής Διεπαφής» (Υπό-Ενότητα «Animations»), με τη χρήση του εργαλείου Unity, δημιουργήσαμε πέντε (5) Animations ικανοποίησης των κανόνων που έχουν επιλεγεί. Ο τρισδιάστατος χώρος δίνει στον χρήστη την δυνατότητα να οραματιστεί τον κανόνα που επιθυμεί και έπειτα να προβεί στον Editor Domino για την δημιουργία του **Error!**
Reference source not found..

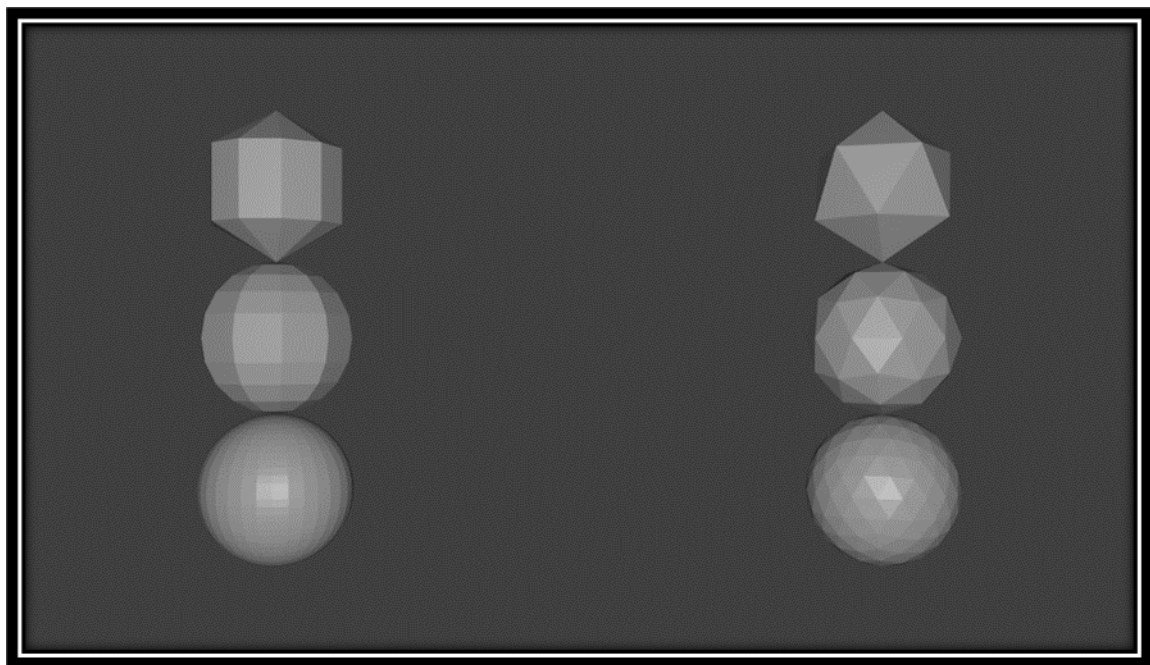
4.5.2 Περιβάλλον

Το περιβάλλον του Unity με μία πρώτη ματιά μοιάζει περίπλοκο, διότι διαθέτει πάρα πολλά εργαλεία για οποιαδήποτε ανάγκη. Εφαρμογές Κινητού Υπολογισμού, κυρίως παιχνίδια, έχουν σχεδιαστεί σε αυτό το τρισδιάστατο περιβάλλον. Θα πρέπει να έχουμε υπόψιν, πως όλες οι πληροφορίες που λαμβάνουμε μέσω ενός κινητού τηλεφώνου απεικονίζονται σε ένα δισδιάστατο πλαίσιο από pixels (Οθόνη). Παρόλα αυτά, η σχεδίαση τέτοιων απαιτήσεων υλοποιείται στο διαδραστικό τρισδιάστατο εικονικό περιβάλλον του Unity.

4.5.3 Μοντελοποίηση τρισδιάστατων Domino και Πλέγματα Αντικειμένων

Σε αυτό το στάδιο κληθήκαμε να αυξήσουμε τις χωρικές διαστάσεις των Domino από τα δισδιάστατα (Εικόνα 93 Εικόνα 93. Animation αποθηκευμένου κανόνα.) στα τρισδιάστατα. Στην μοντελοποίηση ξεκινάμε πάντοτε από την επιλογή κατάλληλου mesh (πλέγμα αντικειμένου). Τα meshes είναι προ υπάρχοντα βασικά αντικείμενα με τα οποία μπορούμε να ξεκινήσουμε την μοντελοποίηση χωρίς να τα δημιουργήσουμε. Παράδειγμα από meshes είναι:

- Επίπεδο (plane – 2D)
- Κύβος (cube – 3D)
- Κύκλος (circle – 2D)
- UV Σφαίρα (UV sphere – 3D)
- Ico Σφαίρα (Ico sphere – 3D)
- Κύλινδρος (cylinder – 3D)
- Κώνος (cone – 3D)



Εικόνα 90. Διαφορά UV Σφαίρα (Sphere) και Σφαίρα (Sphere).

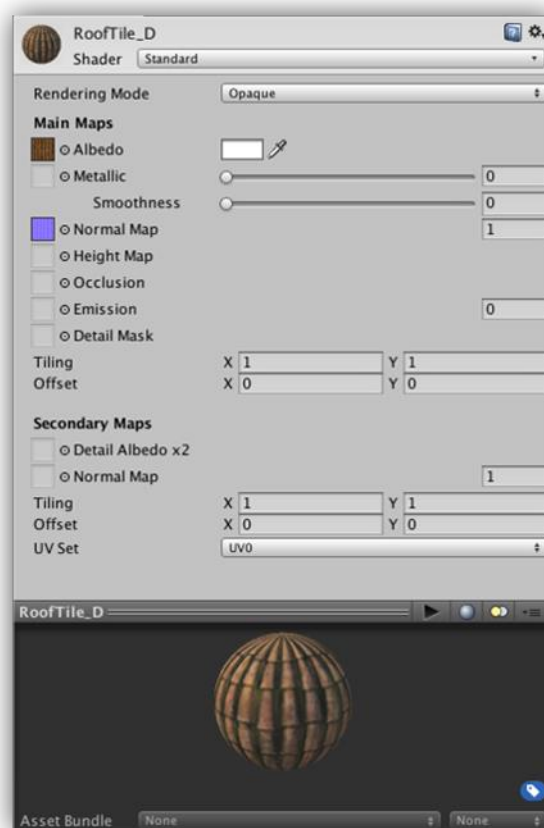
Παρατηρούμε πως η διαφορά είναι αρκετά έντονη (Εικόνα 99), καθώς αυξάνονται τα τμήματα υποδιαίρεσεων των αντικειμένων (σειρές 1 – 3). Στην σειρά 3 έχουμε πολλές υποδιαίρεσεις με αποτέλεσμα τα αντικείμενα να είναι σχεδόν παρόμοια. Η επιλογή πάντοτε εξαρτάται από το οπτικό αποτέλεσμα που θέλουμε να πετύχουμε. Για παράδειγμα για την μοντελοποίηση μίας μπάλας golf θα ήταν καλό να επιλέγαμε μια σφαίρα ICO.

4.5.4 Δημιουργία του πρώτου τρισδιάστατου Domino

Ξεκινάμε με τη χρήση του κύβου (cube mesh 3D) και όντας στην επιλογή, τροποποίησης του αντικειμένου, μπορούμε να τραβήξουμε τις πλευρές μέχρι να φτάσουμε στο επιθυμητό αποτέλεσμα (scaling). Επίσης θα χρησιμοποιήσουμε ένα επίπεδο (plane 2D) το οποίο θα τοποθετηθεί στο πρώτο μισό του Domino αναπαριστώντας την έξυπνη συσκευή.

4.5.5 Δημιουργία Material για χρήση στο 2D Plane

Τα materials – υλικά , χωρίζονται σε πολλές κατηγορίες με βάση το πως συμπεριφέρονται στο φως και στις σκιές του τρισδιάστατου χώρου (Εικόνα 100).



Εικόνα 91. RoolTile Material.

Το παραπάνω material χρησιμοποιείται για την δημιουργία των σκεπών στα τρισδιάστατα σπίτια.

Έτσι εμείς θα πρέπει να δημιουργήσουμε materials τα οποία θα αντιπροσωπεύουν το background του plane, δηλαδή του icon το οποίο προσδιορίζει την έξυπνη συσκευή.

4.5.6 Δημιουργία UV Σφαιρών για την χρήση αριθμητικών μεταβλητών στο κάτω μέρος του Domino

Προσθέτοντας σφαίρες στον χώρο και μορφοποιώντας αυτές στις κατάλληλες διαστάσεις για να θυμίζουν αριθμητικές μεταβλητές, μπορούμε να τις τοποθετήσουμε στο κάτω μέρος του Domino. Ο σκοπός για τη χρήση των σφαιρών και γενικά αριθμητικών μεταβλητών στο κάτω μέρος, είναι για την αντίληψη του χρήστη πως πρόκειται για Domino. Θα χρειαστούμε ένα μεταβαλλόμενο πλήθος με βάση την έξυπνη συσκευή.

4.5.7 Δημιουργία διαχωριστικής γραμμής

Η γραμμή η οποία χωρίζει το εικονίδιο από τις σφαίρες είναι ένας κύβος ο οποίος έχει έρθει στις κατάλληλες διαστάσεις για να θυμίζει διαχωριστική γραμμή.

4.5.8 Domino Group

Όλα τα παραπάνω, συν την προσθήκη κατάλληλου κειμένου για τον προσδιορισμό της έξυπνης συσκευής άλλα και του κανόνα / συμβάν, αποτελούν ένα σύνολο (Group) από βασικά πλέγματα αντικειμένων τα οποία συνδυάζονται για την τρισδιάστατη αναπαράσταση του 3D-Domino.

4.5.9 Δημιουργία Animation

Για να δημιουργήσουμε το Animation θα πρέπει να συμπεριλάβουμε κάποιους παράγοντες φυσικής στον χώρο μας. Τα Domino θα πρέπει να έχουν βάρος καθώς και τριβή με το έδαφος για να αναπαραστήσουμε την συνεχή ρίψη Domino. Δημιουργώντας το 3D-Domino αφού πρόκειται για ένα custom mesh μπορούμε εύκολα να προσθέσουμε αυτούς τους παράγοντες από τις ρυθμίσεις του mesh. Επίσης θα πρέπει να δημιουργηθεί και η βαρύτητα, καθώς πρόκειται για μια ακόμη επιλογή που ενεργοποιείται από τις ρυθμίσεις του χώρου μας. Τέλος θα πρέπει το πρώτο Domino ή τα πρώτα Domino (αν είναι δηλαδή η συσχέτιση AND η OR) να πέσουν. Αυτό σημαίνει πως θα πρέπει να ασκηθεί μια δύναμη στα αυτά τα Domino ικανή και αναγκαία ώστε να ξεκινήσει η ρίψη των Domino. Αυτό επιτυγχάνεται μέσω την χρήση εξωτερικού script σε γλώσσα προγραμματισμού C# το οποίο θα ενεργοποιηθεί μόλις ξεκινήσουμε την εγγραφή του animation.

4.5.10 Τοποθέτηση κάμερας και φωτισμού

Για να ξεκινήσουμε την εγγραφή του animation θα πρέπει να συμπεριλάβουμε πηγές φωτός στον χώρο μας καθώς και την κάμερα η οποία είναι υπεύθυνη για την εγγραφή.

4.5.11 Γωνίες Λήψης

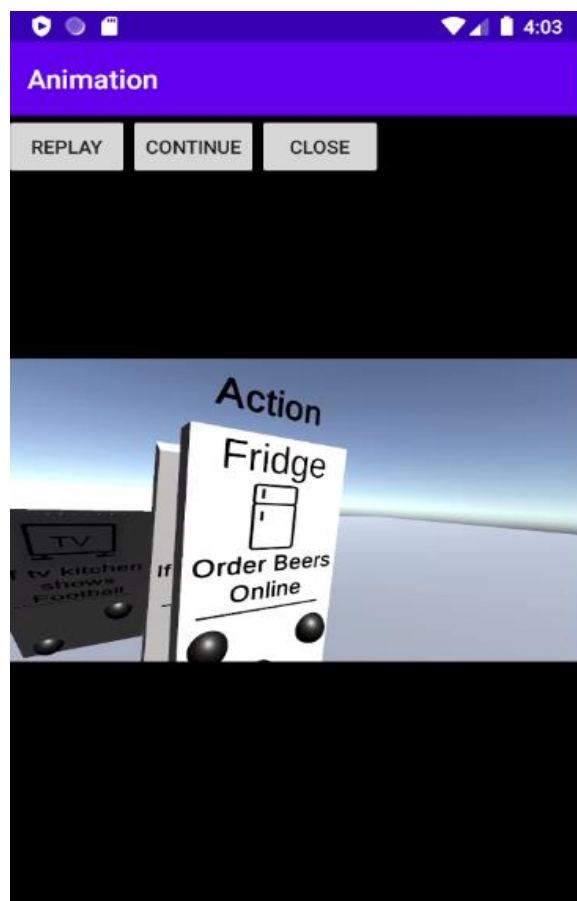
Αρκετά σημαντικές είναι οι γωνίες λήψης καθώς θα πρέπει κατά την διάρκεια του animation η κάμερα να κινείται κατά μήκος των Domino σε μια ικανή απόσταση που θα είναι εμφανές και το Domino και το συμβάν (event)-δράση (action) που περιγράφεται.

4.5.12 Μήνυμα Επιτυχίας

Στο τέλος του animation εμφανίζεται το μήνυμα επιτυχίας (“SUCCESS”) το οποίο ενημερώνει τον χρήστη πως η δράση έχει ενεργοποιηθεί καθώς παρατηρεί τα την ρίψη των Domino.

4.5.13 Τελική Μορφή Animation

Κανόνας : *If bottle (s) of beer are less than 2 **AND** if time is 19:00 **AND** if tv program is football **OR** tv program is Game of Thrones **AND** I am at home **then** fridge do online order.*



Εικόνα 92. Προβολή κανόνα σε animation μέσω της εφαρμογής.

5

Υλοποίηση Εφαρμογής Android

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα καλύψουμε τις λειτουργίες της πλατφόρμας, καθώς θα αναφερθούμε στα πιο σημαντικά σημεία της υλοποίησης. Θα συμπεριλάβουμε τμήματα κώδικα που έκαναν τις λειτουργίες να πραγματοποιηθούν καθώς και θα αναφερθούμε στις καλύτερες τεχνικές (Best – Practices) υλοποίησης κώδικα.

Best – Practices

Υπάρχουν μέθοδοι στις γλώσσες προγραμματισμού οι οποίες μπορούν να βελτιώσουν σημαντικά την απόδοση της εφαρμογής. Στην Java, τα πιο συνηθισμένα μοντέλα είναι το Singleton Pattern και το Factory Pattern.

Singleton Pattern

Σε αυτή την υλοποίηση χρησιμοποιείται μια στατική μέθοδος που επιστρέφει το αντικείμενο της κλάσης καθώς γίνεται ιδιωτικός (private) ο κατασκευαστής (constructor). Χρησιμοποιείται όταν χρειαζόμαστε μόνο ένα στιγμιότυπο αντικειμένου.

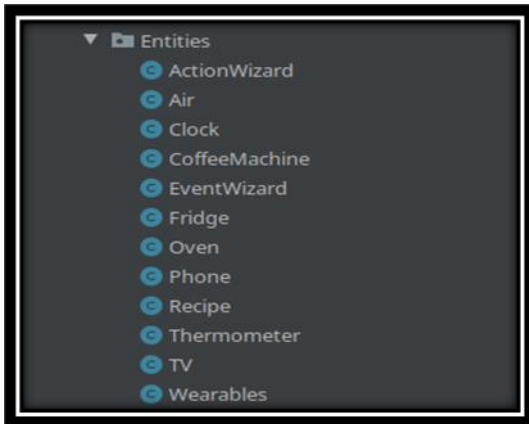
Factory Pattern

Στο συγκεκριμένο πρότυπο, χρησιμοποιείται μια κλάση ως Factory η οποία είναι υπεύθυνη για την δημιουργία αντικειμένων. Αρχικά δημιουργούμε μια διεπαφή (interface), η οποία θα χρησιμοποιηθεί από

την κάθε κλάση που θέλει να ενταχθεί στην κλάση Factory. Έπειτα οι κλάσεις κάνουν override τις μεθόδους της διεπαφής και η Factory είναι υπεύθυνη να τις δημιουργήσει.

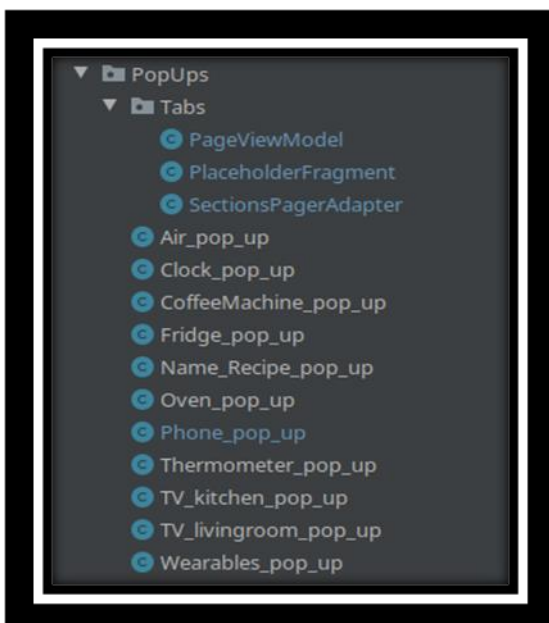
5.2 Δομή Πακέτων

Συνολικά η εφαρμογή κινητού υπολογισμού που υλοποιήσαμε διαθέτει πέντε πακέτα. Κάθε πακέτο έχει δημιουργηθεί για ξεχωριστό σκοπό και εξυπηρετεί διαφορετικούς λογούς ύπαρξης.



Εικόνα 93. Πακέτο “Entities”.

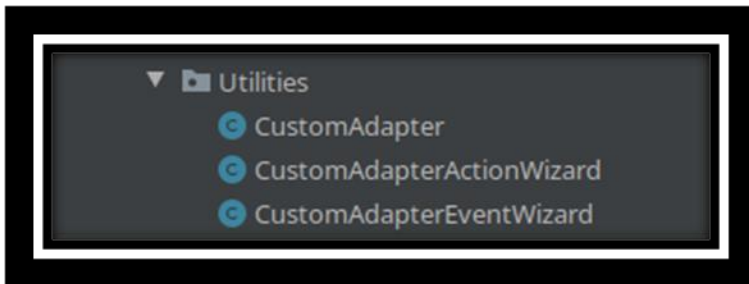
Οι οντότητες που χρησιμοποιούμε είναι όλες οι συσκευές που μπορούν να δημιουργηθούν από τον χρήστη. Δημιουργήσαμε και συμπεριλάβαμε δύο ακόμα που αφορούν το συμβάν και την δράση του Editor E-Wizard καθώς πρόκειται για δύο λίστες.



Εικόνα 94. Πακέτο “PopUps”.

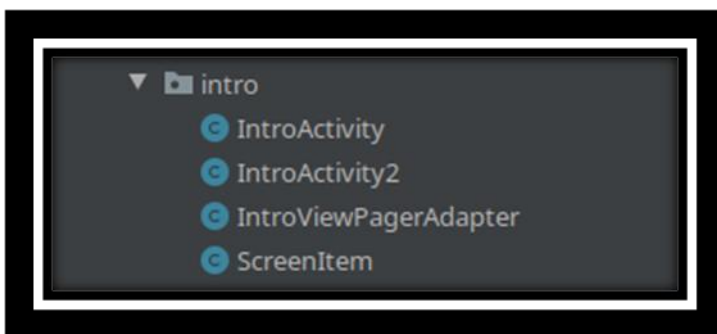
Δημιουργήσαμε αντίστοιχα και τα Pop-Up παράθυρα τα οποία είναι υπεύθυνα να προσδιορίζουν την συσκευή που πατήθηκε και να εμφανίζουν τις δυνατές ενέργειες αυτής είτε στο Action είτε το Event.

Όπως παρατηρούμε υπάρχει και το package Tabs το οποίο χρησιμοποιείται από το Phone καθώς πρόκειται για ένα πιο σύνθετο Pop-Up.



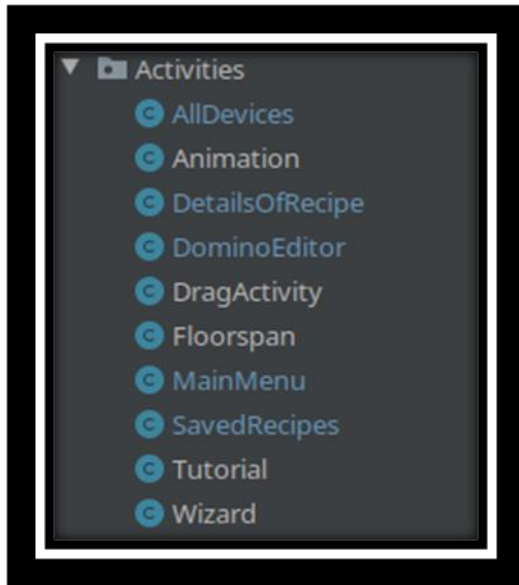
Εικόνα 95. Πακέτο “Utilities”.

Έχουμε δημιουργήσει τρεις Custom Adapters οι οποίοι χρησιμοποιούνται για την λίστα που υπάρχει στους αποθηκευμένους κανόνες και για τις λίστες του Event και Action του E-Wizard. Η προκαθορισμένη λίστα που χρησιμοποιεί το Android είναι αρκετά απλή, καθώς με την χρήση αυτών προσθέτουμε εικόνες και κείμενο δυναμικά σε προκαθορισμένα από τους Adapters σημεία πάνω στις λίστες.



Εικόνα 96. Πακέτο “intro”.

Στο συγκεκριμένο πακέτο, έχουμε δημιουργήσει δύο Intro Activities τα οποία είναι υπεύθυνα για την εκπαίδευση των χρηστών. Παρατηρούμε πως υπάρχει ακόμα ένας Adapter ο οποίος αντίστοιχα, μας δημιουργεί ένα κατάλληλο στατικό περιβάλλον στο οποίο μπορούμε να προσθέσουμε δυναμικά δεδομένα μέσω του προτύπου (template) που δημιουργούμε.



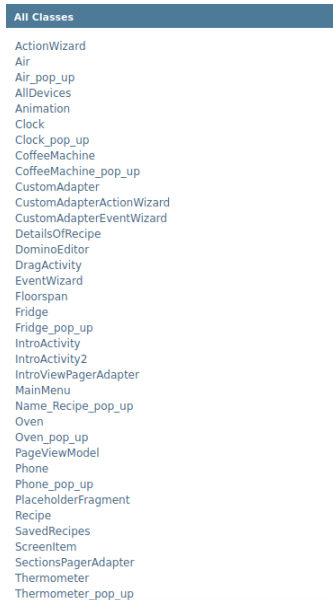
Εικόνα 97. Πακέτο “Activities”

Εδώ έχουμε συμπεριλάβει όλες τις λειτουργίες του Front-End χωρισμένες σε 10 Activities. Περιέχεται:

1. Το κεντρικό μενού.
2. Οι έξι διαθέσιμες επιλογές για τους χρήστες μέσω του μενού.
3. Η προβολή του κανόνα σε φυσική γλώσσα μέσω του SavedRecipes.
4. Προβολή του Animation μέσω του DetailsOfRecipe.
5. Ενδιάμεσο Activity για την επεξεργασία και φόρτωση της εικόνας Floorspan.

5.2.1 Javadoc

Για τις λεπτομερή ανάγνωση όλου του πηγαίου κώδικα έχουμε συμπεριλάβει ένα φάκελο με όνομα Javadoc όπου βρίσκεται το **index.html** με όλα τα δεδομένα υπό την μορφή του **Java API** σε **HTML**.



Εικόνα 98. Όλες οι κλάσεις και δραστηριότητες της εφαρμογής.

All Classes

Packages

com.example.diplomatiki.Activities
com.example.diplomatiki.Entities
com.example.diplomatiki.intro
com.example.diplomatiki.PopUps
com.example.diplomatiki.PopUps.Tabs
com.example.diplomatiki.Utilities

Εικόνα 99. Όλα τα πακέτα της εφαρμογής.

5.3 Editor Domino

5.3.1 Εισαγωγή

Για τον συγκεκριμένο Editor χρειάστηκε ένας συνδυασμός από εξωτερικές βιβλιοθήκες καθώς και τη χρήση υπάρχον που προσφέρει το Android. Χρησιμοποιήσαμε στατικές και μη μεταβλητές για να μπορούμε να αποθηκεύσουμε προσωρινά δεδομένα. Οι λειτουργίες του Editor είναι οι εξής :

- Προσθήκη Domino στο Grid
- Διαγραφή πρόσφατου Domino (Event)
- Ανανέωση ενεργειών έξυπνων συσκευών
- Προσδιορισμός Λογικής Συσχέτισης
- Αποθήκευση Κανόνα
- Επανεκκίνηση Editor

5.3.2 Προσθήκη Domino στο Grid

Για τη δημιουργία του στατικού πλαισίου χρησιμοποιήσαμε το Grid Layout σε συνδυασμό με ένα οριζόντιο και κάθετο Scroll View ώστε οι χρήστες να μπορούν ανεξαρτήτου οθόνης να περιηγηθούν σε όλο το πλαίσιο.

```
<ScrollView
    android:layout_width="409dp"
    android:layout_height="450dp"
    android:fadeScrollbars="false"
    android:scrollbarThumbVertical="@drawable/custom_scroll_style"
    android:scrollbarSize="8dp"
    android:id="@+id/sc">
    <HorizontalScrollView
        android:layout_width="409dp"
        android:layout_height="400dp"
        android:scrollbarThumbHorizontal="@drawable/custom_scroll_style"
        android:scrollbarSize="8dp"
        android:fadeScrollbars="false">
        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:rowCount="5"
            android:columnCount="5"
            android:id="@+id/row"
            android:orientation="horizontal">
        </GridLayout>
    </HorizontalScrollView>
</ScrollView>
```

Εικόνα 100. XML Κώδικας Grid Layout.

Έπειτα θα πρέπει να δημιουργηθούν οι κατάλληλες συνθήκες στον Editor για να είναι σε θέση να τοποθετεί συσκευές.

Θα πρέπει να δημιουργηθεί ένα μενού με όλες τις έξυπνες συσκευές ως Image Views αντικείμενα.



Εικόνα 101. Έξυπνες συσκευές ως ImageViews.



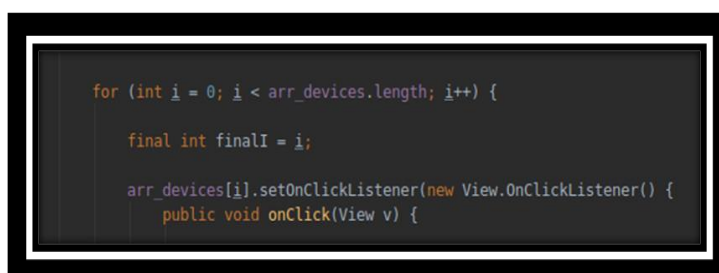
Εικόνα 102. Βασικές μεταβλητές Editor Domino.

Επίσης, εδώ βλέπουμε την δήλωση του Μενού (FlowingDrawer) που περιέχει τις συσκευές (ImageViews).

Θα πρέπει να υπάρχουν μεταβλητές προσδιορισμού της θέσης των Domino αλλά και των Shadows Dominos πάνω στο πλέγμα.

- *arrlist_coordinates_dominos* : Προσθέτουμε όλα τα Domino υπό την μορφή συντεταγμένων, π.χ. “3,2”.
- *hmap_images_coordinates* : Προσθέτουμε όλες τις έξυπνες συσκευές για την αναπαράστασή τους στο πλαίσιο.
- *hmap_texts_coordinates* : Προσθέτουμε όλες τις αναπαραστάσεις είτε συμβάντος είτε δράσης σε φυσική γλώσσα μέσω των TextViews, καθώς και τις συντεταγμένες του Domino που αντιστοιχούν.
- *all_domino_images* : Όλα τα Domino που έχουν προστεθεί στο grid.
- *allShadows* : Προσθέτουμε όλες τις πιθανές τοποθετήσεις συσκευών στο πλέγμα (Shadows) καθώς και τις συντεταγμένες αυτών.

Δημιουργία Listeners σε κάθε αντικείμενο ImageView που υπάρχει στο μενού:



Εικόνα 103. Listener αντικειμένων τύπου ImageView.

Το πρώτο domino που θα επιλεγεί θα προστεθεί στο Action, καθώς θα τοποθετηθεί και ένα Shadow Domino στο πλαίσιο του Event το οποίο είναι υπεύθυνο να αναδείξει την πιθανή θέση τοποθέτησης συσκευής.

Αφού γίνει κλικ το Shadow Domino και είναι η δεύτερη φορά που θα επιλεγεί συσκευή από το μενού, τότε αυτή η συσκευή θα αντικαταστήσει το Shadow στο πλαίσιο.

5.3.3 Διαγραφή Πρόσφατου Domino

Θα πρέπει να ξέρουμε εάν η συσκευή που πατήθηκε ήταν η πιο πρόσφατη, εάν ήταν τότε μπορούμε να προβούμε στον μηχανισμό της διαγραφής.

- `current_domino`: Domino που πατήθηκε, τύπου `ImageView`
- `delete`: εάν είναι εφικτή η διαγραφή, τύπου `boolean`.

```
current_domino.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View view) {

        if(delete){
            AlertDialog.Builder builder = new AlertDialog.Builder(context: MainActivity.this);
            builder.setMessage("Add one in order to delete again")
                .setTitle("You can delete one Domino at a time!")
                .setCancelable(true);

            AlertDialog alert = builder.create();
            alert.show();
        }else {
            AlertDialog.Builder builder = new AlertDialog.Builder(context: MainActivity.this);
            builder.setMessage("Tap DELETE to proceed")
                .setTitle("Are you sure you would like to delete this Domino?")
                .setCancelable(false)
                .setNegativeButton("Delete",
                    (dialog, id) -> {

                        String coords = returnCoordFromImageView(hmap_images_coordinates, current_domino);
                        String x_y[] = coords.split(regex: " ");
                        System.out.println(x_y[0]+", "+x_y[1]);

                        delete = deleteCurrentDomino(current_domino);
                        Log.e(tag: "after DEL: ",allShadows.toString());
                        Log.e(tag: "after DEL DOM: ",arlist_coordinates_domino.toString());

                    });
            AlertDialog alert = builder.create();
            alert.show();
        }
        return true;
    }
});
```

Εικόνα 104. Κώδικας διαγραφής domino.

```
public boolean deleteCurrentDomino(ImageView current_domino) {
```

Εικόνα 105. Υπογραφή μεθόδου διαγραφής domino.

Παρατηρούμε από την υπογραφή της μεθόδου πως η παράμετρος που χρειαζόμαστε είναι το Domino που έχει πατηθεί εκείνη τη στιγμή.

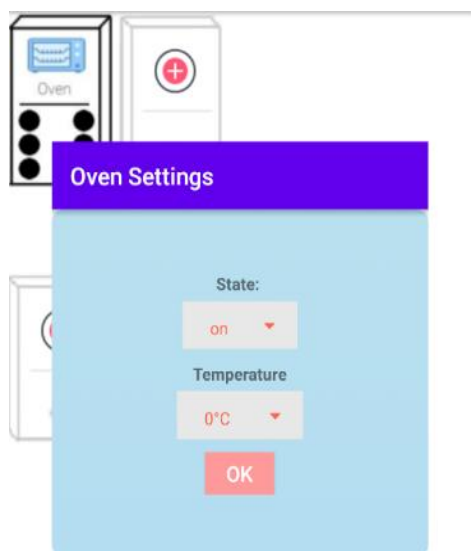
Έπειτα αρκεί μια αναζήτηση στις θέσεις των Dominos καθώς και των Shadows για να βρεθεί το υπάρχον Domino με τα Shadows που το συνοδεύουν. Μόλις βρεθεί αντικατασταίτε με Shadow καθώς αφαιρούνται τα δεξιά και κάτω Shadows.

Προσοχή: Εάν το Domino που πατήθηκε έχει δεξιά ή κάτω από αυτό άλλο Domino τότε θα πρέπει να προσέξουμε να μην διαγράψουμε αυτές τις συντεταγμένες.

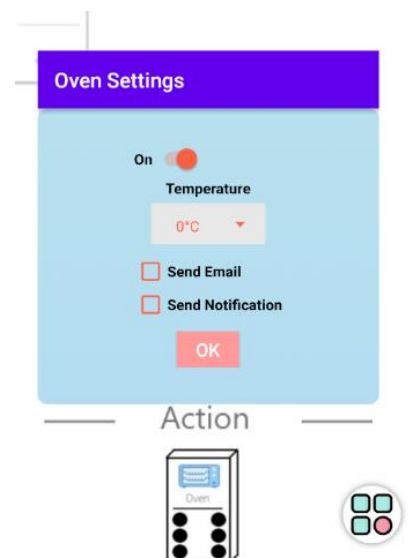
Στην περίπτωση που η διαγραφή είναι εφικτή, η μέθοδος επιστρέφει true και ταυτόχρονα διαγράφει και το Domino, διαφορετικά επιστρέφει false.

5.3.4 Ανανέωση Ενέργειών Έξυπνων Συσκευών

Έχουμε φτιάξει δύο μεθόδους που είναι υπεύθυνες για την εκκίνηση των αντιστοίχων λειτουργιών του Domino. Θα πρέπει να γνωρίζουμε σε ποια θέση είναι το Domino που πατήθηκε, καθώς τα Pop-Ups του συμβάντος και της δράσης διαφέρουν.



Εικόνα 106. Pop - up παράθυρο συμβάντος έξυπνου φούρνου.



Εικόνα 107. Pop - up παράθυρο δράσης έξυπνου φούρνου.

Για τον σκοπό αυτό, υλοποιήσαμε τις δύο μεθόδους:

- public void Action () μέθοδο
- public void Trigger () μέθοδο

οι οποίες είναι υπεύθυνες για τον χειρισμό του Editor. Με βάση τη συσκευή που πατήθηκε η κάθε μέθοδος αναζητεί τη συσκευή αυτή και προβάλλει μέσω του Pop-up παραθύρου τα δεδομένα της σωστής συσκευής.

5.3.5 Προσδιορισμός Λογικής Συσχέτισης

Με βάση το υπάρχον πλαίσιο οι συσκευές τοποθετούνται κάθετα και οριζόντια στο πλαίσιο. Δημιουργήσαμε έναν αλγόριθμο ο οποίος είναι υπεύθυνος να δημιουργεί τον κανόνα σε φυσική γλώσσα προσθέτοντας τις φράσεις «AND» και «OR» με βάση την θέση τους στο πλαίσιο.

Γνωρίζοντας την θέση των Domino στο πλαίσιο μπορούμε να υπολογίσουμε εάν το Domino αυτό βρίσκεται μόνο του οριζόντια ή κάθετα.

Ο αλγόριθμος είναι ως εξής:

1. Φτιάξε μια συμβολοσειρά (που θα περιέχει όλον τον κανόνα σε φυσική γλώσσα)
2. Για κάθε σειρά στο πλαίσιο:
 1. Πρόσθεσε το Event του πρώτου Domino στην συμβολοσειρά
 2. Εάν βρεις κι άλλο πρόσθεσε την λέξη **OR** και το Event του Domino
 3. Εάν δεν υπάρχει άλλο άλλαξε γραμμή και:
 1. Εάν υπάρχει Domino στην σειρά αυτή, πρόσθεσε στην συμβολοσειρά τη λέξη **AND** και το Event του Domino. Συνέχισε στο βήμα 2.
 2. Εάν δεν υπάρχει άλλο Domino στη σειρά αυτή πρόσθεσε τη λέξη **THEN** στη συμβολοσειρά και το Action του Domino.

5.3.6 Αποθήκευση Κανόνα

Οι υπολογισμοί που αναφέρθηκαν παραπάνω θα γίνουν μέσω της μεθόδου *saveRule (...)* μόλις ο χρήστης επιλέξει να αποθηκεύσει τον κανόνα που έχει φτιάξει έως τώρα.

Ο χρήστης καλείται να δώσει ένα όνομα για τον κανόνα αυτό και να τον αποθηκεύσει. Σε αυτό το σημείο προγραμματιστικά γίνονται οι εξής ενέργειες:

1. Κλήση της μεθόδου *saveRule ()*.
2. Καθορισμός εικονιδίου δημιουργίας κανόνα, σε αυτή τη περίπτωση είναι το icon του Editor Domino.
3. Ανακατεύθυνση στη λίστα με τους αποθηκευμένους κανόνες.

5.3.7 Επανεκκίνηση Editor

Ο χρήστης έχει και την επιλογή να πραγματοποιήσει επανεκκίνηση σε όλον τον Editor ξεκινώντας τον από την αρχή. Σε αυτή τη περίπτωση πρέπει να γίνουν οι διαγραφές όλων των τοποθετημένων συσκευών στο πλαίσιο καθώς και να διαγραφούν και όλα τα Pop-Ups με τις αποθηκευμένες ρυθμίσεις.

Καλώντας την μέθοδο *resetAction ()* για το Action και κάνοντας **clear ()** το Map με όλα τα αποθηκευμένα instances των συσκευών για το Event, ανανεώνεται ο Editor.

Σχεδιαστικά, θα πρέπει να γίνει κλήση αντίστοιχης μεθόδου για να επιτυγχάνουμε σωστή κελυφοποίηση δεδομένων όπως έγινε για το Action.

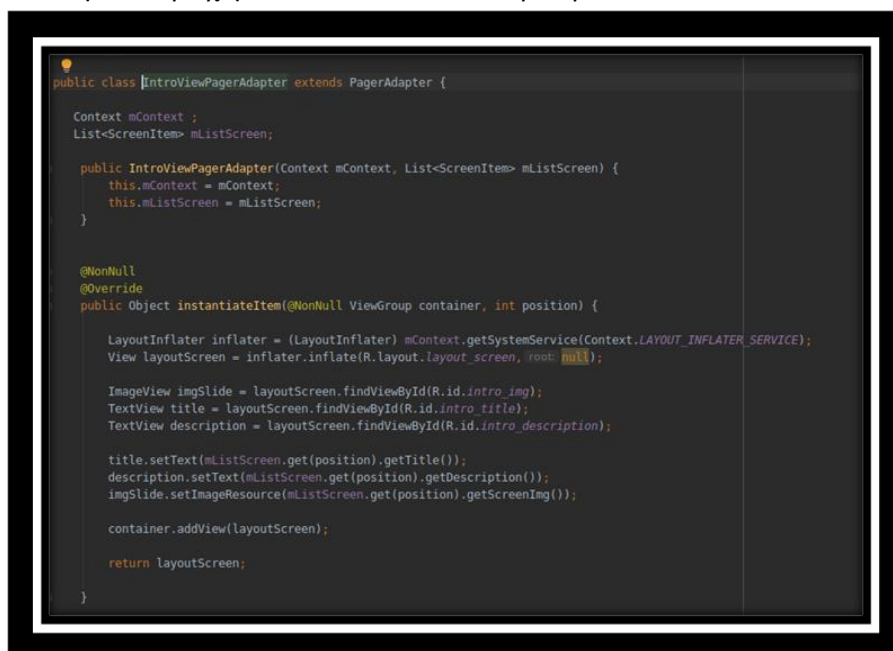


Εικόνα 108. Κώδικας επανεκκίνησης Editor Domino.

5.4 Εκπαίδευση στους Editors

Για το κομμάτι της εκπαίδευσης έχουμε χρησιμοποιήσει ένα Activity με δύο Buttons. Με βάση την επιλογή του χρήστη ξεκινούμε ένα Intro Activity.

Χρησιμοποιήσαμε μια κλάση η οποία κληρονομεί τον **PagerAdapter** με αποτέλεσμα να μπορούμε να δημιουργήσουμε δυναμικό περιεχόμενο πάνω σε ένα σταθερό πρότυπο.

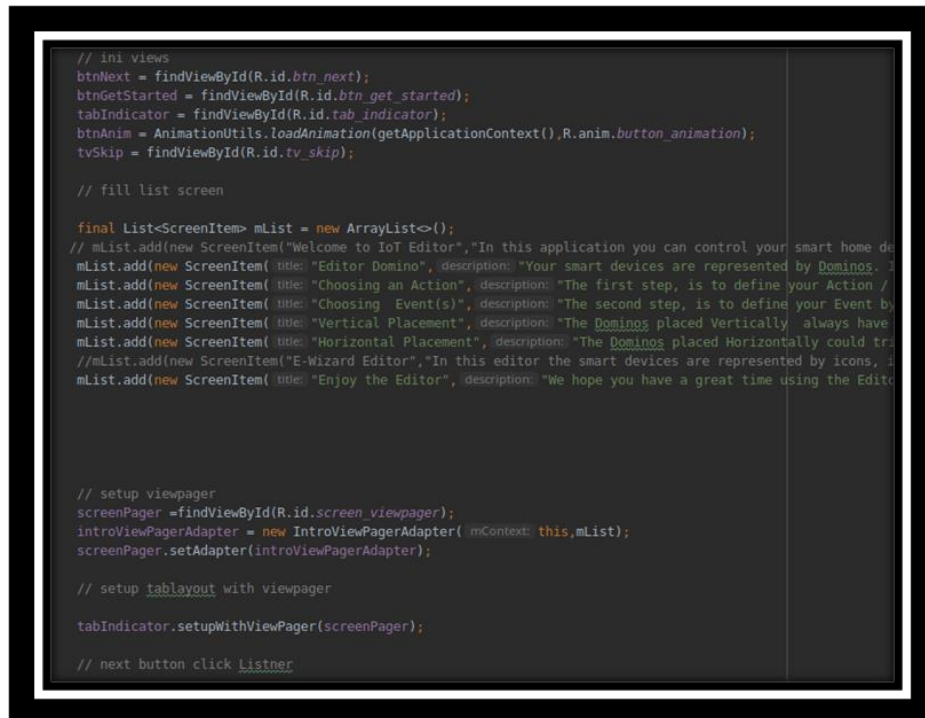


Εικόνα 109. Κώδικας δημιουργίας προτύπου.

Όπως βλέπουμε το σταθερό πρότυπο φτιάχνεται από ένα *ImageView* και δύο *TextViews* τα οποία είναι ο τίτλος και ο υπότιτλος του *IntroActivity*.

5.4.1 Δημιουργία Activity για χρήση του Προτύπου

Έπειτα δημιουργούμε το Activity και περνάμε σε μια λίστα τις τρεις παραμέτρους που προαναφέρθηκαν.



Εικόνα 110. Κώδικας για τη χρήση του προτύπου.

Μπορούμε να προσθέσουμε επιπλέον στατικά δεδομένα στο Activity όπως φαίνεται και στην παραπάνω εικόνα, όπως είναι το Next Button το οποίο θα πηγαίνει στο επόμενο αντικείμενο της λίστας καθώς και το Skip Button που είναι υπεύθυνο να μας πάει στο τελευταίο αντικείμενο της λίστας.



Εικόνα 111. Οπτικό αποτέλεσμα κώδικα "Εκπαίδευση στους Editors".

```
for(int i=0;i<64;i++)
{
    LinearLayout myLinear= new LinearLayout(getApplicationContext());
    myLinear.setOnDragListener(new MyDragListener());
    myLinear.setBackgroundResource(R.drawable.shape);
    floorspan.addView(myLinear);
    myLinear.getLayoutParams().height=120;
    myLinear.getLayoutParams().width=120;
}
```

Εικόνα 112. Πρώτο μέρος κώδικα "Drag Activity".

Εάν δούμε προσεκτικά το πρώτο αντικείμενο της λίστας, θα δούμε πως αναγράφεται ο τίτλος, ο υπότιτλος καθώς και το icon.

5.5 Κάτοψη (Floorspan)

5.5.1 Drag Activity

Υλοποιείται όλη η λειτουργία του Drag and Drop των Dominos καθώς και το resize της ανεβασμένης εικόνας έτσι ώστε ανεξαρτήτου μεγέθους εικόνα να χωρέσει στο πλέγμα. Το πλέγμα που χρησιμοποιήσαμε πρόκειται για ένα Grid Layout το οποίο αρχικοποιείται δυναμικά με 64 Linear Layouts τα οποία περιέχουν τις πιθανές τοποθετήσεις των Dominos. Θέτοντας το background του Grid Layout στην εικόνα που επιθυμεί ο χρήστης καταφέρνουμε να δημιουργήσουμε το αναδραστικό πλαίσιο τοποθέτησης των εικονιδίων.

Παρακάτω βλέπουμε πως επιτυχάναμε την δημιουργία του πλαισίου, αφού έχουμε ορίσει το background με τη χρήση της μεθόδου `setBackground()`.

```
String pathName = getApplicationContext().getFilesDir()+"/Floorspan_user.png";
Drawable floorImage = Drawable.createFromPath(pathName);
floorspan.setBackground(floorImage);
```

Εικόνα 113. Δεύτερο μέρος κώδικα "Drag Activity".

5.5.2 Floorspan Activity και Android Permissions

Είναι σημαντικό να αναφέρουμε πως θα χρειαστούμε την άδεια του χρήστη για να προβούμε σε επεξεργασία της εικόνας.

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Εικόνα 114. Δήλωση άδειας στο Android Manifest.

Καθώς και τον τρόπο που το ζητούμε από τον χρήστη μόλις χρησιμοποιηθεί το κουμπί Upload.

```
upload.setOnClickListener((v) -> {  
  
    Intent photoPickerIntent = new Intent(Intent.ACTION_PICK);  
    photoPickerIntent.setType("image/*");  
    startActivityForResult(photoPickerIntent, requestCode: 1);  
  
});
```

Εικόνα 115. Κώδικας αιτήματος άδειας χρήστη.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
  
    if (resultCode == RESULT_OK) {  
        try {  
            final Uri imageUri = data.getData();  
            final InputStream imageStream = getContentResolver().openInputStream(imageUri);  
            final Bitmap selectedImage = BitmapFactory.decodeStream(imageStream);  
            floorspan.setImageBitmap(selectedImage);  
  
            File imageFile = new File(getApplicationContext().getFilesDir(), child: "Floorspan_user.png");  
            FileOutputStream fos = null;  
            try {  
                fos = new FileOutputStream(imageFile);  
                selectedImage.compress(Bitmap.CompressFormat.PNG, quality: 100, fos);  
                fos.close();  
            }  
            catch (IOException e) {  
                Log.e( tag: "app", e.getMessage());  
                if (fos != null) try {  
                    fos.close();  
                } catch (IOException e1) {  
                    e1.printStackTrace();  
                }  
            }  
            Toast.makeText( context: Floorspan.this, getApplicationContext().getFilesDir().toString(), Toast.LENGTH_LONG).show();  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
            Toast.makeText( context: Floorspan.this, text: "Something went wrong", Toast.LENGTH_LONG).show();  
        }  
    }  
}
```

Εικόνα 116. Κώδικας διαγραφής άδειας χρήστη.

Θα χρειαστεί η χρήση της μεθόδου `onActivityResult()` για να είμαστε σε θέση να διαχειριστούμε την άδεια που μας έχει δώσει ο χρήστης

Παρατηρούμε πως σε περίπτωση που το resultCode είναι 1 (**RESULT_OK**) μπορούμε να αναθέσουμε την εικόνα που έχει επιλεγεί στο Grid Layout. Η ανάγνωση και αποθήκευση της εικόνας επιτυγχάνονται με τη χρήση ροών.

5.6 Editor E – Wizard

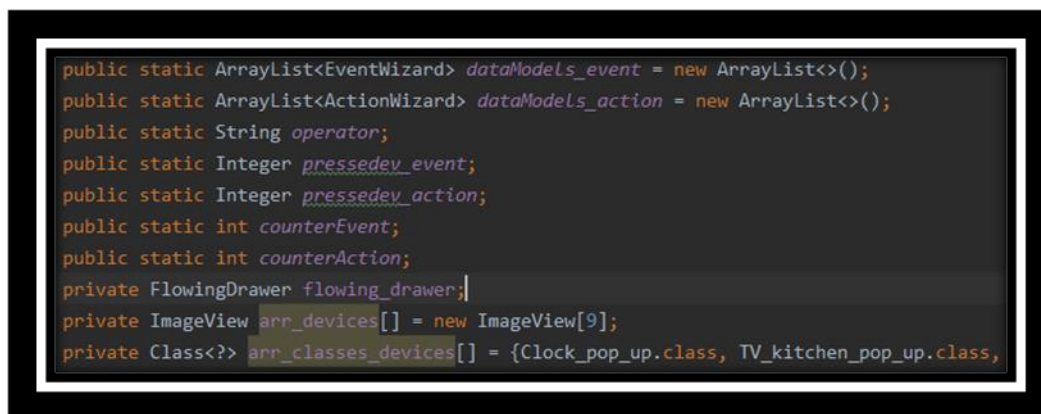
5.6.1 Εισαγωγή

Ο αριθμός γραμμών του κώδικα που έχει συνολικά η δραστηριότητα (activity) που αφορά τον Editor E-Wizard είναι, περίπου, 680 γραμμές. Ο κώδικας περιλαμβάνει όλες τις κύριες λειτουργίες που διαθέτει ο Editor E-Wizard. Οι λειτουργίες αυτές είναι :

- Άνοιγμα/Κλείσιμο του μενού έξυπνων συσκευών
- Τοποθέτηση έξυπνης συσκευής σε κανόνα :
 - ο Τοποθέτηση έξυπνης συσκευής στο μέρος του συμβάντος
 - ο Τοποθέτηση έξυπνης συσκευής στο μέρος της δράσης
- Διαγραφή/Ανανέωση έξυπνης συσκευής είτε στο μέρος της δράσης είτε στο μέρος του συμβάντος
- Προσθήκη λογικών τελεστών «OR» και «AND» στον κανόνα
- Αποθήκευση/Επανεκκίνηση Editor

5.6.2 Βασικές Μεταβλητές για την λειτουργία του E – Wizard

Οι βασικές μεταβλητές που χρησιμοποιούνται για την ορθή λειτουργία του E-Wizard είναι οι παρακάτω :



```
public static ArrayList<EventWizard> dataModels_event = new ArrayList<>();
public static ArrayList<ActionWizard> dataModels_action = new ArrayList<>();
public static String operator;
public static Integer pressedev_event;
public static Integer pressedev_action;
public static int counterEvent;
public static int counterAction;
private FlowingDrawer flowing_drawer;
private ImageView arr_devices[] = new ImageView[9];
private Class<?> arr_classes_devices[] = {Clock_pop_up.class, TV_kitchen_pop_up.class,
```

Εικόνα 117. Βασικές μεταβλητές Editor E-Wizard.

- *dataModels_event* : Διατηρεί όλες τις έξυπνες συσκευές που ορίζονται ως συμβάντα, τις ενέργειες και τους τελεστές αυτών.
- *dataModels_action* : Διατηρεί όλες τις έξυπνες συσκευές που ορίζονται ως δράσεις και τις ενέργειες αυτών.
- *operator* : Αναλόγως την επιλογή του χρήστη η συγκεκριμένη μεταβλητή αρχικοποιείται με «AND» ή «OR».

- *pressdev_event* : Μεταβλητή η οποία χρησιμοποιείται για την επιλογή της διαγραφής ή της ανανέωσης μιας έξυπνης συσκευής στο μέρος των συμβάντων. Αρχικοποιείται με την θέση στην λίστα στην οποία βρίσκεται η έξυπνη συσκευή που πρόκειται να διαγραφεί ή ανανεωθεί.
- *pressdev_action* : Χρησιμοποιείται για του ίδιου λόγους που χρησιμοποιείται η μεταβλητή *pressdev_event* με την διαφορά ότι αφορά τις έξυπνες συσκευές που βρίσκονται στο κομμάτι της δράσης.
- *counterEvent* : Μετρητής ο οποίος αυξάνεται ανάλογα με τις έξυπνες συσκευές στο μέρος των συμβάντων.
- *counterAction* : Μετρητής ο οποίος αυξάνεται ανάλογα με τις έξυπνες συσκευές στο μέρος της δράσης.
- *Flowing_drawer* : Μενού έξυπνων συσκευών.
- *arr_devices[]* : Πίνακας ο οποίος περιέχει τα εικονίδια των έξυπνων συσκευών.
- *arr_classes_devices* : Πίνακας ο οποίος περιέχει τις δραστηριότητες (activities) για κάθε μία έξυπνη συσκευή.

5.6.3 Άνοιγμα/Κλείσιμο του μενού έξυπνων συσκευών

Όπως είδαμε στην Ενότητα «4.3.3 Editor E - Wizard» το άνοιγμα του μενού των έξυπνων συσκευών μπορεί να γίνει με τη χρήση τεσσάρων κουμπιών :

- Κουμπί «ADD EVENT»
- Κουμπί «ADD ACTION»
- Κουμπί «OR»
- Κουμπί «AND»

Οι χρήστες δεν μπορούν να ανοίξουν το μενού με οριζόντια κύλιση από αριστερά προς τα δεξιά διότι δεν επιθυμούσαμε να πραγματοποιείται ο συγκεκριμένος τρόπος ανοίγματος για το μενού. Ο λόγος είναι ότι όταν άνοιγε το μενού μ' αυτόν τον τρόπο και επέλεγε μια έξυπνη συσκευή υπήρχε κόλλημα της εφαρμογής διότι οι εικόνες κάθε έξυπνης συσκευής με την αφή πάνω σε μια από αυτές πραγματοποιούσε μια ενέργεια προκαθορισμένη εάν πρωτίστως είχε πατηθεί κάποιο από τα τέσσερα παραπάνω κουμπιά που αναφέρθηκαν. Έτσι, η εφαρμογή δεν γνώριζε που επρόκειτο να τοποθετηθεί αυτή η έξυπνη συσκευή στο μέρος του συμβάντος ή στο μέρος της δράσης. Το πρόβλημα αυτό λύθηκε, όπως αναφέρθηκε, καθώς το μενού πλέον ανοίγει μόνο με τα τέσσερα παραπάνω κουμπιά. Ωστόσο, οι χρήστες έχοντας ανοιχτό το μενού μπορούν να το κλείσουν κάνοντας οριζόντια κύλιση από δεξιά προς τα αριστερά αλλά για να το ξανά ανοίξουν θα πρέπει να πατήσουν ένα από τα τέσσερα παραπάνω κουμπιά. Παρακάτω φαίνεται ο κώδικας για το άνοιγμα και το κλείσιμο του μενού των έξυπνων συσκευών (Εικόνα 106).



Εικόνα 118. Κώδικας ανοίγματος/κλεισίματος μενού έξυπνων συσκευών.

Όπως παρατηρείται το αντικείμενο που αναπαριστά το μενού ονομάζεται *flowing_drawer* και είναι αντικείμενο της ομώνυμης εξωτερικής κλάσης *Flowing Drawer* η οποία βρίσκεται σε εξωτερική βιβλιοθήκη. Διαθέτει έναν listener ο οποίος περιέχει δύο μεθόδους όπου σε κάθε αλλαγή της κατάστασης του μενού καλούνται. Η πρώτη μέθοδος *onDrawerStateChange ()* παίρνει ως παραμέτρους δύο τιμές, την *oldState* και την *newState*. Η πρώτη παράμετρος αφορά την προηγούμενη κατάσταση απ' αυτήν που βρίσκεται τώρα το μενού και η δεύτερη παράμετρος την τωρινή κατάσταση. Η δεύτερη μέθοδος με όνομα *onDrawerSlide ()* έχει, κι αυτή, δύο παραμέτρους, η *openRatio* και η *offsetPixels*. Και οι δύο αυτές παράμετροι επιστρέφουν μαθηματικές τιμές σχετικά με το ποσοστό που είναι ανοιχτό το μενού στην οθόνη της εφαρμογής. Στην προκειμένη περίπτωση οι αλλαγές καταστάσεων που έχουμε προσθέσει είναι οι παρακάτω :

- STATE_OPEN = Μενού ανοιχτό
- STATE_CLOSED = Μενού κλειστό
- STATE_OPENING = Το μενού είναι σε διαδικασία ανοίγματος
- STATE_CLOSING = Το μενού είναι σε διαδικασία κλεισίματος

Αναλόγως την κατάσταση στην οποία βρίσκεται το μενού πραγματοποιούνται και οι αντίστοιχες λειτουργίες.

5.6.4 Τοποθέτηση έξυπνης συσκευής στο μέρος του συμβάντος του κανόνα

Ο κώδικας που αφορά την τοποθέτηση έξυπνων συσκευών στον κανόνα με τη χρήση Editor E-Wizard βρίσκεται μέσα σε έναν listener του πίνακα που περιέχει τα εικονίδια όλων των έξυπνων συσκευών. Ο

listener αναλόγως με την έξυπνη συσκευή που έχει επιλεγθεί να τοποθετηθεί, αναγνωρίζει την θέση της έξυπνης συσκευής στον πίνακα και κατά αυτό τον τρόπο αναγνωρίζει για ποια έξυπνη συσκευή πρόκειται. Με τις δύο μεταβλητές `boolean`, `flag_event` και `flag_action` πραγματοποιεί έλεγχο για το μέρος που ανήκει η επιλεγμένη έξυπνη συσκευή στον κανόνα. Αν η μεταβλητή `flag_event` είναι αληθής (`true`) η έξυπνη συσκευή πρόκειται να τοποθετηθεί στο κομμάτι των συμβάντων και αν η `flag_action` είναι αληθής η έξυπνη συσκευή πρόκειται να τοποθετηθεί στο κομμάτι των δράσης. Οπότε πραγματοποιούνται και οι αντίστοιχες ενέργειες.

Όσον αφορά τον κώδικα τοποθέτησης έξυπνων συσκευών στο μέρος του συμβάντος ακολουθεί τα ίδια προκαθορισμένα βήματα για όλες τις έξυπνες συσκευές.

```
    } else if (flag_event) {
        presseddev_event = null;
        flowing_drawer.closeMenu(animate: true);
        fab.setVisibility(View.VISIBLE);
        btnEvent.setVisibility(View.INVISIBLE);
        selectedEventDevice = finalI;
        try {
            Trigger(arr_classes_devices[finalI].newInstance());
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        } catch (InstantiationException e) {
            e.printStackTrace();
        }
        ListViewEvent.setOnItemLongClickListener((parent, view, position, arg3) -> {
            presseddev_event = position;
            AlertDialog.Builder builder = new AlertDialog.Builder(context: Wizard.this);
            builder.setMessage("Update / Delete selected Event")
                .setTitle("Update / Delete Smart Device")
                .setCancelable(false)
                .setNegativeButton("Delete",
                    (dialog, id) -> {
                        adapter_event.remove(adapter_event.getItem(position));
                        adapter_event.notifyDataSetChanged();
                        arrListEvents.remove(position);
                        deleteEvent(position);
                        counterEvent--;
                        if (position == dataModels_event.size()) {
                            if (dataModels_event.size() != 0) {
                                dataModels_event.get(dataModels_event.size() - 1).setVisibility(false);
                                operator = null;
                            }
                        }
                        if (dataModels_event.size() == 0) {
                            btnEvent.setVisibility(View.VISIBLE);
                            fab.setVisibility(View.INVISIBLE);
                            operator = null;
                        }
                    })
                .setPositiveButton("Update",
                    (dialog, id) -> {
                        Trigger(arrListEvents.get(presseddev_event));
                    })
                .create();
            AlertDialog alert = builder.create();
            alert.show();
            return true;
        });
        counterEvent++;
    }
```

Εικόνα 119. Κώδικας τοποθέτησης έξυπνησης συσκευής στο κομμάτι του συμβάντος στον Editor E-Wizard.

Το πρώτο βήμα είναι ο ορισμός της μεταβλητής `presseddev_event` σε `null` έτσι ώστε (να υπάρχει σιγουριά ότι) σε κάθε επανάληψη που πραγματοποιείται η μεταβλητή δεν έχει κρατήσει κάποια παλιότερη τιμή η οποία θα πραγματοποιήσει (κάποιες) ανεπιθύμητες, (σ' αυτήν την φάση), ενέργειες.

Έπειτα ακολουθεί το κλείσιμο του μενού το οποίο αποτελείται από τις εξής ενέργειες :

1. Εμφάνιση του κουμπιού `fab` το οποίο εμπεριέχει τα κουμπιά «OR» και «AND»
2. Απομάκρυνση του κουμπιού «ADD EVENT» διότι πλέον η προσθήκη έξυπνων συσκευών στο κομμάτι του συμβάντος θα γίνεται μέσω των κουμπιών «OR» και «AND»

3. Ορισμός της μεταβλητής *selectedEventDevice* με την τιμή του *finalI*, όπου το *finalI* περιέχει την θέση στην οποία βρίσκεται η έξυπνη συσκευή στον πίνακα άρα την έξυπνη συσκευή που επιλέχθηκε.

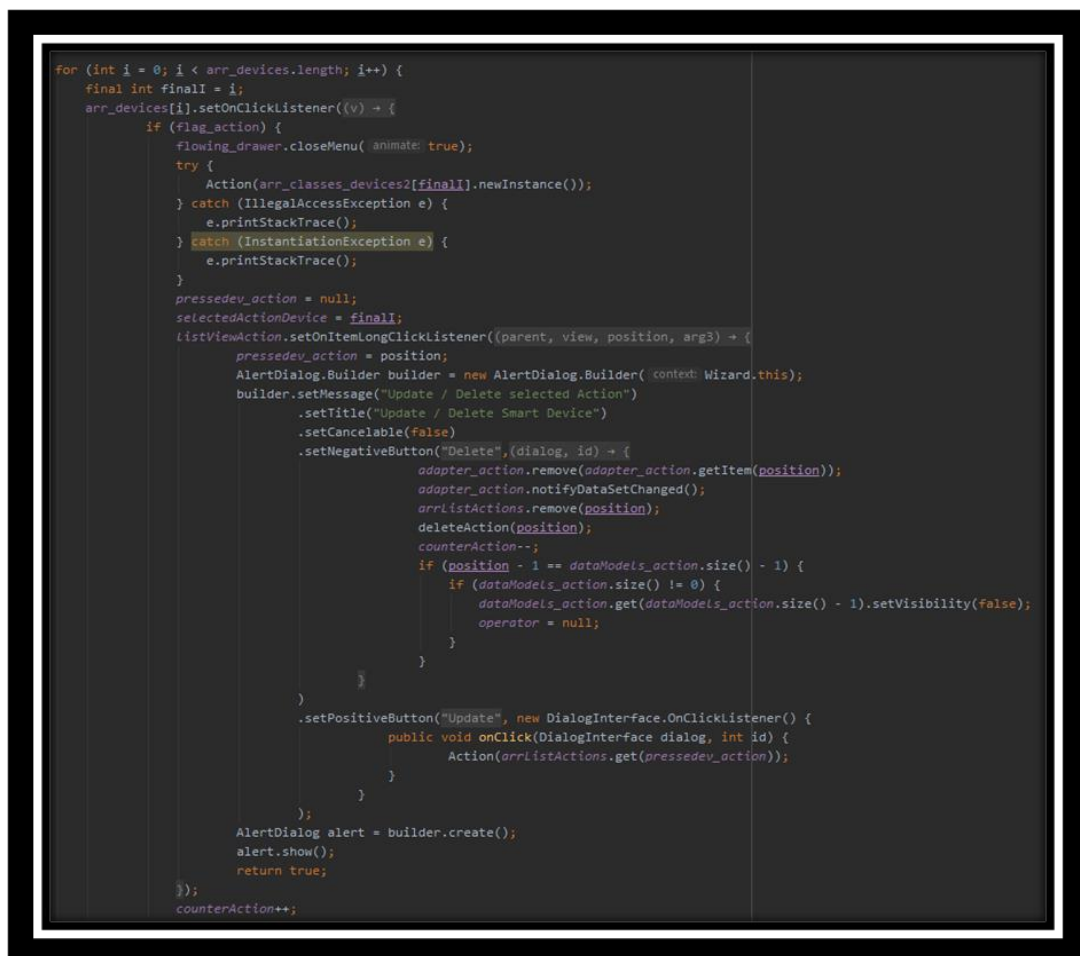
Στην συνέχεια, καλείται η μέθοδος *Trigger ()* η οποία παίρνει μια παράμετρο, την έξυπνη συσκευή, και θα πραγματοποιήσει τις ενέργειες που απαιτούνται για την τελική προσθήκη της έξυπνης συσκευής στο κομμάτι του συμβάντος στον κανόνα.

Σε περίπτωση που οι χρήστες επιθυμούν να πραγματοποιήσουν διαγραφή ή ανανέωση κάποιας έξυπνης συσκευής που υπάρχει ήδη στο κομμάτι του συμβάντος στον κανόνα τότε καλείται ο listener της *listviewEvent* όπου αναλόγως την επιθυμία χρήστη πραγματοποιεί τις αντίστοιχες ενέργειες.

Εάν επιλέξει διαγραφή καλείται η εσωτερική μέθοδος *setNegativeButton (...)*, εάν επιλέξει ανανέωση καλείται η εσωτερική μέθοδος *setPositiveButton (...)*.

5.6.5 Τοποθέτηση έξυπνης συσκευής στο μέρος της δράσης του κανόνα

Όπως και στον κώδικα τοποθέτησης έξυπνης συσκευής στο μέρος του συμβάντος έτσι και στο μέρος της δράσης τα βήματα που ακολουθούνται είναι τα ίδια για κάθε έξυπνη συσκευή.



Εικόνα 120. Κώδικας τοποθέτησης έξυπνης συσκευής στο κομμάτι της δράσης στον Editor E-Wizard.

Στο σημείο αυτό γίνεται εμφανής ο listener του πίνακα που περιέχει τα εικονίδια όλων των έξυπνων συσκευών όπου αρχικά ξεκινάει η επανάληψη σύμφωνα με το μέγεθος του πίνακα. Στην προκειμένη περίπτωση το μέγεθος του πίνακα είναι 9 και μόλις βρεθεί η έξυπνη συσκευή που επιλέχθηκε πραγματοποιείται ο έλεγχος σε ποιο μέρος του κανόνα πρόκειται να τοποθετηθεί.

Στην αρχή του κώδικα εκτελείται η εντολή που πυροδοτεί το κλείσιμο του μενού των έξυπνων συσκευών και κατόπιν καλείται η μέθοδος *Action ()* με παράμετρο, την έξυπνη συσκευή, η οποία θα πραγματοποιήσει τις αντίστοιχες ενέργειες που απαιτούνται για την προσθήκη της στο κομμάτι της δράσης.

Η μεταβλητή *pressdev_action* που είναι η αντίστοιχη της μεταβλητής *pressdev_event* γίνεται κι αυτή null για τους ίδιους λόγους που αναφέρθηκαν παραπάνω για το *pressdev_event*.

Σε περίπτωση που οι χρήστες επιθυμούν να πραγματοποιήσουν διαγραφή ή ανανέωση κάποιας έξυπνης συσκευής που υπάρχει ήδη στο κομμάτι του συμβάντος στον κανόνα, τότε καλείται ο listener της *listviewAction*. Αναλόγως την επιθυμία χρήστη καλείται ο listener που βρίσκεται στις εσωτερικές μεθόδους *setNegativeButton (...)*, *setPositiveButton (...)* για διαγραφή και ανανέωση.

5.6.5.1 Κώδικας μεθόδου *Trigger (...)*

Η μέθοδος *Trigger ()* όπως αναφέρθηκε και παραπάνω δέχεται μία παράμετρο, το αντικείμενο της έξυπνης συσκευής. Έπειτα, διατρέχει τους ελέγχους που απαιτούνται έτσι ώστε να βρεθεί ποιας κλάσης αντικείμενο είναι η συγκεκριμένη έξυπνη συσκευή. Οποιασδήποτε κλάσης να είναι αντικείμενο, ακολουθούνται οι ίδιες ενέργειες.

Πραγματοποιείται κλήση της μεθόδου *setDevice (...)*, στέλνει την έξυπνη συσκευή ως αντικείμενο στην αντίστοιχη κλάση και εμφανίζει το pop-up παράθυρο της αντίστοιχης κλάσης. Η μέθοδος *setDevice (...)* έχει δύο παραμέτρους, η πρώτη μπορεί να λάβει την τιμή 0 ή 1, το 0 είναι ο προκαθορισμένος αριθμός που αφορά το κομμάτι του συμβάντος και το 1 είναι το κομμάτι δράσης του κανόνα.

Στη μέθοδο *setDevice (...)* η πρώτη καθώς και η δεύτερη παράμετρος παραμένουν σταθερές σε οποιαδήποτε μπλοκ της *if* και αν καλούνται. Η δεύτερη παράμετρος μπορεί να λάβει την τιμή Wizard ή Domino. Η τιμή της αλλάζει σύμφωνα με τον Editor στον οποίο είναι οι χρήστες.

```
public void Rule(Object selectedDevice) {  
    if (selectedDevice instanceof Clock) {  
        Clock_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[0]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof TV) {  
        TV_kitchen_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[1]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof Thermometer) {  
        Thermometer_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[2]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof CoffeeMachine) {  
        CoffeeMachine_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[3]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof Fridge) {  
        Fridge_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[4]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof Oven) {  
        Oven_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[5]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof Air) {  
        Air_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[6]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof Phone) {  
        Phone_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[7]);  
        startActivity(intent);  
    } else if (selectedDevice instanceof Wearables) {  
        Wearables_pop_up.setDevice( dev: 0, edic: "Wizard");  
        Intent intent = new Intent(getApplicationContext(), arr_classes_devices[8]);  
        startActivity(intent);  
    }  
}
```

Εικόνα 121. Κώδικας μεθόδου Trigger(...).

5.6.5.2 Κώδικας μεθόδου Action (...)

Ο κώδικας της μεθόδου *Action (...)* είναι παρόμοιος με αυτόν της μεθόδου *Trigger (...)*. Η δομή και οι ενέργειες που πραγματοποιούνται είναι ίδιες και το μοναδικό σημείο στο οποίο διαφέρουν είναι στην τιμή της πρώτης παραμέτρου της μεθόδου *setDevice (...)* όπου στην προκειμένη περίπτωση είναι 1.

5.6.6 Διαγραφή/Ανανέωση έξυπνης συσκευής

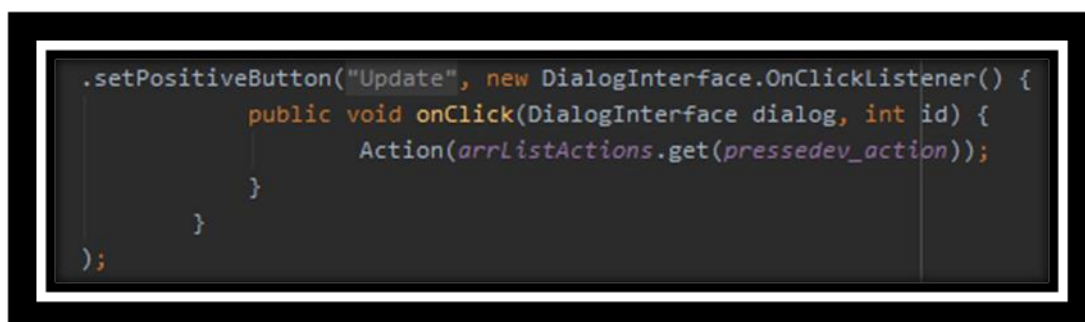
Κατά τη δημιουργία του κανόνα οι έξυπνες συσκευές που προστίθενται είτε στο κομμάτι της δράσης είτε στο κομμάτι του συμβάντος διατηρούνται σε μια αντίστοιχη λίστα. Κύριος σκοπός είναι η εύρεση της θέσης της έξυπνης συσκευής που επιλέχθηκε για τη διαγραφή και η απομάκρυνση της από το αντίστοιχο κομμάτι του κανόνα. Ο κώδικας διαγραφής πραγματοποιείται για κάποια έξυπνη συσκευή που βρίσκεται στο κομμάτι της δράσης είτε για κάποια συσκευή που βρίσκεται στο κομμάτι του συμβάντος. Το μόνο που αλλάζει είναι η λίστα από την οποία αφαιρείται η έξυπνη συσκευή.



Εικόνα 122. Κώδικας διαγραφής έξυπνης συσκευής που ανήκει στις δράσεις στον Editor E-Wizard.

Αρχικά, η μέθοδος *deleteAction* (...) λαμβάνει μία ακέραια παράμετρο η οποία αναφέρεται στον στη θέση της έξυπνης συσκευής στη λίστα *arrListActions*. Έπειτα διατρέχεται η λίστα *arrListActions* έτσι ώστε να βρεθεί η κλάση στην οποία ανήκει η έξυπνη συσκευή που πρόκειται να διαγραφεί από το μέρος της δράσης του κανόνα. Εφ' όσον βρεθεί και διαγραφεί, τότε η θέση της επόμενης έξυπνης συσκευής απ' αυτήν μειώνεται κατά ένα. Ο κώδικας για την διαγραφή στο κομμάτι του συμβάντος είναι ο ίδιος με μοναδική αλλαγή ότι χρησιμοποιείται η λίστα με όνομα *arrListEvents*.

Η λειτουργία της ανανέωσης μιας έξυπνης συσκευής που υπάρχει ήδη στη λίστα των συμβάντων ή των δράσεων πραγματοποιείται με την επανάκληση της μεθόδου *Trigger* (...) ή *Action* (...) αλλά με παράμετρο την μεταβλητή *pressdev_event* ή *pressdev_action* αντίστοιχα. Οι δύο μεταβλητές λειτουργούν ως δείκτες των αντικειμένων που υπάρχουν στις δύο λίστες. Μ' αυτόν τον τρόπο γίνεται γνωστή η έξυπνη συσκευή που επιλέχθηκε.



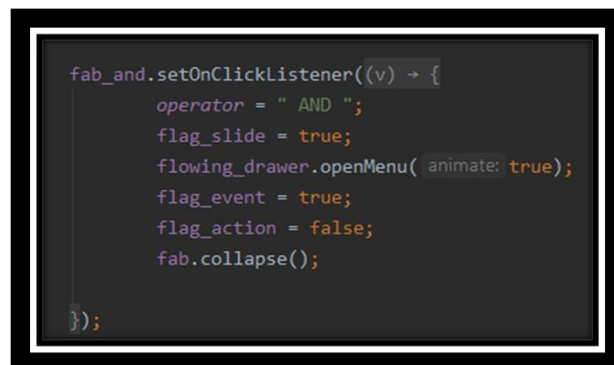
Εικόνα 123. Κώδικας ανανέωσης έξυπνης συσκευής που βρίσκεται στις δράσεις στον Editor E-Wizard.



Εικόνα 124. Κώδικας ανανέωσης έξυπνης συσκευής που βρίσκεται στα συμβάντα στον Editor E-Wizard.

5.6.7 Προσθήκη λογικών τελεστών OR και AND στον κανόνα

Η προσθήκη λογικών τελεστών στον κανόνα όπως προαναφέρθηκε γίνεται με την χρήση των κουμπιών «AND» και «OR». Ο τρόπος με τον οποίο προστίθενται στον κανόνα εξαρτάται, αρχικά, από πιο κουμπί θα πατήσει ο χρήστης. Αν πατήσει το κουμπί «OR» τότε η μεταβλητή operator θα πάρει την τιμή «OR», αν πατήσει το κουμπί «AND» τότε η μεταβλητή operator θα πάρει την τιμή «AND». Κάθε έξυπνη συσκευή που προστίθεται ως συμβάν στον κανόνα μαζί με τις ενέργειες που πραγματοποιεί προστίθεται και ο τελεστής που έχει επιλεγεί. Αυτό συμβαίνει σε περίπτωση που υπάρχουν περισσότερα από μία έξυπνες συσκευές στο κομμάτι των συμβάντων.

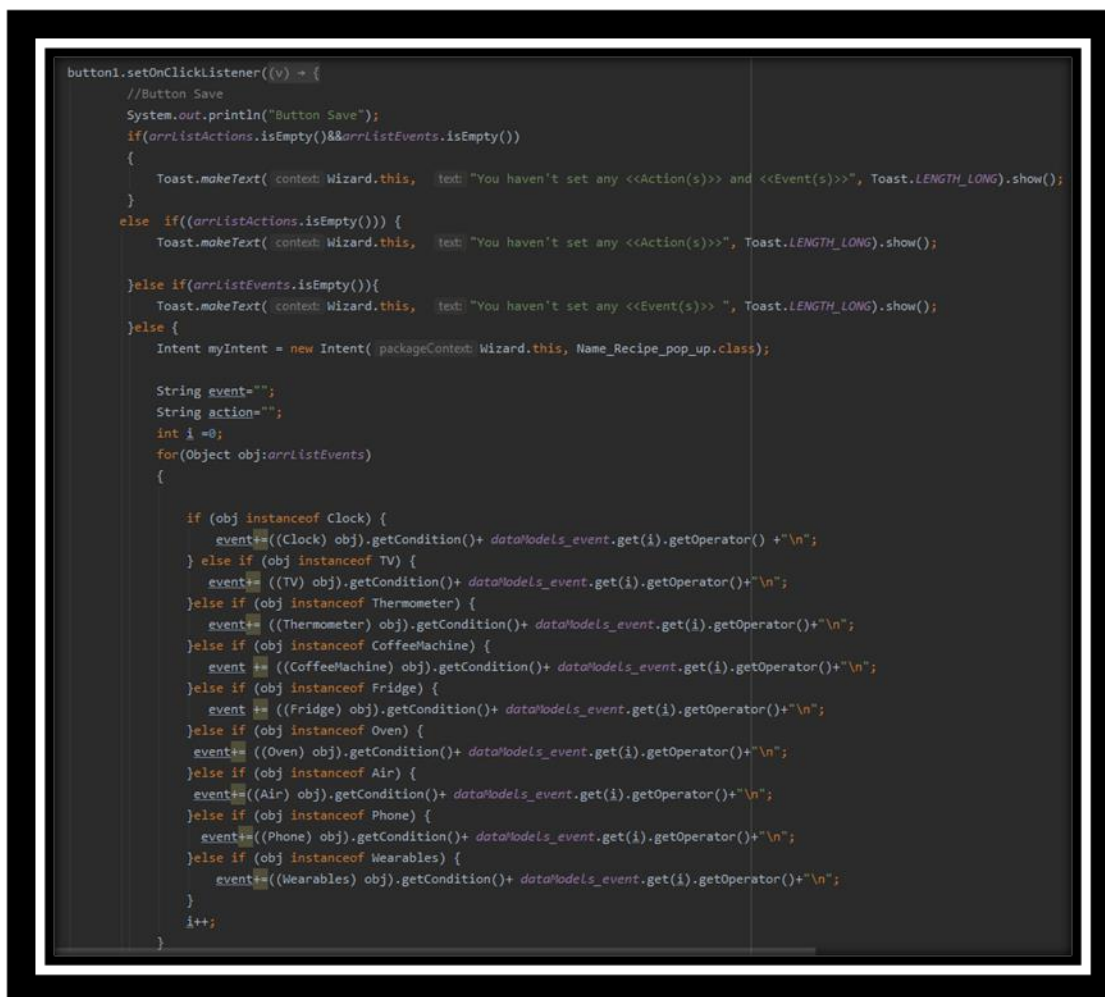


Εικόνα 125. Κώδικας αρχικοποίησης μεταβλητής operator στο κουμπί AND.

Στην περίπτωση που πατηθεί το κουμπί «OR» η μεταβλητή operator αλλάζει σε «OR». Το υπόλοιπο τμήμα του κώδικα παραμένει ως έχει.

5.6.8 Αποθήκευση/Επανεκκίνηση Editor

Όταν μια έξυπνη συσκευή προστίθεται στον κανόνα γίνεται έλεγχος για το κομμάτι του κανόνα που πρόκειται να προστεθεί. Αν προστεθεί στο κομμάτι των συμβάντων τότε η έξυπνη συσκευή και οι ενέργειες τις προστίθενται στην λίστα `dataModels_events`, διαφορετικά προστίθενται στην λίστα `dataModels_action`. Όταν ο χρήστης επιλέξει να αποθηκεύσει τον κανόνα πραγματοποιούνται οι παρακάτω ενέργειες και για τις δύο αυτές λίστες.



Εικόνα 126. Κώδικας αποθήκευσης συμβάντων στον Editor E-Wizard.

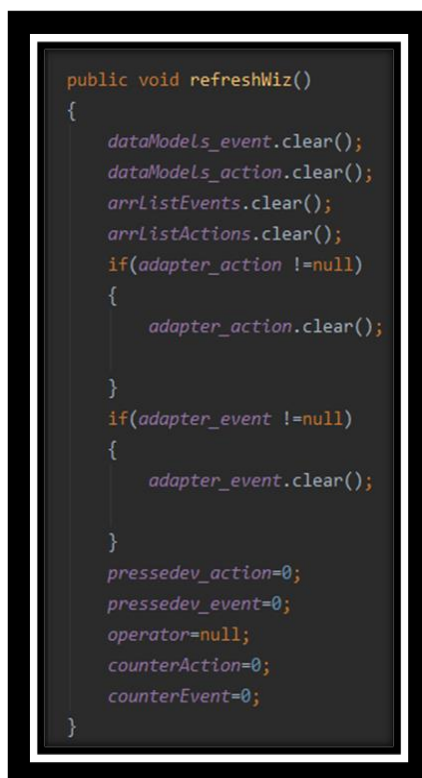
Αρχικά, γίνεται έλεγχος ο οποίος αποσκοπεί να εξετάσει αν οι δύο λίστες περιέχουν, τουλάχιστον, ένα αντικείμενο έξυπνης συσκευής. Έπειτα, γίνεται έλεγχος ξεχωριστά για κάθε μία λίστα έτσι ώστε να επιβεβαιωθεί ότι υπάρχει συμβάν και δράση στον κανόνα. Εφ' όσον όλα τα παραπάνω ισχύουν τότε προχωράει η διαδικασία της αποθήκευσης.

Οι δύο μεταβλητές τύπου *String* με όνομα *event* και *action* θα λάβουν τις ενέργειες κάθε έξυπνης συσκευής που βρίσκεται στον κανόνα. Η πρώτη για τις ενέργειες των έξυπνων συσκευών που αφορούν τα συμβάντα και η δεύτερη που αφορούν τις δράσεις.

Ύστερα, διατρέχεται η λίστα που αφορά τα συμβάντα. Με βάσει το είδος της έξυπνης συσκευής, προστίθεται στην μεταβλητή *event* η ενέργεια αυτής μαζί με τον τελεστή που έχει επιλεγεί. Η διαδικασία αυτή ολοκληρώνεται όταν δεν υπάρχουν πλέον έξυπνες συσκευές στο κομμάτι των συμβάντων. Σε περίπτωση που η έξυπνη συσκευή δεν έχει κάποιον operator, δηλαδή έχουμε μία έξυπνη συσκευή στο κομμάτι του συμβάντος, τότε η συνάρτηση *getOperator()* επιστρέφει null και η διαδικασία τερματίζεται.

Ομοίως, γίνεται η διαδικασία της αποθήκευσης των ενεργειών των έξυπνων συσκευών που βρίσκονται στο κομμάτι της δράσης του κανόνα. Η διαφορά είναι ότι διατρέχεται η λίστα *dataModels_action*. Σε περίπτωση που υπάρχουν περισσότερες από μία δράσεις τότε προστίθεται, αυτόματα, ο λογικός τελεστής «AND» για να τις συνδέσει.

Ο κώδικας της επανεκκίνησης καλύπτεται απ' την μέθοδο που έχει δημιουργηθεί και ονομάζεται *refreshWiz ()*. Η συγκεκριμένη μέθοδος εκτελεί όλες τις απαραίτητες ενέργειες έτσι ώστε να εξασφαλίσει ότι ο Editor E-Wizard είναι έτοιμος για επαναχρησιμοποίηση από την αρχή.

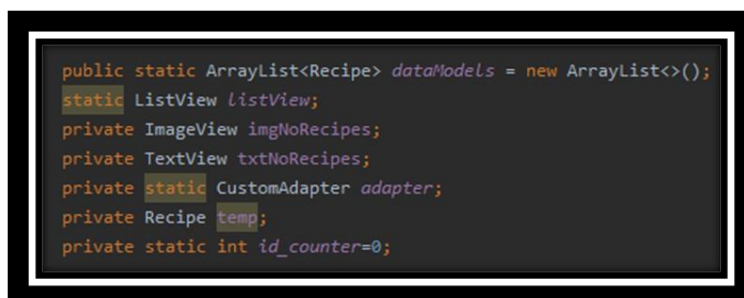


```
public void refreshWiz()
{
    dataModels_event.clear();
    dataModels_action.clear();
    arrListEvents.clear();
    arrListActions.clear();
    if(adapter_action !=null)
    {
        adapter_action.clear();
    }
    if(adapter_event !=null)
    {
        adapter_event.clear();
    }
    pressedev_action=0;
    pressedev_event=0;
    operator=null;
    counterAction=0;
    counterEvent=0;
}
```

Εικόνα 127. Κώδικας επανεκκίνησης στον Editor E-Wizard.

5.7 Αποθηκευμένοι Κανόνες

Οι χρήστες αφού ολοκληρώσουν την δημιουργία κανόνα μπορούν να προβούν στην αποθήκευση του. Είτε έχουν δημιουργηθεί στον Editor Domino είτε στον Editor E-Wizard οι κανόνες αποθηκεύονται σε μια κοινή λίστα. Παρακάτω αναφέρονται οι βασικές μεταβλητές που χρησιμοποιεί η συγκεκριμένη δραστηριότητα για την αποθήκευση ενός κανόνα.



```
public static ArrayList<Recipe> dataModels = new ArrayList<>();
static ListView listView;
private ImageView imgNoRecipes;
private TextView txtNoRecipes;
private static CustomAdapter adapter;
private Recipe temp;
private static int id_counter=0;
```

Εικόνα 128. Βασικές μεταβλητές αποθήκευσης κανόνων.

- *dataModels* : Λίστα στην οποία προστίθενται οι αποθηκευμένοι κανόνες.
- *listView* : Γραφικό στοιχείο λίστας.
- *imgNoRecipes* : Γραφικό στοιχείο εικόνας η οποία εμφανίζεται σε περίπτωση που δεν υπάρχει κανένας αποθηκευμένος κανόνας.
- *txtNoRecipes* : Γραφικό στοιχείο κειμένου το οποίο εμφανίζεται σε περίπτωση που δεν υπάρχει κανένας αποθηκευμένος κανόνας.
- *adapter* : Μεταβλητή η οποία χρησιμοποιείται για την ανάθεση δυναμικού περιεχομένου σε στατικό πρότυπο.
- *temp* : Προσωρινή μεταβλητή της κλάσης Recipe η οποία περιλαμβάνει βασικά στοιχεία ενός κανόνα και αλλάζει σε κάθε προσθήκη νέου κανόνα. Περιέχει το όνομα του κανόνα, τα συμβάντα, τις δράσεις, τον editor στον οποίο δημιουργήθηκε καθώς και μια μεταβλητή που αναφέρεται σε εικονίδιο.
- *id_counter* : Μεταβλητή η οποία ορίζει μοναδικά ids στους αποθηκευμένους κανόνες.

Στην αρχή του κώδικα λαμβάνονται τα αντίστοιχα γραφικά στοιχεία για την αρχικοποίηση των μεταβλητών *listView*, *imgNoRecipes* και *txtNoRecipes*.



Εικόνα 129. Κώδικας αποθήκευσης κανόνων.

Κατόπιν, γίνεται έλεγχος ο οποίος αποσκοπεί στο να ελέγξει από ποια δραστηριότητα ο χρήστης κατέφθασε στο activity που περιλαμβάνει τους αποθηκευμένους κανόνες. Υπάρχουν δύο περιπτώσεις για να μεταβεί στην δραστηριότητα με τους αποθηκευμένους κανόνες, είτε μέσω του μενού είτε μέσω της αποθήκευσης κάποιου κανόνα. Στην πρώτη περίπτωση δεν θα πρέπει να εκτελεστούν οι πέντε γραμμές κώδικα που βρίσκονται στο μπλοκ της `if` διότι ο χρήστης θέλει να δει τους αποθηκευμένους κανόνες. Στην δεύτερη περίπτωση πρέπει να εκτελεστούν διότι πρόκειται για έναν κανόνα που δημιούργησε και θέλει να αποθηκεύσει. Εφ' όσον είναι αληθής ο έλεγχος τότε δημιουργείται το αντίστοιχο αντικείμενο κανόνα με πεδία το όνομα, τα συμβάντα, τις δράσεις, το εικονίδιο και τον editor του κανόνα και προστίθεται στην λίστα με τους υπόλοιπους κανόνες. Με αυτόν τον τρόπο,

αυξάνεται ο μετρητής και εξαφανίζονται τα δύο γραφικά στοιχεία *imgNoRecipes*, *txtNoRecipes* από την δραστηριότητα.

Υστερα, δημιουργείται ο adapter ο οποίος θα ορίσει την δομή που θα εμφανίζονται οι αποθηκευμένοι κανόνες και έπειτα ορίζεται η λίστα που θα περιέχει τους κανόνες να έχει την συγκεκριμένη δομή.

Τέλος, γίνεται έλεγχος της λίστας ένα είναι κενή έτσι ώστε να εμφανιστούν ή να εξαφανιστούν τα δύο γραφικά στοιχεία *imgNoRecipes* και *txtNoRecipes*.

5.8 Πληροφορίες Έξυπνων Συσκευών

Ο κώδικας των πληροφοριών των έξυπνων συσκευών που παρουσιάζει στους χρήστες της ενέργειες των έξυπνων συσκευών έχει απλή δομή. Για κάθε έξυπνη συσκευή ορίζονται οι αντίστοιχες μεταβλητές για τις πληροφορίες στο μέρος των συμβάντων και για το μέρος της δράσης.



Εικόνα 130. Κώδικας πληροφοριών έξυπνων συσκευών.

Αρχικά, λαμβάνεται το αντίστοιχο γραφικό στοιχείο για μία συγκεκριμένη έξυπνη συσκευή και έπειτα της εκχωρούνται οι ενέργειες της αντίστοιχης έξυπνης συσκευής που την αφορά σε μορφή απλού κειμένου. Στην προκειμένη περίπτωση έχουμε δύο παραδείγματα για τις έξυπνες συσκευές ρολόι και τηλεόραση. Κατά αυτόν τον τρόπο ορίζονται όλες οι πληροφορίες των έξυπνων συσκευών για την συγκεκριμένη επιλογή από το μενού.

5.9 Κώδικας Αναγραφής κανόνα σε φυσική γλώσσα

Μέσω της επιλογής του χρήστη από τη λίστα με τις αποθηκευμένες συσκευές μπορεί να δει τον κανόνα που έχει δημιουργήσει σε φυσική γλώσσα. Πρόκειται για ένα απλά δομημένο Activity με τις επιλογές του Delete και Animation Buttons. Για τον σκοπό αυτό αρκεί να πάρουμε το επιλεγμένο Recipe από την λίστα και να το εμφανίσουμε δυναμικά στο layout.

```
final Recipe temp = (Recipe) getIntent().getSerializableExtra( name: "selected_recipe");

ImageView img_domino= findViewById(R.id.item_recipe_image);
TextView condition=findViewById(R.id.condition);
TextView action = findViewById(R.id.action);
Button btnDelete = findViewById(R.id.btnDelete);
Button btnAnimation = findViewById(R.id.btnAnimation);

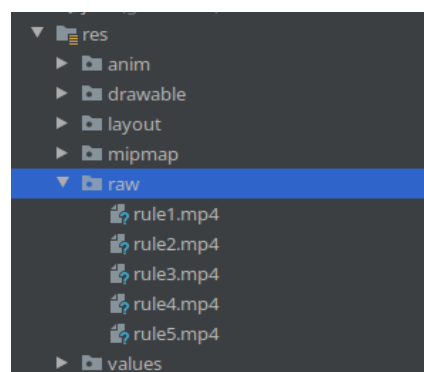
img_domino.setBackgroundResource(temp.getDrawable());
condition.setText(temp.getCondition());
action.setText(temp.getAction());
```

Εικόνα 131. Κώδικας αναγραφής κανόνα σε φυσική γλώσσα.

5.10 Κώδικας Animation

Ο κώδικας που αφορά το Animation είναι άμεσα συνδεδεμένος με τον κώδικα της αναγραφής κανόνα σε φυσική γλώσσα καθώς στην περίπτωση που ο χρήστης επιλέξει το Animation θα πρέπει να τον πάμε σε ένα νέο Activity με τις επιλογές Play, Pause, Continue και Stop για να είναι σε θέση να διαχειριστή το βίντεο.

Για αυτό τον σκοπό δημιουργήσαμε ένα νέο πακέτο με όνομα “raw” όπου εκεί βρίσκονται τα Animations που φτιάξαμε από το Unity σε μορφή mp4.



Εικόνα 132. Φάκελος "animation" στον κώδικα Android.

Στο Activity του Animation χρησιμοποιούμε ένα XML Element που ονομάζεται VideoView. Με τη χρήση τις μεθόδου `setVideoUri ()` αρχικοποιούμε το video player και με τις μεθόδους `start ()` και `pause ()` μπορούμε να επέμβουμε στην ροή.

```
public void replayVideo(View v){
    pause.setText("PAUSE");
    video.start();
}

public void pauseVideo(View v){
    if(pause.getText().toString().equalsIgnoreCase("Pause")){
        video.pause();
        pause.setText("Continue");
    }else if(pause.getText().toString().equalsIgnoreCase("Continue")){
        video.start();
        pause.setText("Pause");
    }
}

public void closeVideo(View v) { finish(); }
```

Εικόνα 133. Μέθοδοι Animation στον κώδικα Android.

5.11 Επαναχρησιμοποιήσιμα τμήματα κώδικα

Θεωρούμε επαναχρησιμοποιήσιμα τμήματα κώδικα, διάφορα tasks ή κλάσεις που χρησιμοποιούνται από πολλά Activities. Οι κλάσεις που εμπεριέχονται στο πακέτο Entities χρησιμοποιούνται και από τους δύο Editors. Οι Custom Adapters του πακέτου Utilities που έχουμε συμπεριλάβει προσδίδουν επαναχρησιμοποίηση του προτύπου οπουδήποτε αλλού κι αν φανεί χρήσιμο.

5.12 Εξωτερικές Βιβλιοθήκες

Οι εξωτερικές βιβλιοθήκες δίνουν γραφικό χαρακτήρα στην εφαρμογή κινητού υπολογισμού μας και χρησιμοποιήθηκαν μέσω του Maven repository. Θα τις συμπεριλάβουμε υπό την μορφή Gradle Groovy DSL.

1. implementation 'com.mxn.soul:flowingdrawer-core:2.1.0' , χρήση του slide μενού και τους δύο Editors.
2. implementation 'com.github.Hitomis:CircleMenu:v1.1.0' , κουμπί για χρήση για το κεντρικό μενού της εφαρμογής.
3. implementation 'com.getbase:floatingactionbutton:1.10.1' , AND και OR κουμπιά στον Editor Wizard.

4. implementation 'com.oguzdev:CircularFloatingActionMenu:1.0.2' , κουμπί στο κάτω μέρος των δύο Editor για αποθήκευση και ανανέωση.
5. implementation 'com.jaredrummler:material-spinner:1.3.1' , custom Spinner στο οποίο μπορούμε να ελέγχουμε το μήκος της dropdown λίστας αντικειμένων.

6

Έλεγχος Συστήματος

6.1 Εισαγωγή

Το τελευταίο κομμάτι που περιλάμβανε η διπλωματική μας εργασία ήταν αυτή της δοκιμής με εξωτερικούς χρήστες. Οι χρήστες κλήθηκαν να δημιουργήσουν πέντε (5) κανόνες με κυμαινόμενη δυσκολία στους δύο editors, Domino και E-Wizard. Με τον όρο κυμαινόμενη δυσκολία αναφερόμαστε τόσο στον αριθμό των έξυπνων συσκευών που περιλαμβάνει ένας κανόνας όσο και στις λογικές συσχετίσεις αυτών. Σε θεωρητικό επίπεδο, όσες περισσότερες έξυπνες συσκευές περιέχει ένας κανόνας αλλά και η σωστή λογική διασύνδεση αυτών καθιστούν τον κανόνα είτε πιο εύκολο είτε πιο δύσκολο στη δημιουργία του, δίνοντας περισσότερη βαρύτητα στη λογική διασύνδεση.

Αρχικά, έπρεπε μέσω της επιλογής «Εκπαίδευση στους Editors» που παρέχει η εφαρμογή μας να γνωρίσουν τους δύο editors. Κάθε editor έχει τον δικό του τρόπο ορισμού λογικών συσχετίσεων και τοποθέτησης έξυπνων συσκευών και γι' αυτό ήταν αναγκαία η περιήγηση στην συγκεκριμένη επιλογή πριν την αλληλεπίδραση με τους Editors. Έκτος από την περιήγηση τους στη επιλογή «Εκπαίδευση στους Editors», τυχόν απορίες λύθηκαν προφορικά από τα δύο μέλη της διπλωματικής εργασίας σε τέτοιο βαθμό ώστε να μην υπέρ-πληροφόρηση και υπέρ-ανάλυση.

Πραγματοποιήθηκαν δύο δοκιμές όπου στην πρώτη βρέθηκαν τεχνικά και σχεδιαστικά σφάλματα τα οποία αφού διορθώθηκαν πραγματοποιήθηκε η δεύτερη και τελική δοκιμή. Τα κριτήρια των δοκιμών ήταν ο χρόνος ολοκλήρωσης του κανόνα και ο αριθμός αγγιγμάτων της οθόνης.

Τέλος, εκτός από τα δύο παραπάνω κριτήρια αξιολόγησης, οι χρήστες κλήθηκαν να απαντήσουν σ' ένα ερωτηματολόγιο τύπου SUS όπου αναλόγως το SUS σκορ γίνεται αντιληπτή η τελική αξιολόγηση για την εφαρμογή μας. Όσο μεγαλύτερο είναι το SUS σκορ τόσο πιο ευχάριστη ήταν η ενασχόληση τους με

την εφαρμογή, όσο μικρότερο είναι το SUS σκορ τόσο πιο δυσάρεστη ήταν η ενασχόληση τους με την εφαρμογή. Το SUS σκορ κυμαίνεται από το 0-100% με βάση το 68%.

6.2 Σενάρια Δοκιμών

Τα σενάρια δοκιμών που δόθηκαν στους χρήστες ήταν συνολικά πέντε (5) με κυμαινόμενη δυσκολία και οριστικοποιήθηκαν έπειτα από συμφωνία μεταξύ του επιβλέποντα καθηγητή και εμάς. Πρόκειται για πέντε (5) κανόνες όπου στους πρώτους απ' αυτούς δεν υπάρχει ιδιαίτερη δυσκολία στην υλοποίηση τους στους δύο Editors αλλά όσο οι χρήστες προχωρούσαν στους επομένους η δυσκολία αυξανόταν. Κάθε κανόνας έχει καθοριστεί έτσι ώστε να είναι αληθοφανείς και αντιπροσωπεύει μια φανταστική πραγματικότητα. Επομένως, εφ' όσον οι χρήστες είχαν στην διάθεση τους αυτές τις έξυπνες συσκευές ένας πιθανός κανόνας θα μπορούσε να είναι ένας απ' αυτούς τους πέντε (5) που θα είχαν σκεφτεί ακόμα και οι ίδιοι να δημιουργήσουν για να ικανοποιήσουν μια ανάγκη ή επιθυμία τους.

6.2.1 ΚΑΝΟΝΑΣ 1

Έξυπνες συσκευές: Κινητό τηλέφωνο.

Κανόνας 1: **AN** το ποσοστό της μπαταρίας του κινητού τηλεφώνου είναι μικρότερο από 15% **TOTE** στείλε ειδοποίηση στην φορητή συσκευή android.

Ο πρώτος κανόνας περιλαμβάνει μία έξυπνη συσκευή η οποία παίρνει μέρος και στο κομμάτι του συμβάντος (event) αλλά και στο κομμάτι της δράσης (action) και πρόκειται για το έξυπνο κινητό τηλέφωνο. Οι χρήστες θα πρέπει να τοποθετήσουν το έξυπνο κινητό τηλέφωνο και στο κομμάτι του συμβάντος και στο κομμάτι της δράσης και για τους δύο Editors και να προχωρήσουν στις κατάλληλες ενέργειες ώστε να οριστικοποιηθεί ο παραπάνω κανόνας.

Στο κομμάτι του συμβάντος θα πρέπει βρίσκοντας την καρτέλα που αναφέρεται στην μπαταρία να ορίσουν το πρώτο μέρος του κανόνα : « **AN** το ποσοστό ... είναι μικρότερο από 15%».

Στο κομμάτι της δράσης θα πρέπει, πάλι, βρίσκοντας την καρτέλα που αναφέρεται στις έξυπνες φορητές συσκευές να ορίσουν το δεύτερο κομμάτι του κανόνα : « **TOTE** στείλε ειδοποίηση ... android». Η ιδιαιτερότητα του συγκεκριμένου κανόνα είναι ότι η ίδια έξυπνη συσκευή παίρνει μέρος και στο κομμάτι του συμβάντος και στο κομμάτι της δράσης.

6.2.2 ΚΑΝΟΝΑΣ 2

Έξυπνες συσκευές: Ρολόι, Τηλεόραση, Φούρνος.

Κανόνας 2: **AN** η ώρα είναι 21:00 **KAI** το πρόγραμμα της τηλεόρασης είναι MasterChef **TOTE** άνοιξε τον φούρνο με θερμοκρασία 20 °C.

Στον δεύτερο κανόνα παίρνουν μέρος τρεις (3) έξυπνες συσκευές, δύο στο κομμάτι του συμβάντος (event), το ρολόι και η τηλεόραση και μία στο κομμάτι της δράσης, ο φούρνος. Στο κομμάτι του συμβάντος έχει προστεθεί και το λογικό AND και πρόκειται για την πρώτη επαφή των χρηστών με κανόνα ο οποίος περιέχει λογική συσχέτιση μεταξύ συσκευών.

Στον Editor Domino, οι χρήστες θα πρέπει να τοποθετήσουν πρώτα το έξυπνο ρολόι, ρυθμισμένο στις 21:00, και έπειτα την έξυπνη τηλεόραση, με επιλογή το MasterChef, το ένα κάτω από το άλλο, κάθετα, ώστε να οριστεί σωστά η λογική συσχέτιση AND. Κατά αυτό τον τρόπο έχει οριστεί το πρώτο μέρος του κανόνα : « **AN** η ώρα ... είναι MasterChef « στον Editor Domino.

Αντίθετα, στον Editor E-Wizard , αρχικά, θα πρέπει να γίνει η προσθήκη του έξυπνου ρολογιού, ρυθμισμένου στις 21:00, και έπειτα θα πρέπει να επιλέξουν σωστά απ' το κατάλληλο κουμπί το λογικό AND και κατόπιν να προσθέσουν την έξυπνη τηλεόραση, με επιλογή το MasterChef.

Το κομμάτι της δράσης και για τους δύο Editors δεν αλλάζει και οι χρήστες θα πρέπει να προσθέσουν τον έξυπνο φούρνο ρυθμισμένο έτσι ώστε να ανοίξει με θερμοκρασία 20 °C. Ο δεύτερος κανόνας αυξάνει κατά ένα βαθμό το επίπεδο δυσκολίας διότι περιλαμβάνει διαφορετικές έξυπνες συσκευές καθώς και λογική συσχέτιση. Τέλος, ρεαλιστικά ο κανόνας θα μπορούσε να ισχύσει και να ήταν χρήσιμος.

6.2.3 ΚΑΝΟΝΑΣ 3

Έξυπνες συσκευές: Κινητό τηλέφωνο.

Κανόνας 3: **AN** τρέχω **H** περπατάω **KAI** τα ακουστικά είναι συνδεδεμένα στο κινητό τηλέφωνο **TOTE** άνοιξε την λίστα αναπαραγωγής μουσικής.

Ο τρίτος κανόνας περιέχει, όπως και ο πρώτος, μόνο μία έξυπνη συσκευή η οποία είναι το κινητό τηλέφωνο. Το επίπεδο δυσκολίας αυξάνεται διότι ο συγκεκριμένος κανόνας περιλαμβάνει δύο λογικές συσχετίσεις οι οποίες είναι διαφορετικές μεταξύ τους, η μία είναι **OR** και η άλλη είναι **AND**.

Επίσης, γίνεται αντιληπτό προς τους χρήστες ότι πέρα απ' ότι μία έξυπνη συσκευή μπορεί να είναι και στο κομμάτι του συμβάντος και στο κομμάτι της δράσης (1^{ος} κανόνας) μπορεί στο κομμάτι του συμβάντος να είναι και παραπάνω από μία φορά και να συνδέεται και με λογικές συσχετίσεις. Κατά αυτό τον τρόπο γίνεται αντιληπτό προς τους χρήστες δυνατότητες που προσφέρουν οι Editors.

6.2.4 ΚΑΝΟΝΑΣ 4

Έξυπνες συσκευές: Θερμόμετρο, Ρολόι, Κινητό τηλέφωνο, Κλιματιστικό

Κανόνας 4: **AN** η θερμοκρασία είναι στους 10 °C **KAI** η ώρα είναι 15:00 **KAI** η τοποθεσία μου είναι σπίτι **TOTE** άνοιξε το κλιματιστικό σε τρόπο λειτουργίας ζεστό με θερμοκρασία 25 °C.

Ο προτελευταίος κανόνας αποτελείται από τέσσερις (4) συνολικά έξυπνες συσκευές. Οι έξυπνες συσκευές που αποτελούν το κομμάτι του συμβάντος είναι το θερμόμετρο, ρολόι και το κινητό τηλέφωνο ενώ στο κομμάτι της δράσης είναι το έξυπνο κλιματιστικό. Ο κανόνας πρόκειται για μια παραλλαγή του δεύτερου κανόνα διότι προστίθεται μία ακόμη ίδια λογική συσχέτιση AND και μία ακόμη έξυπνη συσκευή.

Εφ' όσον, οι χρήστες έχουν κατανοήσει πρωτίστως τις λογικές συσχετίσεις και τον τρόπο με τον οποίο ρυθμίζονται οι έξυπνες συσκευές τότε μπορούν να βρίσκονται σε θέση να δημιουργήσουν κανόνες με αρκετές σε πλήθος έξυπνες συσκευές. Γι' αυτό το λόγο ο συγκεκριμένος κανόνας περιλαμβάνει ίδια λογική συσχέτιση αλλά κάθε κομμάτι του περιλαμβάνει διαφορετική έξυπνη συσκευή.

6.2.5 ΚΑΝΟΝΑΣ 5

Έξυπνες συσκευές: Ψυγείο, Ρολόι, Τηλεόραση, Κινητό τηλέφωνο

Κανόνας 5: *AN τα μπουκάλια της μπύρας είναι λιγότερα από δύο (2) στο ψυγείο KAI η ώρα είναι 19:00 KAI το πρόγραμμα της τηλεόρασης είναι ποδόσφαιρο H το πρόγραμμα της τηλεόρασης είναι Game of Thrones KAI η τοποθεσία μου είναι σπίτι TOTE το ψυγείο να αποστέλλει ηλεκτρονικά παραγγελία.*

Ο τελευταίος περιέχει συνολικά τέσσερις (4) έξυπνες συσκευές, όπως και ο τέταρτος κανόνας, όπου στο κομμάτι του συμβάντος είναι το έξυπνο ψυγείο, ρολόι και η έξυπνη τηλεόραση ενώ στο κομμάτι της δράσης βρίσκεται το έξυπνο ψυγείο.

Πρόκειται για τον πιο δύσκολο σε υλοποίηση κανόνα που κλήθηκαν να δημιουργήσουν οι χρήστες καθώς περιέχει διαφορετικές έξυπνες συσκευές και αρκετές λογικές συσχετίσεις που το καθιστά εύκολο για την ύπαρξη σφαλμάτων από τους χρήστες. Εμπλέκει την ίδια έξυπνη συσκευή και στο κομμάτι του συμβάντος και στο κομμάτι της δράσης (έξυπνο ψυγείο) καθώς και ίδια συσκευή παραπάνω από μία φορές στο κομμάτι του συμβάντος (έξυπνη τηλεόραση). Είναι δυνατότητες τις οποίες έχουν δει οι χρήστες σταδιακά στους προηγούμενους κανόνες αλλά πλέον οι γνώσεις που απέκτησαν κατά τη δημιουργία των προηγούμενων κανόνων συνδυάζονται σ' έναν.

6.3 Δοκιμή Χρηστών

6.3.1 Πρώτη Δοκιμή

Στην πρώτη δοκιμή που πραγματοποιήθηκε συνολικά πήραν μέρος έξι χρήστες και σκοπός ήταν η εύρεση τεχνικών και σχεδιαστικών σφαλμάτων που είχαν παραλειφθεί ή δεν εντοπίστηκαν από τους δημιουργούς της εφαρμογής κατά τη διάρκεια υλοποίησης της. Η δοκιμή πραγματοποιήθηκε σε συνάντηση με τους χρήστες στους οποίους δόθηκε το κινητό τηλέφωνο που ήταν εγκατεστημένη η εφαρμογή. Αρχικά, δόθηκε εξήγηση για το πλαίσιο και τον σκοπό της εφαρμογής σ' όλους τους χρήστες πριν την ανάδραση τους μ' αυτή και έπειτα ξεκίνησε η διαδικασία ανάδρασης με την εφαρμογή με παρουσία ενός εκ των δημιουργών και του χρήστη. Όταν ένας χρήστης πραγματοποιούσε την ανάδραση, οι υπόλοιποι χρήστες βρίσκόντουσαν εκτός αυτού του χώρου. Τα βήματα τα οποία έπρεπε να ακολουθήσουν οι χρήστες ήταν τα εξής:

1. Άνοιγμα της εφαρμογής
2. Επιλογή του Editor E-Wizard
3. Δημιουργία του κανόνα
4. Ορισμός ονόματος κανόνα
5. Αποθήκευση του κανόνα

Η ίδια διαδικασία επαναλήφθηκε για τον Editor Domino.

Παρακάτω παρουσιάζονται σχεδιαστικές και τεχνικές παρατηρήσεις που προήλθαν απ' τους χρήστες :

- Δυσκολία στην ορισμό της δράσης στον πρώτο κανόνα. Συγκεκριμένα οι χρήστες δυσκολεύτηκαν στην εύρεση της ρύθμισης “ στείλε ειδοποίηση στην φορητή συσκευή android ”.
- Δεν υπήρχε δυνατότητα αφαίρεσης-διαγραφής έξυπνης συσκευής κατά τη διάρκεια δημιουργία του κανόνα στο κομμάτι του συμβάντος και της δράσης στον Editor Domino σε περίπτωση που είχε προστεθεί ανεπιθύμητη συσκευή.
- Δεν ήταν ορατό το περίγραμμα των pop-up παραθύρων.

- Πολλές έξυπνες συσκευές για να ρυθμιστούν χρησιμοποιήθηκε για γραφικό στοιχείο του Spinner που παρέχει το Android Studio. Όμως, επειδή ο spinner είχε οριστεί ως αναπτυσσόμενη λίστα (drop-down list) σε συνάρτηση ότι πολλές συσκευές είχαν πολλές δυνατές επιλογές ρύθμισης με αποτέλεσμα να καλύπτεται το navigation bar.

Για όλα τα παραπάνω έγιναν οι κατάλληλες αλλαγές έτσι ώστε να καλυφθούν οι παρατηρήσεις των χρηστών τις οποίες θεωρήσαμε σημαντικές.

6.3.2 Δεύτερη Δοκιμή

Στην δεύτερη δοκιμή που πραγματοποιήθηκε πήραν μέρος, πάλι, έξι (6) συνολικά χρήστες όπου τέσσερις εξ' αυτών ήταν και στην πρώτη δοκιμή και για τους υπόλοιπους δύο (2) ήταν η πρώτη τους επαφή με την εφαρμογή [12]. Πραγματοποιήθηκε μια πρώτη εισαγωγή απ' τους δημιουργούς προς όλους τους χρήστες για το πλαίσιο και τον σκοπό της εφαρμογής όπως και στην πρώτη δοκιμή και έπειτα ξεκίνησε η ανάδραση τους με τον ίδιο τρόπο που πραγματοποιήθηκε και στην πρώτη δοκιμή. Αυτή τη φορά σε κάθε χρήστη για κάθε κανόνα υπολογιζόταν ο χρόνος ολοκλήρωσης του κανόνα και ο αριθμός αγγιγμάτων της οθόνης. Ο χρόνος ολοκλήρωσης του κάθε κανόνα και ο αριθμός των αγγιγμάτων της οθόνης υπολογιζόταν από την εφαρμογή διότι είχε ενσωματωθεί κατάλληλη συνάρτηση στο πρόγραμμα. Επιπλέον, καταγράφηκαν τα λάθη που πραγματοποίησαν οι χρήστες καθ' όλη τη διάρκεια ενασχόλησης τους με τους Editors. Τα λάθη χωρίζονται στις εξής κατηγορίες :

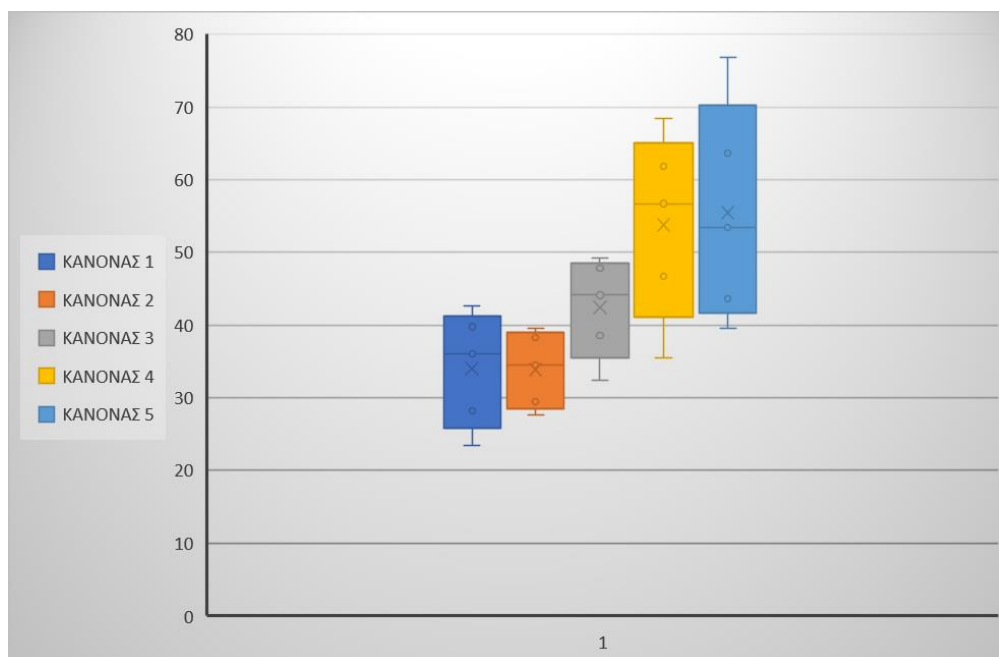
- *Λογικά λάθη* : Λανθασμένη επιλογή λογικού AND ή OR απ' αυτήν που αναφέρει ο κανόνας.
- *Ρύθμισης έξυπνης συσκευής* : Λάθος επιλογή ρύθμισης έξυπνης συσκευής απ' αυτήν που αναφέρει ο κανόνας.
- *Επιλογή λάθους έξυπνης συσκευής* : Επιλογή άλλης έξυπνης συσκευής απ' αυτήν που αναφέρει ο κανόνας.
- *Λάθος τοποθέτηση έξυπνης συσκευής* : Τοποθέτηση έξυπνης συσκευής στο κομμάτι του συμβάντος ενώ ο κανόνας αναφέρει ότι βρίσκεται στο κομμάτι της δράσης και αντιστρόφως.

Παρακάτω παρουσιάζονται οι χρόνοι ολοκλήρωσης, ο αριθμός των αγγιγμάτων της οθόνης και ο αριθμός των λαθών για κάθε χρήστη και για τους δύο Editors :

Editor Domino

	Κανόνας 1	Κανόνας 2	Κανόνας 3	Κανόνας 4	Κανόνας 5
Χρήστης 1, 22 ετών	25,8	28,2	32,4	45	39,6
Χρήστης 2, 22 ετών	40,8	39,6	48	61,8	65,4
Χρήστης 3, 22 ετών	36,6	27,6	37,8	35,4	42
Χρήστης 4, 22 ετών	35,4	33	40,8	61,8	48,6
Χρήστης 5, 25 ετών	42,6	39	47,4	51,6	58,2
Χρήστης 6, 30 ετών	23,4	36	49,2	68,4	76,8
Μέσος Όρος	40,92	40,68	51,12	64,8	66,12

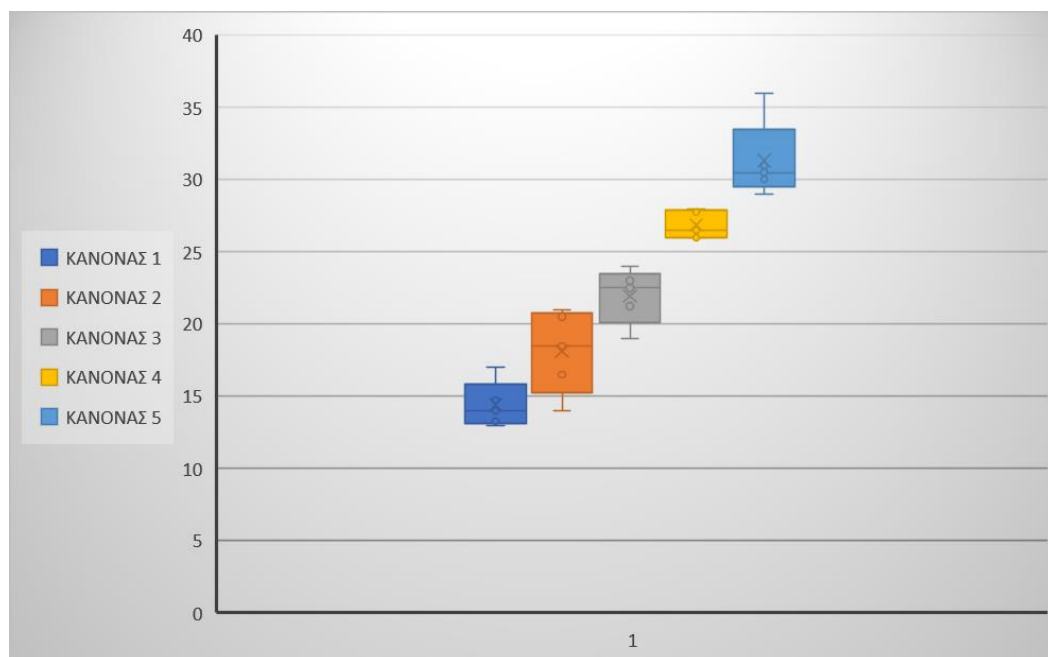
Πίνακας 7. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor Domino.



Boxplot 1. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor Domino.

	Κανόνας 1	Κανόνας 2	Κανόνας 3	Κανόνας 4	Κανόνας 5
Χρήστης 1, 22 ετών	14	19	22	26	31
Χρήστης 2, 22 ετών	13	21	23	28	30
Χρήστης 3, 22 ετών	15	18	24	26	36
Χρήστης 4, 22 ετών	17	21	23	27	29
Χρήστης 5, 25 ετών	13	16	21	26	30
Χρήστης 6, 30 ετών	14	14	19	28	31
Μέσος Όρος	17,2	21,8	26,4	32,2	37,4

Πίνακας 8. Αριθμός αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor Domino.



Boxplot 2. Αριθμός αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor Domino.

Χρήστης 1, 22 ετών	1
Χρήστης 2, 22 ετών	2
Χρήστης 3, 22 ετών	0
Χρήστης 4, 22 ετών	3
Χρήστης 5, 25 ετών	1
Χρήστης 6, 30 ετών	1

Πίνακας 9. Αριθμός λαθών χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor Domino.

Όπως μπορούμε να παρατηρήσουμε από το Boxplot 1 που αφορά την διάρκεια ολοκλήρωσης κανόνα στον Editor Domino οι χρήστες για τον πρώτο και δεύτερο κανόνα χρειάστηκαν σχεδόν τον ίδιο χρόνο για να τους ολοκληρώσουν καθώς η ενδιάμεση τιμή είναι 36 και 34,5 δευτερόλεπτα αντίστοιχα. Για τον κανόνα 1 η άνω εξακοντισμένη τιμή είναι 42,6 δευτερόλεπτα και η κάτω εξακοντισμένη τιμή είναι 23,4 δευτερόλεπτα. Για τον κανόνα 2 η άνω εξακοντισμένη τιμή είναι 39,6 δευτερόλεπτα και η κάτω εξακοντισμένη τιμή είναι 27,6 δευτερόλεπτα. Επομένως, θεωρούμε ότι αντιμετώπισαν σχεδόν τις ίδιες δυσκολίες για τη σύλληψη και την δημιουργία του κανόνα. Γι' αυτούς τους δύο πρώτους κανόνες ο αριθμός αγγιγμάτων της οθόνης (Boxplot 2) διαφέρει καθώς ο δεύτερος κανόνας περιέχει περισσότερες έξυπνες συσκευές ως συνέπεια περισσότερες παραμετροποιήσεις. Αναλυτικά, οι ενδιάμεσες τιμές των δύο κανόνων είναι 14 και 18,5 αγγίγματα οθόνης για κάθε κανόνα αντίστοιχα. Ενώ, η άνω και κάτω εξακοντισμένες τιμές είναι 13 και 17 αγγίγματα οθόνης και 14 και 21 αγγίγματα οθόνης για κάθε κανόνα αντίστοιχα. Πιστεύουμε ότι ο πρώτος κανόνας δυσκόλεψε περισσότερο τους χρήστες διότι η ίδια συσκευή λαμβάνει μέρος και στο κομμάτι του συμβάντος και στο κομμάτι της δράσης, γεγονός που οι χρήστες χρειάστηκαν μερικά δευτερόλεπτα παραπάνω για να το συνειδητοποιήσουν και να καταλάβουν ότι μπορούν να το εφαρμόσουν. Σε αντίθεση με τον πρώτο κανόνα, στον δεύτερο συμπεριλαμβάνοντας τις

σωστές διαφορετικές έξυπνες συσκευές και κάνοντας τις κατάλληλες παραμετροποιήσεις κατάφεραν να ολοκληρώσουν τον κανόνα.

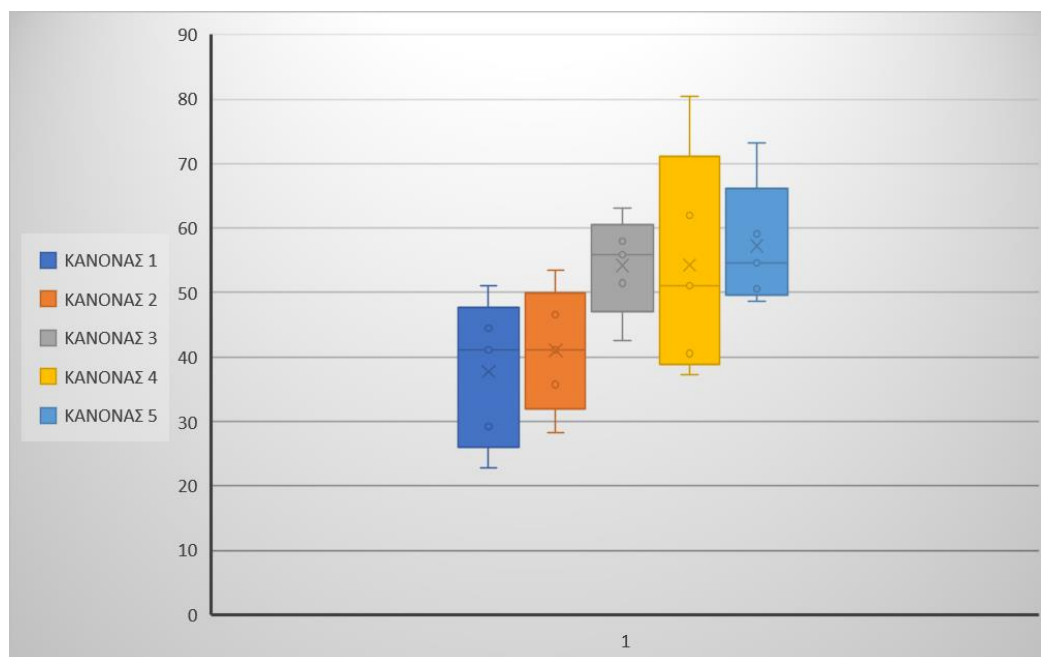
Επιπροσθέτως, όσον αφορά τον τέταρτο και τον πέμπτο κανόνα μπορούμε να παρατηρήσουμε ότι η διάρκεια ολοκλήρωσης αυτών βρίσκεται πολύ κοντά. Συγκεκριμένα, οι ενδιάμεσες τιμές ολοκλήρωσης των συγκεκριμένων κανόνων είναι 56,7 και 53,4. Η άνω εξακοντισμένη τιμή για τον κανόνα 4 ήταν 68,4 δευτερόλεπτα ενώ για τον κανόνα 5 ήταν 76,8 δευτερόλεπτα. Επίσης, για τον κανόνα 4 η κάτω εξακοντισμένη τιμή ήταν 35,4 δευτερόλεπτα ενώ για τον κανόνα 5 ήταν 39,6 δευτερόλεπτα. Ωστόσο, ήταν σαφές και για τους δύο κανόνες ποιες έξυπνες συσκευές έπρεπε να χρησιμοποιήσουν, πως να τις ρυθμίσουν και πως να εφαρμόσουν τις λογικές συσχετίσεις καθώς είχαν εξοικειωθεί ως ένα βαθμό με τον Editor. Παρ' όλα αυτά, ο αριθμός των αγγιγμάτων της οθόνης (Boxplot 2) γι' αυτούς τους δύο κανόνες έχει μια άνοδο, διότι ο πέμπτος κανόνας και πιο δύσκολος κανόνας περιέχει έξυπνες συσκευές οι οποίες για την ρύθμιση τους χρειάζονται αρκετά αγγίγματα στην οθόνη. Πιο λεπτομερώς, μπορούμε να διαπιστώσουμε ότι η ενδιάμεση τιμή για τον αριθμό αγγιγμάτων οθόνης για τον κανόνα 4 είναι 26,5 ενώ για τον κανόνα 30,5 αγγίγματα οθόνης. Ο κανόνας 4 έχει άνω εξακοντισμένη τιμή 28 αγγίγματα οθόνης. Αντίθετα, οι τιμές αυτές διαφέρουν στον πέμπτο κανόνα για τους λόγους που προαναφέρθηκαν και βλέπουμε ότι η κάτω εξακοντισμένη τιμή είναι 29, πλησιάζει την αντίστοιχη τιμή του κανόνα 4, και η άνω εξακοντισμένη τιμή είναι 36. Απέχει αρκετά από την αντίστοιχη τιμή του κανόνα 5.

Τέλος, οι χρήστες χρειάστηκαν κατά μέσο όρο περίπου 44,1 δευτερόλεπτα και 22,55 αγγίγματα οθόνης για να ολοκληρώσουν τον 3^ο κανόνα. Η κάτω εξακοντισμένη τιμή διάρκειας ολοκλήρωσης του 3^{ου} κανόνα είναι 32,4 δευτερόλεπτα ενώ η άνω 49,2. Αντίστοιχα, η άνω εξακοντισμένη τιμή αγγιγμάτων οθόνης είναι 24 και η κάτω 19.

Editor E – Wizard

	Κανόνας 1	Κανόνας 2	Κανόνας 3	Κανόνας 4	Κανόνας 5
Χρήστης 1, 22 ετών	22,8	28,2	42,6	39,6	51
Χρήστης 2, 22 ετών	37,8	36,6	58,2	43,2	59,4
Χρήστης 3, 22 ετών	44,4	35,4	57	37,2	58,2
Χρήστης 4, 22 ετών	44,4	53,4	50,4	58,8	50,4
Χρήστης 5, 25 ετών	51	46,8	63	63	48,6
Χρήστης 6, 30 ετών	26,4	45,6	54,6	80,4	73,2
Μέσος Όρος	45,36	49,2	65,16	64,44	68,16

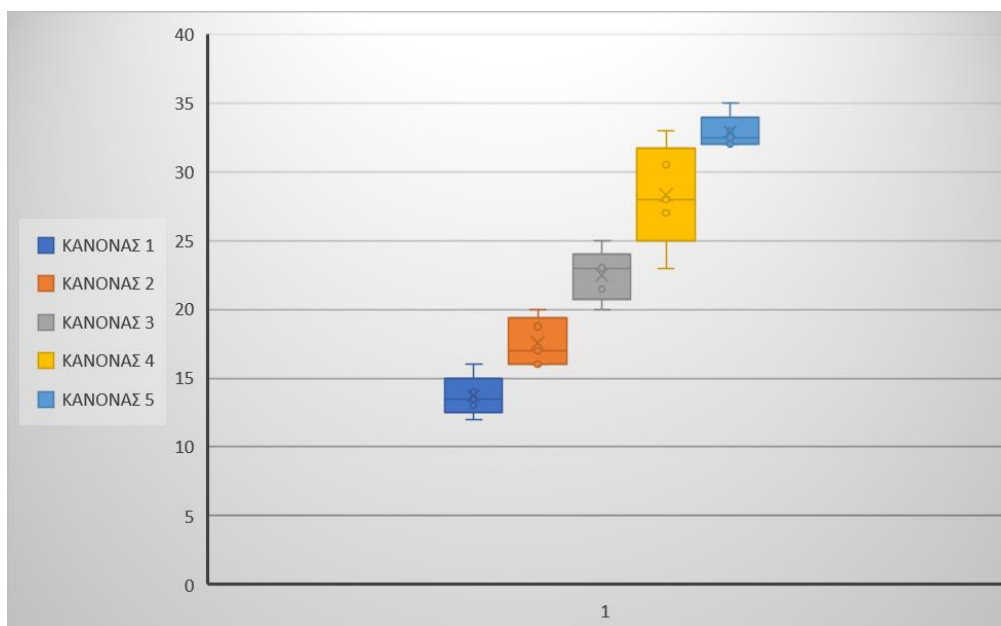
Πίνακας 10. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor E - Wizard.



Boxplot 3. Χρόνος (δευτερόλεπτα) χρηστών για την ολοκλήρωση κανόνων στον Editor E - Wizard.

	Κανόνας 1	Κανόνας 2	Κανόνας 3	Κανόνας 4	Κανόνας 5
Χρήστης 1, 22 ετών	16	20	23	27	32
Χρήστης 2, 22 ετών	14	16	25	29	32
Χρήστης 3, 22 ετών	14	19	23	31	33
Χρήστης 4, 22 ετών	13	16	23	33	35
Χρήστης 5, 25 ετών	12	16	20	27	33
Χρήστης 6, 30 ετών	13	18	21	23	32
Μέσος Όρος	16,4	21	27	34	39,4

Πίνακας 11. Αριθμός αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor E - Wizard.



Boxplot 4. Αριθμού αγγιγμάτων οθόνης χρηστών κατά τη διαδικασία δημιουργίας κανόνα στον Editor E-Wizard.

Χρήστης 1, 22 ετών	0
Χρήστης 2, 22 ετών	1
Χρήστης 3, 22 ετών	1
Χρήστης 4, 22 ετών	1
Χρήστης 5, 25 ετών	1
Χρήστης 6, 30 ετών	1

Πίνακας 12. Αριθμός Λαθών χρηστών κατά τη διαδικασία δημιουργίας κανόνων στον Editor E - Wizard

Σχετικά με την διάρκεια ολοκλήρωσης κανόνα και τον αριθμό αγγιγμάτων στον Editor E-Wizard παρατηρείται ότι για όλους τους κανόνες οι χρήστες χρειάστηκαν περισσότερο χρόνο (Boxplot 3). Ένας λόγος είναι ότι οι δοκιμές πραγματοποιήθηκαν πρώτα στον Editor E-Wizard και έπειτα στον Editor Domino. Ως συμπέρασμα, μπορεί να θεωρηθεί ότι επειδή ήταν η πρώτη τους επαφή με περιβάλλον δημιουργίας κανόνα χρειάστηκαν συνολικά περισσότερο χρόνο προσαρμογής.

Αφού ολοκλήρωναν τους κανόνες στον Editor E-Wizard και προχωρούσαν στην διαδικασία ανάπτυξης κανόνων στον Editor Domino υπήρχε μια μικρή γνώση για αυτό που επρόκειτο να συναντήσουν και το μόνο πρωτόγνωρο γι' αυτούς ήταν η απεικόνιση των έξυπνων συσκευών καθώς και ο διαφορετικός τρόπος ορισμού του λογικού AND και OR.

Παρατηρείται ότι η διάρκεια ολοκλήρωσης των κανόνων αυξάνεται προχωρώντας από έναν κανόνα στον άλλο αλλά με το γεγονός ότι οι χρήστες χρειάστηκαν λίγα δευτερόλεπτα παραπάνω για να ολοκληρώσουν τον τρίτο κανόνα σε σχέση με τον 4^ο καθώς η ενδιάμεση τιμή του 3^{ου} κανόνα ξεπερνά αυτή του 4^{ου}.

Ο τρίτος κανόνας αποδεικνύεται ότι οι χρήστες χρειάστηκαν περισσότερο χρόνο για να τον ολοκληρώσουν σε σχέση με τον τέταρτο κανόνα που είναι πιο δύσκολος. Παρ' όλα αυτά ο αριθμός των

αγγιγμάτων της οθόνης είναι μικρότερος σε σχέση με τον τέταρτο κανόνα, πράγμα που φανερώνει ότι οι χρήστες χρειάστηκαν περισσότερο χρόνο να τον επεξεργαστούν.

Επίσης μέσω του Boxplot 4 διαπιστώνεται ότι ο αριθμός αγγιγμάτων οθόνης αυξάνεται σε κάθε κανόνα.

Τέλος, είναι ευχάριστο και αξιοσημείωτο ότι τα λάθη των χρηστών που καταγράφηκαν ήταν μόνο λάθη ρύθμισης έξυπνης συσκευής. Λάθη λογικά, επιλογής και τοποθέτησης έξυπνης συσκευής δεν πραγματοποιήθηκαν συμπεραίνοντας ότι οι χρήστες κατανόησαν την ερμηνεία του λογικού AND και OR, η απεικόνιση των έξυπνων συσκευών ήταν ξεκάθαρη και η διαφορά συμβάντος και δράσης αντιληπτή.

6.4 Αξιολόγηση

Αφού ολοκληρώθηκε η διαδικασία αλληλεπίδρασης των χρηστών με την εφαρμογή τους δόθηκε ένα ερωτηματολόγιο τύπου SUS. Το ερωτηματολόγιο SUS πρόκειται για ένα κριτήριο αξιολόγησης το οποίο θα μας βοηθήσει να κατανοήσουμε τον βαθμό στον οποίο η εφαρμογή μας είναι εύχρηστη αλλά και γενικότερα την συνολική εμπειρία των χρηστών με την εφαρμογή. Πρόκειται για ένα ερωτηματολόγιο το οποίο περιέχει δέκα (10) ερωτήσεις με πέντε (5) πιθανές απαντήσεις με κλίμακα από 1 έως 5 για κάθε ερώτηση (Παράρτημα I – System Usability Scale).

Στον Πίνακα 13 θα παρουσιαστούν:

- Οι απαντήσεις των χρηστών σε κάθε ερώτηση
- Το SUS σκορ του κάθε χρήστη

Στον Πίνακα 14 θα παρουσιαστούν:

- Ο μέσος όρος κάθε απάντησης του ερωτηματολογίου και των SUS σκορ.
- Η τυπική απόκλιση κάθε απάντησης του ερωτηματολογίου και των SUS σκορ.

Στον Πίνακα 15 θα παρουσιαστεί:

- Ο μικρότερος μέσος όρος συνολικά
- Ο μεγαλύτερος μέσος όρος συνολικά
- Το μεγαλύτερο SUS σκορ χρήστη

Σύμφωνα με τον μέσο όρο όλων των SUS σκορ θα κριθεί ο βαθμός στον οποίο οι χρήστες βρήκαν την εφαρμογή εύχρηστη και ικανοποιήθηκαν από όλη την εμπειρία. Εφ' όσον ο μέσος όρος είναι άνω του 68 θεωρείται ότι η εφαρμογή κινητού υπολογισμού μας καλύπτει τις ανάγκες, τους λόγους που δημιουργήθηκε ως ένα βαθμό διαφορετικά χρίζει βελτίωσης.

#Ερώτησης Χρήστες	E.1	E.2	E.3	E.4	E.5	E.6	E.7	E.8	E.9	E.10	SUS σκορ
Χρήστης 1, 22 Ετών	5	1	4	1	5	1	4	1	5	1	95
Χρήστης 1, 22 Ετών	4	2	5	2	4	2	4	2	4	3	75
Χρήστης 1, 22 Ετών	4	1	4	2	4	1	5	1	4	2	85
Χρήστης 1, 22 Ετών	4	2	4	3	5	1	4	2	3	3	72,5
Χρήστης 1, 22 Ετών	5	1	5	1	5	1	4	1	5	3	92,5
Χρήστης 1, 22 Ετών	4	2	5	1	3	2	5	1	5	2	85

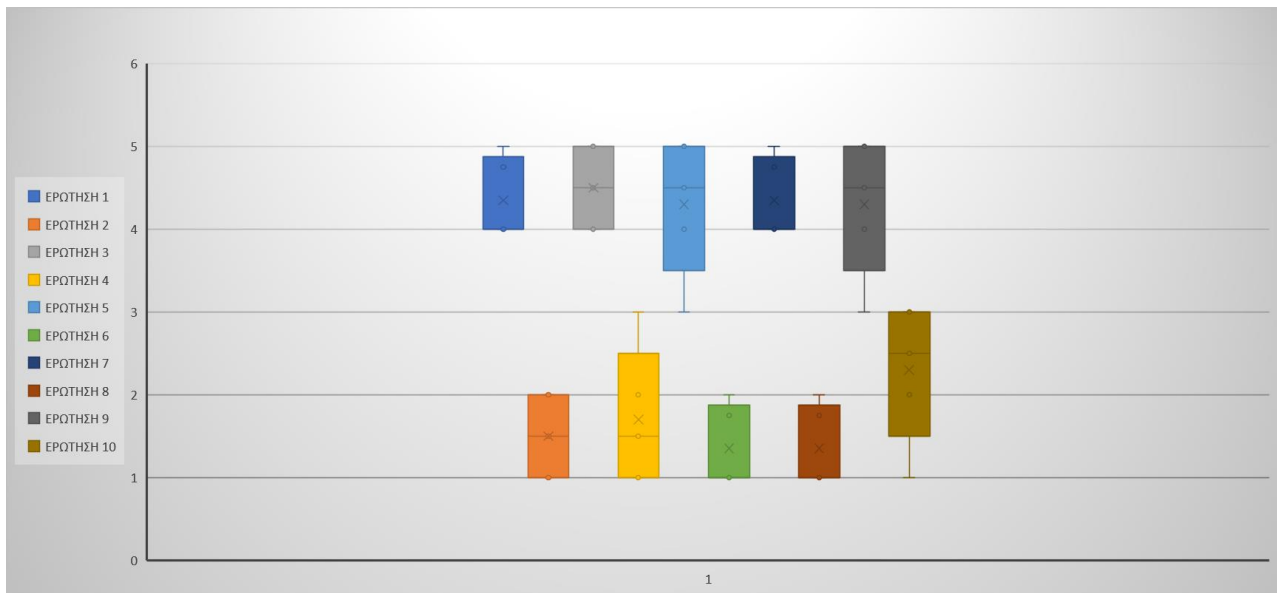
Πίνακας 13. Απαντήσεις χρηστών στο ερωτηματολόγιο SUS και τα αντίστοιχα SUS σκορ.

Μέσος Όρος	4,333	1,5	4,5	1,6667	4,333	1,333	4,333	1,333	4,333	2,333	84,168
Τυπική Απόκλιση	0,516	0,547	0,547	0,816	0,816	0,516	0,516	0,516	0,816	0,816	9,03

Πίνακας 14. Μέσος όρος και τυπική απόκλιση κάθε ερώτησης του SUS ερωτηματολογίου και των SUS σκορ.

Μικρότερος Μέσος Όρος	1,5
Μεγαλύτερος Μέσος Όρος	4,333
Μεγαλύτερο SUS σκορ	95

Πίνακας 15. Μικρότερος, μεγαλύτερος μέσος όρος SUS ερωτηματολογίου και μεγαλύτερο SUS σκορ χρήστη.



Boxplot 5. Απαντήσεις των χρηστών στο SUS ερωτηματολόγιο.

Πρώτα απ' όλα, παρατηρείται από το γράφημα 5 ότι οι ερωτήσεις που έχουν θετικό χαρακτήρα ως προς το εργαλείο κυμαίνονται μεταξύ του 3 και του 5 και αυτές που έχουν αρνητικό χαρακτήρα κυμαίνονται μεταξύ του 1 και του 3.

Συμπεραίνοντας παρατηρείται από το Boxplot 5 ότι:

- ✓ Ένα ικανοποιητικό ποσοστό χρηστών θα χρησιμοποιούσε συχνά την συγκεκριμένη εφαρμογή καθώς η απάντηση στην ερώτηση 1 βρίσκεται στο διάστημα 4 - 5 με ενδιάμεση τιμή το 4.
- ✓ Οι ερωτήσεις που σχετίζονται με την ευκολία μάθησης είναι οι ερωτήσεις 4, 7 και 10. Οι απαντήσεις της ερώτησης 4 παρατηρούμε ότι βρίσκονται στο διάστημα 1 έως 3 με το μεγαλύτερο ποσοστό χρηστών να απάντησε με 1 ή 2, η ενδιάμεση τιμή είναι 1,5 και η άνω εξακοντισμένη τιμή το 3. Καταλήγουμε ότι οι πλειοψηφία των χρηστών θα μπορούσε να διαχειριστεί την εφαρμογή χωρίς την παρέμβαση τεχνικού. Επιπλέον, οι απαντήσεις της ερώτησης 7 εμπεριέχονται στο διάστημα 4 έως 5, με ενδιάμεση τιμή το 4 και άνω εξακοντισμένη τιμή το 5. Συνεπώς, οι χρήστες συμφωνούν ότι η εκμάθηση του εργαλείου μπορεί να πραγματοποιηθεί γοργά. Επιπροσθέτως, οι απαντήσεις της ερώτησης 10 παρατηρείται ότι βρίσκονται στο διάστημα 1 έως 3, με κάτω εξακοντισμένη τιμή το 1 και ενδιάμεση τιμή το 2,5. Άρα, οι χρήστες κατέληξαν στο συμπέρασμα ότι οι δεν ήταν απαραίτητο να έχουν ιδιαίτερα πολλή γνώση πριν χρησιμοποιήσουν την εφαρμογή.
- ✓ Οι απαντήσεις της ερώτησης 3 παρατηρούμε ότι περιορίζονται στο 100% στις επιλογές 4 και 5, με ενδιάμεση τιμή το 4,5. Δηλαδή, όλοι οι χρήστες θεώρησαν σε ικανοποιητικό βαθμό την εφαρμογή μας εύκολη στη χρήση. Επίσης, οι απαντήσεις της ερώτησης 9 περιέχονται στο διάστημα 3 έως 5, με κάτω εξακοντισμένη τιμή το 3 και ενδιάμεση τιμή το 4,5. Μπορούμε να υποστηρίξουμε ότι οι περισσότεροι χρήστες ένοιωθαν σίγουροι για τον τρόπο με τον οποίο χειρίζονται την εφαρμογή.
- ✓ Για την ερώτηση 5, οι απαντήσεις βρίσκονται στο διάστημα 3 έως 5, με κάτω εξακοντισμένη τιμή το 3 και ενδιάμεση τιμή το 4,5. Είμαστε στη θέση να πούμε με ασφάλεια ότι οι λειτουργίες του συστήματος είναι σε καλό βαθμό προσαρμοσμένες στο εργαλείο μας.

- ✓ Οι απαντήσεις στις ερωτήσεις 2,6 και 8 διαπιστώνουμε ότι βρίσκονται στο διάστημα 1 έως 2. Μπορούμε να θεωρήσουμε ότι η εφαρμογή μας δεν κρίνεται περίπλοκη και πολύπλοκη άνευ λόγου.

Συνοψίζοντας, μπορεί να υποστηριχθεί ότι η εκμάθηση του εργαλείου που υλοποιήσαμε κρίνεται εύκολη και αποτελεί ένα χαρακτηριστικό της συγκεκριμένης σχεδίασης φιλικής διεπαφής βάσει των παραπάνω απαντήσεων στις ερωτήσεις, κυρίως των απαντήσεων στις ερωτήσεις 4,7 και 10 καθώς και μέσω του συνολικού μέσο όρου των SUS scores ο οποίος είναι 84,168.

6.5 Ικανοποίηση Απαιτήσεων

6.5.1 Ικανοποίηση λειτουργικών απαιτήσεων

ΛΑ1.	Καλύφθηκε μέσω στους δημιουργίας Editor Domino (βλέπε Ενότητα «4.3.2 Editor Domino»).
ΛΑ2.	Καλύφθηκε μέσω στους δημιουργίας Editor E-Wizard (βλέπε Ενότητα «4.3.3 Editor E - Wizard»).
ΛΑ3.	Υλοποιήθηκε κατάλληλη επιλογή για την εκπαίδευση στα περιβάλλοντα δημιουργίας κανόνα (βλέπε Ενότητα «4.3.6 Εκπαίδευση στους Editors»).
ΛΑ4.	Σχεδιάστηκε και υλοποιήθηκε και για τα δύο περιβάλλοντα δημιουργίας κανόνων, Editor Domino και E-Wizard (βλέπε Ενότητες «4.3.2 Editor Domino» και «4.3.3 Editor E - Wizard»).
ΛΑ5.	Σχεδιάστηκε και υλοποιήθηκε για τα δύο περιβάλλοντα δημιουργίας κανόνων, Editor Domino και E-Wizard (βλέπε Ενότητες «4.3.2.1 Λογικό OR», «4.3.2.2 Λογικό AND» και «4.3.3.1 Λογικό OR και AND»).
ΛΑ6.	Καλύφθηκε και για τα δύο περιβάλλοντα δημιουργίας κανόνων, Editor Domino και Wizard (βλέπε Ενότητα «4.4.8 Αναγραφή Ενεργειών Έξυπνων Συσκευών»).
ΛΑ7 – ΛΑ8.	Υλοποιήθηκε αντίστοιχη λειτουργία και για τα δύο περιβάλλοντα δημιουργίας κανόνων, Editor Domino και E-Wizard (βλέπε Ενότητες «4.3.2.6 Ανανέωση και Διαγραφή Έξυπνης Συσκευής» και «4.3.3.3 Ανανέωση και Διαγραφή Έξυπνης Συσκευής»).
ΛΑ9 – ΛΑ10 – ΛΑ11 – ΛΑ12.	Υποστήριξη επανεκκίνησης περιβάλλοντος δημιουργίας κανόνων με τη χρήση έξυπνων συσκευών υπό την μορφή domino. Καλύφθηκε αντίστοιχη λειτουργία και για τα δύο περιβάλλοντα δημιουργίας κανόνων, Editor Domino και E-Wizard (βλέπε Ενότητα «4.3.2.5 Αποθήκευση και Επανεκκίνηση»).
ΛΑ13.	Υλοποιήθηκε αντίστοιχη λειτουργία και για τα δύο περιβάλλοντα δημιουργίας κανόνων, Editor Domino και E-Wizard (βλέπε Ενότητα «4.3.5 Αποθηκευμένοι Κανόνες»).
ΛΑ14.	Υλοποιήθηκε αντίστοιχη λειτουργία και για τα δύο περιβάλλοντα δημιουργίας κανόνων, Editor Domino και E-Wizard (βλέπε Ενότητα «4.3.5 Αποθηκευμένοι Κανόνες»).
ΛΑ15.	Σχεδιάστηκε και υλοποιήθηκε αντίστοιχη λειτουργία (βλέπε Ενότητα «4.4.9 Αναγραφή Αποθηκευμένου Κανόνα»).
ΛΑ16.	Καλύφθηκε αντίστοιχη λειτουργία (βλέπε Ενότητα «4.4.10 Animation»).
ΛΑ17 –	Υλοποιήθηκε αντίστοιχη λειτουργία (βλέπε Ενότητα «4.3.4 Κάτοψη

ΛΑ18.	(Floorspan)»).
ΛΑ19.	Καλύφθηκε αντίστοιχη λειτουργία (βλέπε Ενότητα «4.3.4.2 Τοποθέτηση Έξυπνων Συσκευών»).
ΛΑ20.	Καλύφθηκε αντίστοιχη λειτουργία (βλέπε Ενότητα «4.3.4.3 Αποθήκευση Επεξεργασμένης Εικόνας»).
ΛΑ21.	Σχεδιάστηκε και υλοποιήθηκε αντίστοιχη επιλογή στην εφαρμογή (βλέπε Ενότητα «4.3.7 Πληροφορίες Έξυπνων Συσκευών»).

Πίνακας 16. Ικανοποίηση λειτουργικών απαιτήσεων εφαρμογής.

6.5.2 Ικανοποίηση μη λειτουργικών απαιτήσεων

ΜΛΑ1 –	Καλύφθηκε καθώς για την υλοποίηση του προγράμματος χρησιμοποιήθηκε η
ΜΛΑ2.	γλώσσα προγραμματισμού Java και το περιβάλλον ανάπτυξης του προγράμματος ήταν το Android Studio (βλέπε Ενότητα «4.3.1 Εισαγωγή»).
ΜΛΑ3.	Καλύφθηκε καθώς έγινε έρευνα και ενσωματώσει βασικών σχεδιαστικών στοιχείων και κανόνων που αφορούν την φιλική διεπαφή (βλέπε Ενότητα «4.4 Σχεδιασμός και Λειτουργίες Διεπαφής»).
ΜΛΑ4.	Χρησιμοποιήθηκαν εργαλεία έτσι ώστε να επιτυγχάνεται σε όσο το δυνατόν καλύτερο βαθμό γίνεται (βλέπε Ενότητα «4.3.1 Εισαγωγή»).
ΜΛΑ5.	Οι χρήστες για την ολοκλήρωση της δημιουργίας κανόνα θα πρέπει να έχουν προσθέσει έξυπνες συσκευές και στο κομμάτι του συμβάντος (event) και στο κομμάτι της δράσης (action) (βλέπε Ενότητα «4.3.2.4 Διαχείριση Κανόνα»).
ΜΛΑ6.	Υλοποιήθηκε αντίστοιχη λειτουργία μεταφόρτωσης εικόνας κάτοψης (βλέπε Ενότητα «4.3.4 Κάτοψη (Floorspan)»).
ΜΛΑ7.	Καλύφθηκε αντίστοιχη λειτουργία που αναλόγως το περιβάλλον δημιουργίας κανόνα υιοθετεί αντίστοιχα τον τρόπο λογικής σύνδεσης έξυπνων συσκευών (βλέπε Ενότητες «4.3.2.1 Λογικό OR», «4.3.2.2 Λογικό AND» και «4.3.3.1 Λογικό OR και AND»).
ΜΛΑ8.	Σχεδιάστηκε αντίστοιχος τρόπος απεικόνισης των έξυπνων συσκευών στην εφαρμογή (βλέπε Ενότητα «4.4.3 Χρήση Dominos ως metaphor για την απεικόνιση των έξυπνων συσκευών στον Editor Domino»).
ΜΛΑ9.	Σχεδιάστηκε αντίστοιχος τρόπος απεικόνισης των έξυπνων συσκευών στην εφαρμογή (βλέπε Ενότητα «4.4.4 Χρήση Icons ως metaphor για την απεικόνιση των έξυπνων συσκευών στον Editor E – Wizard»).
ΜΛΑ10.	Καλύφθηκε μέσω της προσθήκης emoji στην απεικόνιση της δράσης σε μορφή απλού κειμένου (βλέπε Ενότητα «4.4.12 Emoji»).
ΜΛΑ11.	Η εφαρμογή θα πρέπει να μην επιτρέπει την δημοσιοποίηση των προσωπικών δεδομένων του χρήστη (βλέπε Ενότητα «4.2.2 Ασφάλεια Συστήματος»).

Πίνακας 17. Ικανοποίηση μη λειτουργικών απαιτήσεων.

7

Αδυναμίες και Βελτίωση Συστήματος

7.1 Αδυναμίες

7.1.1 Εισαγωγή

Κατά τη διάρκεια ανάπτυξης της εφαρμογής κινητού υπολογισμού που κληθήκαμε να υλοποιήσουμε συναντήσαμε προκλήσεις και δυσκολίες. Μερικές απ' αυτές καταφέραμε να τις ξεπεράσουμε και κάποιες να τις βελτιώσουμε στον καλύτερο δυνατό βαθμό. Με την έννοια αδυναμίες αναφερόμαστε στις υπάρχοντες λειτουργίες του συστήματος οι οποίες κρίνονται ως αδύνατα σημεία για την εφαρμογή μας.

7.1.2 Στατικό περιβάλλον τοποθέτησης έξυπνων συσκευών στο μέρος των συμβάντων στον Editor Domino

Όταν οι χρήστες έχουν επιλέξει να δημιουργήσουν κανόνα με τη χρήση του Editor Domino υπάρχει περιορισμός στον αριθμό των έξυπνων συσκευών που μπορούν να προσθέσουν στο κομμάτι του συμβάντος. Το πλέγμα στο οποίο βασίζονται οι χρήστες για να τοποθετήσουν τις έξυπνες συσκευές έχει μέγεθος 5x5, που σημαίνει ότι συνολικά μπορούν να τοποθετήσουν 25 έξυπνες συσκευές συνδεδεμένες μεταξύ τους με λογικές συσχετίσεις. Επιπλέον, εφ' όσον σε κάθε γραμμή και στήλη του πλέγματος τοποθετούνται έως πέντε έξυπνες συσκευές υπάρχει η περίπτωση της επιθυμίας απ' τους χρήστες να επιθυμούν να τοποθετήσουν έξι (6) ή παραπάνω έξυπνες συσκευές σε μία γραμμή ή/και στήλη για να δημιουργήσουν τον κανόνα που επιθυμούν. Ως αποτέλεσμα, να μην μπορεί η παραπάνω επιθυμία των χρηστών να ικανοποιηθεί από τον Editor Domino. Σίγουρα, ο συγκεκριμένος αριθμός καλύπτει σημαντικά

ένα μεγάλο ποσοστό κανόνων που μπορούν να ικανοποιηθούν. Παρ' όλα αυτά περιορίζει ως ένα βαθμό την δυνατότητα του Editor Domino αλλά και τις δυνατότητες που παρέχονται στους χρήστες. Στον Editor E-Wizard δεν υπάρχει περιορισμός στον αριθμό των έξυπνων συσκευών που μπορούν να χρησιμοποιηθούν είτε στο κομμάτι του συμβάντος είτε στο κομμάτι της δράσης. Οπότε, αν οι χρήστες συλληφθούν έναν πολυπληθή κανόνα, που είτε ο αριθμός των έξυπνων συσκευών ξεπερνάει τους 25 είτε δεν ικανοποιούνται απ' τις λογικές συσχετίσεις, προτείνεται να χρησιμοποιήσουν τον Editor E-Wizard.

7.1.3 Ορισμός μίας δράσης στον Editor Domino και διαγραφής της ήδη ορισμένης δράσης

Στον Editor E-Wizard οι χρήστες μπορούν να συμπεριλάβουν περισσότερες από μία έξυπνες συσκευές στο κομμάτι της δράσης ενώ στον Editor Domino υπάρχει ο περιορισμός του ορισμού μίας έξυπνης συσκευής ως δράση. Ο λόγος είναι ότι όπως αναφέρθηκε στην Ενότητα «Editor Domino» ο συγκεκριμένος Editor βασίστηκε στο ομώνυμο παιχνίδι και σ' έναν συγκεκριμένο τρόπο του παιχνιδιού αυτού, ο οποίος είναι η τοποθέτηση των Domino σε όρθια θέση και η ρίψη του πρώτου ή πρώτων με σκοπό την πτώση του τελευταίου. Έτσι, κύριος στόχος είναι η πτώση της τελευταίας έξυπνης συσκευής domino που ορίστηκε ως δράση. Ακόμη, δεν δίνεται η δυνατότητα διαγραφής της έξυπνης συσκευής που έχει οριστεί ως δράση στον Editor Domino σε περίπτωση που ο χρήστης επιθυμεί την αλλαγή της δράσης ενώ βρίσκεται στη διαδικασία ανάπτυξης του κανόνα.

7.1.4 Ανανέωση, επεξεργασία και μετονομασία αποθηκευμένου κανόνα

Γνωρίζουμε ότι δύο βασικές λειτουργίες, γενικότερα, είναι η διαγραφή (delete) και η ανανέωση (update). Η εφαρμογή μας δίνει την δυνατότητα της διαγραφής και της ανανέωσης για τις έξυπνες συσκευές κατά τη διάρκεια ανάπτυξης των κανόνων στους δύο Editors καθώς επίσης και την επιλογή διαγραφής αποθηκευμένου κανόνα. Όμως, δεν υπάρχει η επιλογή της ανανέωσης/επεξεργασίας ενός ήδη αποθηκευμένου κανόνα. Σε περίπτωση που κάποιος κανόνας αποθηκευτεί έπειτα προσφέρεται μόνο η επιλογή της διαγραφής του κανόνα. Έτσι, αν κάποιος χρήστης επιθυμεί την επεξεργασία ενός αποθηκευμένου κανόνα πρέπει, προς το παρόν, να τον διαγράψει και να τον επαναλάβει από την αρχή ή να τον διατηρήσει και να τον ξανά επαναλάβει από την αρχή. Τέλος, όπως έγινε γνωστό στην Ενότητα «4.3.2.5 Αποθήκευση και Επανεκκίνηση» η τελευταία ενέργεια που πρέπει να πραγματοποιήσουν οι χρήστες για να αποθηκεύσουν τον κανόνα είναι να ορίσουν το όνομα του. Ωστόσο, δεν δίνεται η δυνατότητα αλλαγής του ονόματος ενός αποθηκευμένου κανόνα σε περίπτωση που κάποιος χρήστης επιθυμεί την μετονομασία του.

7.2 Βελτίωση

7.2.1 Editor Domino

Ο Editor Domino καλύπτει σ' έναν ικανοποιητικό βαθμό την βασική ιδέα για την οποία δημιουργήθηκε. Πρόκειται για έναν Editor ο οποίος εξασφαλίζει την διάκριση των λογικών συσχετίσεων, τον διαχωρισμό

συμβάντος και δράσης και η απεικόνιση των έξυπνων συσκευών είναι ευδιάκριτη. Όμως, χρήζει βελτίωσης σε ορισμένα κρίσιμα σημεία τα οποία θα κάνουν την εμπειρία του χρήστη πιο εύκολη και εύχρηστη.

Τα σημεία που χρήζουν βελτίωση είναι τα εξής:

- Ο αριθμός των έξυπνων συσκευών στο κομμάτι του συμβάντος θα πρέπει να είναι απροσδιόριστος. Όπως αναφέρθηκε στο ίδιο Κεφάλαιο υπάρχει ο περιορισμός της τοποθέτησης έως πέντε έξυπνων συσκευών σε κάθε στήλη και γραμμή του πλέγματος.
- Προσθήκη λειτουργίας διαγραφής της έξυπνης συσκευής στο κομμάτι της δράσης.

Διαγραφή οποιασδήποτε έξυπνης συσκευής στο κομμάτι του συμβάντος κατά τη διάρκεια ανάπτυξης του κανόνα. Όπως αναφέρθηκε στην Ενότητα «4.3.2.6 Ανανέωση και Διαγραφή Έξυπνης Συσκευής» παρέχεται η δυνατότητα διαγραφής της τελευταίας domino έξυπνης συσκευής που προστέθηκε στο πλέγμα των συμβάντων. Θα ήταν προτιμότερο να εξασφαλίζεται η δυνατότητα διαγραφής οποιασδήποτε έξυπνης συσκευής domino σε οποιοδήποτε σημείο του πλέγματος οποιαδήποτε στιγμή το επιθυμεί ο χρήστης.

7.2.2 Editor E – Wizard

Συγκριτικά με τις βελτιώσεις που χρήζει ο Editor E-Wizard σε σχέση με τον Editor Domino διαφέρουν. Στον Editor E-Wizard τα σημεία που χρήζουν βελτίωση δεν συμπίπτουν με τον Editor Domino. Η διαγραφή και η ανανέωση κατά τη διαδικασία ανάπτυξης του κανόνα μπορεί να πραγματοποιηθεί οποιαδήποτε στιγμή, για οποιαδήποτε έξυπνη συσκευή είτε στο κομμάτι του συμβάντος είτε στο κομμάτι της δράσης και ο αριθμός των έξυπνων συσκευών είναι απροσδιόριστος και για τα δύο μέρη του κανόνα.

Τα σημεία που μπορούν να βελτιωθούν είναι :

- Καλύτερη απεικόνιση των εικονιδίων στην αναγραφή του κανόνα στο κομμάτι της δράσης. Για παράδειγμα, Αν ο χρήστης προσθέσει μια έξυπνη συσκευή η οποία παρέχει πολλές ενέργειες και την παραμετροποίηση με σχεδόν όλες τις ενέργειες που παρέχει τότε το εικονίδιο συμπιέζεται με αποτέλεσμα να μην είναι τόσο καλά ευδιάκριτο.
- Πιο σωστή δομή και στοίχιση στην απεικόνιση των ενεργειών της έξυπνης συσκευής στο κομμάτι της δράσης.

Επιπλέον, η αναγραφή των αποθηκευμένων κανόνων που δημιουργήθηκαν στον Editor Domino διαθέτει το εικονίδιο της έξυπνης συσκευής που έχει οριστεί ως δράση. Σε αντίθεση με τον Editor E-Wizard που στην αναγραφή δεν υπάρχει η απεικόνιση της/των έξυπνης/ων συσκευής/ων που έχουν οριστεί ως δράση/εις. Επειδή στον Editor E-Wizard δίνεται η δυνατότητα προσθήκης περισσότερων εκ της μίας έξυπνης συσκευής ως δράση καθιστούσε δύσκολο το γεγονός προσθήκης εικονιδίου έξυπνης συσκευής. Το γεγονός αυτό λύθηκε ως ένα βαθμό διότι, στο μέρος της δράσης, πρώτα αναγράφεται η έξυπνη συσκευή σε απλό κείμενο και έπειτα οι ενέργειες της. Ωστόσο, θα ήταν επιθυμητό να υπάρχουν και τα εικονίδια των έξυπνων συσκευών που τις πραγματοποιούν.

7.2.3 Αποθηκευμένοι Κανόνες

Όταν αποθηκεύεται ένας κανόνας προστίθεται στη λίστα με τους ήδη αποθηκευμένους κανόνες αν υπάρχουν. Σε κάθε γραμμή της λίστα απεικονίζεται μία εικόνα που παραπέμπει σε μορφή που θα

μπορούσε να απεικονίζεται μία συνταγή (recipe), το όνομα του κανόνα και το εικονίδιο του Editor στον οποίο φτιάχτηκε ο κανόνας (βλέπε Ενότητα «4.3.5 Αποθηκευμένοι Κανόνες»). Η προσθήκη της ημερομηνίας (ημέρα-μήνας-έτος) και της ώρα αποθήκευσης του κανόνα θα παρείχε μία σημαντική επιπλέον πληροφορία προς του χρήστες.

7.2.4 Κάτοψη (Floorspan)

Στο Κεφάλαιο που συμπεριλαμβάνονται οι πληροφορίες σχετικά με την λειτουργία της Κάτοψης (Floorspan) (βλέπε Ενότητα «4.3.4 Κάτοψη (Floorspan)») παρατηρείται ότι η περιήγηση στο οριζόντιο μενού των έξυπνων συσκευών που απεικονίζονται ως dominos στην επιλογή της επεξεργασίας της μεταφορτωμένης εικόνας γίνεται με την χρήση των δύο μαύρων βελών. Η κύλιση του οριζόντιου μενού χωρίς την χρήση των δύο μαύρων βελών θα αποτελούσε μία αξιοσημείωτη βελτίωση για την επιλογή της επεξεργασίας της εικόνας κάτοψης. Επίσης, η προσαρμογή των μεταφορτωμένων εικόνων στο περιβάλλον της επεξεργασίας γίνεται σύμφωνα με ακριβείς διαστάσεις που έχουν οριστεί στο πρόγραμμα της εφαρμογής. Ο κύριος λόγος του ορισμού των ακριβών διαστάσεων είναι η αποφυγή διαστρεβλώσεων της εικόνας αλλά και γενικότερα ολόκληρου του περιβάλλοντος. Παρ' όλα αυτά η προσθήκη της εικόνας με τις διαστάσεις που ήδη έχει αναγνωρίζεται ως μια βελτίωση του περιβάλλοντος της επεξεργασίας της μεταφορτωμένης εικόνας κάτοψης.

7.2.5 Animation

Συνολικά η εφαρμογή μας διαθέτει πέντε βίντεο εικονικής απεικόνισης κανόνα για πέντε συγκεκριμένους κανόνες για τον Editor Domino. Η δυναμική κατασκευή βίντεο για εικονική απεικόνιση κανόνων για οποιεσδήποτε έξυπνες συσκευές και για οποιεσδήποτε ενέργειες αυτές πραγματοποιούν, για κανόνες που δημιουργούνται στον Editor Domino, θα ήταν μία εξαιρετική βελτίωση του συστήματος μας και θα βελτίωνε σε μεγάλο βαθμό την εμπειρία του χρήστη με τον συγκεκριμένο Editor.

7.2.6 Διαχωρισμός έξυπνων συσκευών

Οι έξυπνες συσκευές που λαμβάνουν χώρα στο μενού των έξυπνων συσκευών για την δημιουργία των κανόνων στον Editor Domino διαχωρίζονται αναλόγως την τοποθεσία που βρίσκονται. Ο διαχωρισμός είναι στατικός και δεν έχει πραγματοποιηθεί από τους χρήστες αλλά εξ' ορισμού από το σύστημα. Η επιλογή μια νέας επιπλέον λειτουργίας στην εφαρμογή που θα δίνει την ευκαιρία ορισμού της τοποθεσίας των έξυπνων συσκευών σ' έναν χώρο από τους χρήστες, πριν την αλληλεπίδραση τους με τους Editors, θα ανέβαζε ένα επίπεδο την εφαρμογή. Ο διαχωρισμός αυτός δεν γίνεται στο μενού των έξυπνων συσκευών στον Editor E-Wizard διότι θέλουμε να δείξουμε και έναν άλλον διαφορετικό, πιο απλό, τρόπο απεικόνισης των έξυπνων συσκευών στο μενού.

7.2.7 Εκπαίδευση στους Editors

Πριν την ενασχόληση των χρηστών με τα περιβάλλοντα δημιουργίας κανόνων προτείνεται η περιήγηση στην επιλογή «Εκπαίδευση στους Editors». Κατά τη διάρκεια της περιήγησης τους στην συγκεκριμένη επιλογή, στο τέλος αυτής για κάθε Editor, θα ήταν αξιόλογη η προσθήκη ενός περιβάλλοντος δημιουργίας πρώτου κανόνα με συγκεκριμένα και προκαθορισμένα βήματα για κάθε Editor.

7.2.8 Επικοινωνία πραγματικού χρόνου εφαρμογής με έξυπνες συσκευές

Ο βαθμός ανάπτυξης στον οποίο βρίσκεται η εφαρμογή δεν διαθέτει την δυνατότητα επικοινωνίας της εφαρμογής με τις έξυπνες συσκευές. Το βήμα που θα έκανε την εφαρμογή πιο ολοκληρωμένη θα ήταν η συγκεκριμένη βελτίωση. Θα έδινε την ευκαιρία της ρεαλιστικής απεικόνισης του κανόνα είτε έχει δημιουργηθεί στον Editor Domino είτε στον Editor E-Wizard.

8

Συμπεράσματα

Η εργασία μας με τίτλο “ Ανάπτυξη και αξιολόγηση εργαλείου για τη δημιουργία IoT εφαρμογών με τη συμμετοχή τελικών χρηστών ” αποτελεί ένα κομμάτι έρευνας με στόχο την γραφική αναπαράσταση και δημιουργία κανόνων για τον προγραμματισμό ενός έξυπνου χώρου. Η ερευνά μας, δίνει έμφαση στη δημιουργία φιλικού αλλά και κατανοητού περιβάλλοντος προς τους τελικούς χρήστες με σκοπό τη δημιουργία κανόνων. Έχουμε αναλύσει τα τεχνολογικά κριτήρια διαθεσιμότητάς της σε ένα περιβάλλον παραγωγής καθώς δημιουργήσαμε την πλατφόρμα IoT για την διάθεση της σε κινητές συσκευές.

8.1 Πλατφόρμα IoT

Κληθήκαμε να αναπτύξουμε την δικιά μας ιδέα αναπαράστασης κανόνων, η οποία ήταν η αναπαράσταση των έξυπνων συσκευών υπό την μορφή Domino, καθώς η λογική συσχέτιση αυτών εξαρτάται από την τοποθέτηση τους στον τρισδιάστατο χώρο. Με αυτό τον τρόπο θα θέλαμε να δώσουμε στους χρήστες την ευκαιρία να μπορούν να οραματιστούν και να συσχετίσουν τις έξυπνες συσκευές μεταξύ τους τοποθετώντας αυτές στον χώρο με κατάλληλη σειρά και στοίχιση.

Στο κομμάτι της έρευνας αυτής, κριτήριο για την βελτίωση του συστήματος ήταν συμμετοχή των τελικών χρηστών. Με βάση την απόδοση αλλά και την γνώμη των τελικών χρηστών, προβήκαμε σε λειτουργικές και σχεδιαστικές αλλαγές με στόχο την βελτίωση της πλατφόρμας.

Στην πλατφόρμα εκτός από το περιβάλλον δημιουργίας του Domino Editor, κληθήκαμε να υλοποιήσουμε και έναν ακόμα Editor, τον E-Wizard. Στόχος ήταν η σύγκριση των δυο με σκοπό την αξιολόγηση και βελτίωση της δικής μας ιδέας ως προς την λειτουργικότητα και στην αναπαράσταση δεδομένων του περιβάλλοντος.

Επιπροσθέτως, συμπεριλάβαμε την εκπαίδευση των χρηστών στους 2 αυτούς Editors με τη χρήση παραδειγμάτων και σεναρίων έκφρασης κανόνων που αφορούν εφαρμογές χρήσιμες στην πραγματικότητα ενός καθημερινού ανθρώπου.

Κρίναμε ακόμη αναγκαία την ενσωμάτωση και επεξεργασία της πιθανής κάτοψης του smart home στην πλατφόρμα. Αφού οι χρήστες ανεβάσουν την υπάρχουσα εικόνα υπό την μορφή κάτοψης ενός διαμερίσματος μπορούν να την επεξεργαστούν προσθέτοντας έξυπνες συσκευές πάνω στην ίδια την εικόνα. Αυτές οι έξυπνες συσκευές στη συνέχεια εμφανίζονται υπό την μορφή Domino στοιχείων στο εργαλείο τελικού χρήστη.

Η πλατφόρμα υποστηρίζει την αποθήκευση των κανόνων καθώς και την προβολή τους σε φυσική γλώσσα. Αναγράφεται το όνομα του κανόνα που έχει δώσει ο χρήστης καθώς και ο Editor που έχει επιλέξει για την δημιουργία του. Επίσης έχουμε συμπεριλάβει ένα θεωρητικό σενάριο ικανοποίησης του κανόνα στο οποίο οι χρήστες ενημερώνονται πως ο κανόνας ενεργοποιήθηκε, με τη χρήση animation.

Η γραφική αναπαράσταση των δεδομένων στην πλατφόρμα κρίνεται σημαντική και για τον σκοπό αυτό προβήκαμε στη χρήση εξωτερικών βιβλιοθηκών, καθώς συμπεριλάβαμε τη χρήση emoji, στην αναπαράσταση του κανόνα σε φυσική γλώσσα. Με τον συνδυασμό των δυο θα θέλαμε να πετύχουμε μια ικανοποιητική και ευπαρουσίαστη γραφική απεικόνιση πληροφοριών.

Προσθέσαμε στην πλατφόρμα την δυνατότητα ενημέρωσης του χρήστη για υπάρχουσες προκαθορισμένες έξυπνες συσκευές με όλες τις δυνατές παραμετροποιήσεις που μπορούν να δεχθούν. Σε ένα σύστημα παραγωγής, σε αυτήν την λειτουργία θα δινόταν η δυνατότητα στον χρήστη να προσθέσει τις δικές του έξυπνες συσκευές.

8.2 *Ανάδραση Χρηστών*

Η πλειοψηφία (90%) των χρηστών βρήκε το σύστημα αρκετά ενδιαφέρον, εύκολο, και φιλικό προς χρήση. Η ιδέα αναπαράστασης των έξυπνων συσκευών υπό την μορφή Domino καθώς και η δημιουργία κανόνων για αυτοματοποιημένες διαδικασίες στο χώρο, τους άρεσε αρκετά και σημείωσαν πως θα ήταν έναν σύστημα το οποίο θα χρησιμοποιούσαν εάν διέθεταν έξυπνες συσκευές.

8.3 *Το Μέλλον των Έξυπνων Κατοικιών*

Σήμερα, παρατηρούμε πως η τεχνολογία σε σύγκριση με τα προηγούμενα χρόνια εξελίσσεται με ραγδαίους ρυθμούς. Ήδη αρκετοί εσωτερικοί αλλά και μη χώροι ανά τον κόσμο, χρησιμοποιούν τις έξυπνες συσκευές για την παρακολούθηση και την καταγραφή δεδομένων και στατιστικών στοιχείων. Δημιουργείται με αυτόν τον τρόπο ένα διάχυτο υπολογιστικό περιβάλλον για τη συλλογή πληροφοριών και την εξαγωγή αποτελεσμάτων. Θεωρούμε ειδικότερα πως το Internet of Things σε κάποια χρόνια θα μπει στην καθημερινότητα των καταναλωτών και θα φέρει ριζικές αλλαγές σε ιδιωτικούς και δημόσιους τομείς. Σε αυτό το πλαίσιο, η διάθεση εργαλείων τελικού χρήστη, όπως αυτό που πραγματεύεται η παρούσα διπλωματική εργασία, θα δώσει τη δυνατότητα στους απλούς χρήστες να συμμετέχουν δημιουργικά στη διαμόρφωση των έξυπνων περιβαλλόντων, σύμφωνα με τις δικές τους ανάγκες, αυξάνοντας έτσι και το βαθμό αποδοχής της νέας τεχνολογίας.

Βιβλιογραφία

- [1] Cappiello, C., Matera, M., & Picozzi, M. (2015). A UI-centric approach for the end-user development of multidevice mashups. *ACM Transactions on the Web (TWEB)*, 9(3), 1-40.
- [2] Ghiani, G., Manca, M., Paternò, F., & Santoro, C. (2017). Personalization of context-dependent applications through trigger-action rules. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(2), 1-33.
- [3] Fogli, D., Lanzilotti, R., & Piccinno, A. (2016, July). End-user development tools for the smart home: a systematic literature review. In *International Conference on Distributed, Ambient, and Pervasive Interactions* (pp. 69-79). Springer, Cham.
- [4] Desolda, G., Ardito, C., & Matera, M. (2017). Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(2), 1-52.
- [5] Mi, X., Qian, F., Zhang, Y., & Wang, X. (2017, November). An empirical characterization of IFTTT: ecosystem, usage, and performance. In *Proceedings of the 2017 Internet Measurement Conference* (pp. 398-404).
- [6] Rahmati, A., Fernandes, E., Jung, J., & Prakash, A. (2017). IFTTT vs. Zapier: A comparative study of trigger-action programming frameworks. *arXiv preprint arXiv:1709.02788*.
- [7] Paternò, F., & Lucci, G. User Conceptual Models of Event-Action IoT Applications.
- [8] Blackstock, M., & Lea, R. (2014, October). Toward a distributed data flow platform for the web of things (distributed node-red). In *Proceedings of the 5th International Workshop on Web of Things* (pp. 34-39).
- [9] spring.io. (2020, September 28). Retrieved from Spring Framework Documantion: <https://spring.io/guides>
- [10] Pitatzis, S., Drosos, N., Goumopoulos, C., & Kameas, A. (2020, July). AmIoT: A Microservices-based IoT Platform to Orchestrate AmI Environments. In *2020 16th International Conference on Intelligent Environments (IE)* (pp. 21-28). IEEE.
- [11] Goumopoulos, C., & Mavrommati, I. (2020). A framework for pervasive computing applications based on smart objects and end user development. *Journal of Systems and Software*, 162, 110496.
- [12] Coutaz, J., & Crowley, J. L. (2016). A first-person experience with end-user development for smart homes. *IEEE Pervasive Computing*, 15(2), 26-39.

Παράρτημα I – System Usability Scale

1. Νομίζω ότι θα ήθελα να χρησιμοποιώ αυτή την εφαρμογή συχνά.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

2. Βρήκα αυτή την εφαρμογή αδικαιολόγητα περίπλοκη.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

3. Σκέφτηκα ότι αυτή η εφαρμογή ήταν εύκολη στη χρήση.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

4. Νομίζω ότι θα χρειαστώ βοήθεια από κάποιον τεχνικό για να είμαι σε θέση να χρησιμοποιήσω αυτή την εφαρμογή.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

5. Βρήκα τις διάφορες λειτουργίες σ' αυτή την εφαρμογή καλά ολοκληρωμένες.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

6. Σκέφτηκα ότι υπήρχε μεγάλη ασυνέπεια σ' αυτή την εφαρμογή.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

7. Φαντάζομαι ότι οι περισσότεροι άνθρωποι θα μάθουν να χρησιμοποιούν αυτή την εφαρμογή πολύ γρήγορα.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

8. Βρήκα αυτή την εφαρμογή πολύ περίπλοκη/δύσκολη στη χρήση.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

9. Ένιωσα πολύ σίγουρος/η χρησιμοποιώντας αυτή την εφαρμογή.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

1	2	3	4	5

10. Χρειάστηκε να μάθω πολλά πράγματα πριν να μπορέσω να ξεκινήσω με αυτή την εφαρμογή.

Διαφωνώ
απολύτως

Συμφωνώ
απολύτως

--	--	--	--	--

1

2

3

4

5