



Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

## Διπλωματική Εργασία

# Σχεδιασμός και υλοποίηση πληροφοριακού συστήματος διαχείρισης εθελοντών και απινιδωτών

Design and implementation of an informative management  
system of volunteers and defibrillators



**Μυρωνίδης Αλέξανδρος**

Επιβλέπων Καθηγητής: **Δρ. Μηνάς Δασυγένης**  
Εργαστήριο Ψηφιακών Συστημάτων  
και Αρχιτεκτονικής Υπολογιστών



## Περιεχόμενα

Κατάλογος εικόνων.....	6
Κατάλογος πινάκων .....	8
Περίληψη.....	10
Abstract .....	12
Δήλωση Πνευματικών Δικαιωμάτων.....	14
Ευχαριστίες.....	15
<b>Κεφάλαιο 1ο – Εισαγωγή.....</b>	<b>16</b>
1.1 Περιγραφή σημερινής κατάστασης.....	16
1.2 Ιδέα και σκοπός υλοποίησης.....	17
1.3 Αντίστοιχα συστήματα και συμπέρασμα υλοποίησης.....	18
1.3.1 AED Locations v2.....	18
1.3.2 Permis de sauver.....	19
1.3.3 AED Alert .....	20
1.3.4 Staying Alive.....	21
1.3.5 DAEdove.....	22
1.3.6 Responders .....	23
1.3.7 SAUV Life .....	24
1.4 Ανάλυση και συμπεράσματα αντίστοιχων εφαρμογών.....	24
1.5 Σύνοψη διπλωματικής εργασίας.....	26
<b>Κεφάλαιο 2ο – Θεωρητικό υπόβαθρο .....</b>	<b>28</b>
2.1 Προγραμματισμός διαδικτύου.....	28
2.1.1 HTML .....	29
2.1.2 CSS.....	30
2.1.3 Javascript.....	31
1.1.1     2.1.4 MYSQL.....	33
2.1.5 PHP .....	34
2.2 Δομές (frameworks), βιβλιοθήκες (libraries), μεταγλωττιστές (compilers), εργαλεία.....	35
2.2.1 Slim Framework.....	35
2.2.2 Material-ui και responsive design .....	37
2.2.3 jQuery.....	37
2.2.4 React.....	38
2.2.5 Fetch.....	39
2.2.6 Redux .....	40
2.2.7 Webpack.....	41
2.2.8 Babel .....	41
2.2.9 Pusher API.....	41
2.2.10 Google-Maps.....	42
2.2.11 React-Native.....	42
2.4 Πλατφόρμες Ανάπτυξης Λογισμικού.....	44
2.4.1 Visual Studio Code.....	44
2.4.2 Expo .....	45
2.4.3 React Native Debugger.....	45

2.5 Σύνοψη κεφαλαίου.....	46
<b>Κεφάλαιο 3ο – Δομικά μέρη συστήματος .....</b>	<b>48</b>
3.1 Απαιτήσεις συστήματος.....	48
3.2 Περιπτώσεις χρήσης (use cases).....	50
3.3 Ανάλυση βάσης δεδομένων.....	51
3.3.1 E_R και σχεσιακό διάγραμμα βάσης δεδομένων .....	52
3.3.2 Δομή πίνακα “Εθελοντής”.....	54
3.3.3 Δομή πίνακα “ΕΚΑΒ”.....	56
3.2.4 Δομή πίνακα “Περιστατικό” .....	57
3.3.5 Δομή πίνακα “Απινιδωτής”.....	58
3.3.6 Δομή πίνακα “ασθενής” .....	59
3.4 Ασφάλεια συστήματος.....	60
3.5 Σύνοψη κεφαλαίου.....	62
<b>Κεφάλαιο 4ο – Λειτουργικότητα συστήματος .....</b>	<b>64</b>
4.1 Ανάλυση λειτουργιών διαδικτυακής εφαρμογής και διεπαφές χρήστη.....	64
4.1.1 Αρχική σελίδα διαδικτυακής εφαρμογής.....	65
4.1.2 Κεντρικό περιβάλλον εξουσιοδοτημένου χρήστη (διαχειριστής).....	66
4.1.3 Κεντρική Σελίδα Χάρτη .....	67
4.1.4 Σελίδα Προφίλ Διαχειριστή.....	69
4.1.5 Σελίδα Διαχειριστών .....	70
4.1.6 Σελίδα Εθελοντών .....	75
4.1.7 Σελίδα Απινιδωτών.....	77
4.1.7 Σελίδα Ασθενών .....	80
4.1.8 Σελίδα Περιστατικών .....	80
4.2 Ανάλυση λειτουργιών εφαρμογής κινητού και διεπαφές χρήστη .....	82
4.2.1 Σελίδα Σύνδεσης Εθελοντή .....	83
4.2.2 Κεντρικό περιβάλλον εξουσιοδοτημένου χρήστη – Οθόνη Χάρτη.....	87
4.2.3 Οθόνη Προφίλ Εθελοντή .....	89
4.2.4 Κεντρική λειτουργία Εφαρμογής .....	92
<b>Κεφάλαιο 5ο – Επίλογος .....</b>	<b>104</b>
<b>5.1 Σύνοψη και συμπεράσματα .....</b>	<b>104</b>
5.2 Προβλήματα που προέκυψαν και η αντιμετώπιση τους .....	105
5.3 Μετρικά κώδικα συστήματος .....	105
5.3.1 Μετρικά κώδικα διαδικτυακής εφαρμογής .....	105
5.3.2 Μετρικά κώδικα κινητής εφαρμογής εφαρμογής .....	106
5.3.3 Μετρικά κώδικα backend .....	106
5.4 Ανάλυση SWOT .....	107
5.5 Μελλοντικές επεκτάσεις .....	108
5.6 Συμπεράσματα .....	109
5.7 Σύνοψη κεφαλαίου.....	109
<b>Βιβλιογραφία .....</b>	<b>110</b>

<b>Παράρτημα .....</b>	<b>112</b>
<b>Οδηγίες εγκατάστασης σε τοπικό υπολογιστή (localhost) .....</b>	<b>112</b>
<b>Οδηγίες κινητής εφαρμογής .....</b>	<b>112</b>
<b>Οδηγίες εγκατάστασης σε shared hosting provider .....</b>	<b>113</b>



## Κατάλογος εικόνων

ΕΙΚΟΝΑ 1: ΕΙΚΟΝΙΔΙΟ ΕΚΤΕΛΕΣΗΣ ΑΝΤΙΣΤΟΙΧΗΣ ΕΦΑΡΜΟΓΗΣ .....	18
ΕΙΚΟΝΑ 2: ΕΙΚΟΝΙΔΙΟ ΕΚΤΕΛΕΣΗΣ ΔΕΥΤΕΡΗΣ ΑΝΤΙΣΤΟΙΧΗΣ ΕΦΑΡΜΟΓΗΣ .....	19
ΕΙΚΟΝΑ 3: ΕΙΚΟΝΙΔΙΟ ΕΚΤΕΛΕΣΗΣ ΤΡΙΤΗΣ ΑΝΤΙΣΤΟΙΧΗΣ ΕΦΑΡΜΟΓΗΣ .....	20
ΕΙΚΟΝΑ 4: ΕΙΚΟΝΙΔΙΟ ΕΚΤΕΛΕΣΗΣ ΤΕΤΑΡΤΗΣ ΑΝΤΙΣΤΟΙΧΗΣ ΕΦΑΡΜΟΓΗΣ.....	21
ΕΙΚΟΝΑ 5: ΕΙΚΟΝΙΔΙΟ ΕΚΤΕΛΕΣΗΣ ΠΕΜΠΤΗΣ ΑΝΤΙΣΤΟΙΧΗΣ ΕΦΑΡΜΟΓΗΣ.....	22
ΕΙΚΟΝΑ 6: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΕΚΤΗΣ ΑΝΤΙΣΤΟΙΧΗΣ ΕΦΑΡΜΟΓΗΣ .....	23
ΕΙΚΟΝΑ 7: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΕΒΔΟΜΗΣ ΑΝΤΙΣΤΟΙΧΗΣ ΕΦΑΡΜΟΓΗΣ .....	24
ΕΙΚΟΝΑ 8: ΠΑΡΑΔΕΙΓΜΑ ΣΥΝΤΑΞΗΣ CSS .....	31
ΕΙΚΟΝΑ 9: ΠΑΡΑΔΕΙΓΜΑ ΣΥΝΤΑΞΗΣ JAVASCRIPT .....	32
ΕΙΚΟΝΑ 10: ΠΑΡΑΔΕΙΓΜΑ SQL ΕΝΤΟΛΗΣ .....	33
ΕΙΚΟΝΑ 11: ΠΑΡΑΔΕΙΓΜΑ ΣΥΝΤΑΞΗΣ PHP .....	35
ΕΙΚΟΝΑ 12: ΠΑΡΑΔΕΙΓΜΑ ΕΝΤΟΛΗΣ JQUERY .....	38
ΕΙΚΟΝΑ 13: ΠΑΡΑΔΕΙΓΜΑ REACT COMPONENT .....	39
ΕΙΚΟΝΑ 14: ΠΑΡΑΔΕΙΓΜΑ ΕΝΤΟΛΗΣ FETCH .....	40
ΕΙΚΟΝΑ 15: ΠΑΡΑΔΕΙΓΜΑ ΛΕΙΤΟΥΡΓΙΑΣ REACT NATIVE DEBUGGER.....	46
ΕΙΚΟΝΑ 16: Ε_Ρ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ HEARTBIT .....	53
ΕΙΚΟΝΑ 17: ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ HEARTBIT.....	53
ΕΙΚΟΝΑ 18: ΑΡΧΙΚΗ ΣΕΛΙΔΑ – ΣΕΛΙΔΑ ΣΥΝΔΕΣΗΣ .....	66
ΕΙΚΟΝΑ 19: ΑΡΧΙΚΗ ΣΕΛΙΔΑ – ΜΗΝΥΜΑ ΕΣΦΑΛΜΕΝΩΝ ΣΤΟΙΧΕΙΩΝ ΣΥΝΔΕΣΗΣ .....	66
ΕΙΚΟΝΑ 20: ΚΕΝΤΡΙΚΗ ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΤΗ – DRAWER ΠΛΟΗΓΗΣΗΣ.....	67
ΕΙΚΟΝΑ 21: ΣΕΛΙΔΑ ΧΑΡΤΗ – ΑΥΤΟCΟΜΠΛΕΤΕ ΑΝΑΖΗΤΗΣΗΣ ΔΙΕΥΘΥΝΣΕΩΝ .....	68
ΕΙΚΟΝΑ 22: ΣΕΛΙΔΑ ΧΑΡΤΗ – ΠΡΟΣΘΗΚΗ ΡΙΝ ΣΤΗΝ ΑΠΑΙΤΟΥΜΕΝΗ ΔΙΕΥΘΥΝΣΗ .....	68
ΕΙΚΟΝΑ 23: ΣΕΛΙΔΑ ΧΑΡΤΗ – ΠΡΟΣΘΗΚΗ ΠΕΡΙΣΤΑΤΙΚΟΥ ΜΕ ΤΟ ΑΝΤΙΣΤΟΙΧΟ ΜΗΝΥΜΑ.....	68
ΕΙΚΟΝΑ 24: ΣΕΛΙΔΑ ΧΑΡΤΗ – ΠΑΡΑΔΕΙΓΜΑ ΣΥΜΠΛΗΡΩΣΗΣ ΛΑΘΟΣ ΔΙΕΥΘΥΝΣΗΣ ΚΑΙ ΑΝΤΙΣΤΟΙΧΟ ΜΗΝΥΜΑ.....	69
ΕΙΚΟΝΑ 25: ΣΕΛΙΔΑ ΠΡΟΦΙΛ ΔΙΑΧΕΙΡΙΣΤΗ.....	69
ΕΙΚΟΝΑ 26: ΣΕΛΙΔΑ ΠΡΟΦΙΛ ΔΙΑΧΕΙΡΙΣΤΗ – ΦΟΡΜΑ ΔΙΑΧΕΙΡΙΣΤΗ .....	70
ΕΙΚΟΝΑ 27: ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΤΩΝ .....	70
ΕΙΚΟΝΑ 28: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΣΕΛΙΔΟΠΟΙΗΣΗ.....	71
ΕΙΚΟΝΑ 29: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΚΟΥΜΠΙ ΠΡΟΣΘΗΚΗΣ ΔΙΑΧΕΙΡΙΣΤΗ & TOOLTIP .....	71
ΕΙΚΟΝΑ 30: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΚΑΡΤΕΛΑ ΕΙΣΑΓΩΓΗΣ ΔΙΑΧΕΙΡΙΣΤΗ.....	72
ΕΙΚΟΝΑ 31: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΜΗΝΥΜΑ ΣΦΑΛΜΑΤΟΣ ΣΤΗΝ ΕΙΣΑΓΩΓΗ ΔΙΑΧΕΙΡΙΣΤΗ (VALIDATION).....	72
ΕΙΚΟΝΑ 32: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΕΙΚΟΝΙΔΙΟ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΙΑΧΕΙΡΙΣΤΗ & TOOLTIP.....	73
ΕΙΚΟΝΑ 33: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΠΑΡΑΔΕΙΓΜΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΙΑΧΕΙΡΙΣΤΗ .....	73
ΕΙΚΟΝΑ 34: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΚΑΡΤΕΛΑ ΔΙΑΧΕΙΡΙΣΤΗ .....	74
ΕΙΚΟΝΑ 35: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΔΙΑΧΕΙΡΙΣΤΩΝ – ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΖΗΤΗΣΗΣ ΔΙΑΧΕΙΡΙΣΤΗ.....	75
ΕΙΚΟΝΑ 36: ΣΕΛΙΔΑ ΕΘΕΛΟΝΤΩΝ.....	76
ΕΙΚΟΝΑ 37: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΕΘΕΛΟΝΤΩΝ - ΠΑΡΑΔΕΙΓΜΑ ΕΝΕΡΓΟΠΟΙΗΣΗΣ ΔΙΚΑΙΩΜΑΤΩΝ ΕΘΕΛΟΝΤΗ .....	76
ΕΙΚΟΝΑ 38: ΣΕΛΙΔΑ ΠΙΝΑΚΑ ΕΘΕΛΟΝΤΩΝ - ΠΑΡΑΔΕΙΓΜΑ ΑΠΕΝΕΡΓΟΠΟΙΗΣΗΣ ΔΙΚΑΙΩΜΑΤΩΝ ΕΘΕΛΟΝΤΗ .....	77
ΕΙΚΟΝΑ 39: ΣΕΛΙΔΑ ΑΠΙΝΙΔΩΤΩΝ .....	77
ΕΙΚΟΝΑ 40 :ΠΑΡΑΔΕΙΓΜΑ ΑΠΩΛΕΙΑΣ ΕΝΟΣ ΑΠΙΝΙΔΩΤΗ ΑΠΟ ΤΟ ΚΑΙ ΑΛΛΑΓΗ ΚΑΤΑΧΩΡΗΣΗΣ ΑΠΟ ARDUINO.....	78
ΕΙΚΟΝΑ 41: ΠΑΡΑΔΕΙΓΜΑ ΚΛΕΙΔΩΜΑΤΟΣ ΕΝΟΣ ΑΠΙΝΙΔΩΤΗ ΑΠΟ ΤΟ ΚΑΙ ΑΛΛΑΓΗ ΚΑΤΑΧΩΡΗΣΗΣ ΑΠΟ ARDUINO.....	79
ΕΙΚΟΝΑ 42: ΠΑΡΑΔΕΙΓΜΑ ΜΗΝΥΜΑΤΟΣ ΥΨΗΛΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΕΝΟΣ ΑΠΙΝΙΔΩΤΗ .....	79
ΕΙΚΟΝΑ 43: ΠΑΡΑΔΕΙΓΜΑ ΜΗΝΥΜΑΤΟΣ ΧΑΜΗΛΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΕΝΟΣ ΑΠΙΝΙΔΩΤΗ .....	79
ΕΙΚΟΝΑ 44: ΣΕΛΙΔΑ ΑΣΘΕΝΩΝ .....	80
ΕΙΚΟΝΑ 45: ΣΕΛΙΔΑ ΠΕΡΙΣΤΑΤΙΚΩΝ.....	81
ΕΙΚΟΝΑ 46: ΣΕΛΙΔΑ ΠΕΡΙΣΤΑΤΙΚΩΝ – ΠΑΡΑΔΕΙΓΜΑ ΕΜΦΑΝΙΣΗΣ ΩΡΑΣ ΑΝΤΑΠΟΚΡΙΣΗΣ.....	81
ΕΙΚΟΝΑ 47: ΚΟΥΜΠΙ ΑΠΟΣΥΝΔΕΣΗΣ ΧΡΗΣΤΗ .....	82
ΕΙΚΟΝΑ 48: ΕΙΚΟΝΙΔΙΟ ΕΦΑΡΜΟΓΗΣ ΚΙΝΗΤΟΥ .....	82
ΕΙΚΟΝΑ 49: ΚΟΥΜΠΙ ΑΠΟΣΥΝΔΕΣΗΣ ΕΦΑΡΜΟΓΗΣ.....	83
ΕΙΚΟΝΑ 50: ΟΘΟΝΗ ΣΥΝΔΕΣΗΣ ΕΦΑΡΜΟΓΗΣ.....	83
ΕΙΚΟΝΑ 51: ΜΗΝΥΜΑ ΑΠΟΤΥΧΗΜΕΝΗΣ ΠΡΟΣΠΑΘΕΙΑΣ ΕΙΣΟΔΟΥ .....	84
ΕΙΚΟΝΑ 52: ΟΘΟΝΗ ΑΝΑΚΤΗΣΗΣ ΚΩΔΙΚΟΥ .....	84
ΕΙΚΟΝΑ 53: ΜΗΝΥΜΑ ΛΑΘΟΣ ΕΙΣΑΓΩΓΗΣ E-MAIL ΓΙΑ ΑΝΑΚΤΗΣΗ ΚΩΔΙΚΟΥ.....	85
ΕΙΚΟΝΑ 54: E-MAIL ΜΕ ΣΥΝΔΕΣΜΟ ΕΠΑΝΑΦΟΡΑΣ ΚΩΔΙΚΟΥ .....	85

ΕΙΚΟΝΑ 55: ΣΕΛΙΔΑ ΕΠΑΝΑΦΟΡΑΣ ΚΩΔΙΚΟΥ ΕΘΕΛΟΝΤΗ.....	86
ΕΙΚΟΝΑ 56: ΜΗΝΥΜΑ ΛΑΘΟΣ ΣΥΜΠΛΗΡΩΣΗΣ ΝΕΟΥ ΚΩΔΙΚΟΥ.....	86
ΕΙΚΟΝΑ 57: ΟΘΟΝΗ ΧΑΡΤΗ ΕΦΑΡΜΟΓΗΣ ΚΙΝΗΤΟΥ .....	87
ΕΙΚΟΝΑ 58: ΕΙΔΟΠΟΙΗΣΗ ΚΛΗΣΗΣ ΑΜΕΣΗΣ ΒΟΗΘΕΙΑΣ ΕΚΑΒ .....	88
ΕΙΚΟΝΑ 59: ΟΘΟΝΗ ΠΡΟΦΙΛ ΕΘΕΛΟΝΤΗ.....	89
ΕΙΚΟΝΑ 60: ΜΠΑΡΑ ΑΠΕΝΕΡΓΟΠΟΙΗΣΗΣ ΧΡΗΣΤΗ .....	90
ΕΙΚΟΝΑ 61: ΚΑΡΤΕΛΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΠΕΔΙΟΥ .....	90
ΕΙΚΟΝΑ 62: ΜΗΝΥΜΑ ΛΑΘΟΣ ΕΙΣΑΓΩΓΗΣ ΚΩΔΙΚΟΥ ΣΤΗΝ ΚΑΡΤΕΛΑ ΕΠΕΞΕΡΓΑΣΙΑΣ .....	91
ΕΙΚΟΝΑ 63: ΜΗΝΥΜΑ ΛΑΘΟΣ ΕΙΣΑΓΩΓΗΣ ΝΕΟΥ ΚΩΔΙΚΟΥ ΣΤΗΝ ΚΑΡΤΕΛΑ ΕΠΕΞΕΡΓΑΣΙΑΣ .....	91
ΕΙΚΟΝΑ 64: ΕΙΔΟΠΟΙΗΣΗ ΝΕΟΥ ΠΕΡΙΣΤΑΤΙΚΟΥ ΣΕ ΚΑΤΑΣΤΑΣΗ ΕΦΑΡΜΟΓΗΣ ΣΤΟ ΠΑΡΑΣΚΗΝΙΟ .....	92
ΕΙΚΟΝΑ 65: ΚΑΡΤΕΛΑ ΝΕΟΥ ΠΕΡΙΣΤΑΤΙΚΟΥ ΜΕ ΜΟΡΦΗ ΕΙΔΟΠΟΙΗΣΗΣ.....	93
ΕΙΚΟΝΑ 66: ΠΛΗΡΗΣΗ ΠΡΟΣ ΤΟΝ ΑΠΙΝΙΔΩΤΗ ΚΑΙ ΤΟ ΠΕΡΙΣΤΑΤΙΚΟ .....	94
ΕΙΚΟΝΑ 67: ΟΔΗΓΙΕΣ ΠΡΟΣ ΤΟ ΠΡΩΤΟ ΣΗΜΕΙΟ ΠΟΥ ΒΡΙΣΚΕΤΑΙ Ο ΑΠΙΝΙΔΩΤΗΣ ΚΑΙ ΕΠΕΙΤΑ ΣΤΟ ΠΕΡΙΣΤΑΤΙΚΟ ΚΑΘΩΣ ΚΑΙ Η ΑΠΟΣΤΑΣΗ ΜΕΤΡΗΜΕΝΗ ΣΤΗΝ ΚΛΙΜΑΚΑ ΤΟΥ ΧΡΟΝΟΥ. ....	94
ΕΙΚΟΝΑ 68: ΚΩΔΙΚΑΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ “REACT-ROUTER-DOM” .....	95
ΕΙΚΟΝΑ 69: ΚΩΔΙΚΑΣ ΑΡΧΕΙΟΥ STORE ΤΗΣ REDUX .....	96
ΕΙΚΟΝΑ 70: ΚΩΔΙΚΑΣ ΣΥΝΑΡΤΗΣΗΣ GEOCODEBYADDRESS .....	97
ΕΙΚΟΝΑ 71: ΚΩΔΙΚΑΣ ΒΙΒΛΙΟΘΗΚΗΣ PUSHER.....	97
ΕΙΚΟΝΑ 72: ΚΩΔΙΚΑΣ ΣΥΝΑΡΤΗΣΗΣ GENERATE-PASSWORD .....	98
ΕΙΚΟΝΑ 73: ΚΩΔΙΚΑΣ ΒΙΒΛΙΟΘΗΚΗΣ GENERATE-PASSWORD .....	98
ΕΙΚΟΝΑ 74: ΚΩΔΙΚΑΣ ΣΥΝΑΡΤΗΣΗΣ HANDLEGETDIRECTIONS.....	99
ΕΙΚΟΝΑ 75: ΚΩΔΙΚΑΣ ΣΥΝΑΡΤΗΣΗΣ GETCURRENTPOSITIONASYNC .....	100
ΕΙΚΟΝΑ 76: ΚΩΔΙΚΑΣ ΣΥΝΑΡΤΗΣΗΣ GETPATHLENGTH .....	100
ΕΙΚΟΝΑ 77: ΚΩΔΙΚΑΣ JWTAUTHENTICATION .....	101
ΕΙΚΟΝΑ 78: ΚΩΔΙΚΑΣ ΜΕ ΤΟ ΣΥΝΟΛΟ VALIDATION ΚΑΝΩΝΩΝ .....	102
ΕΙΚΟΝΑ 79: ΚΩΔΙΚΑΣ ΜΕ ΧΡΗΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ EXPO->NOTIFY .....	102
ΕΙΚΟΝΑ 80: ΚΩΔΙΚΑΣ ΜΕ ΧΡΗΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ PUSHER->TRIGGER .....	103



## Κατάλογος πινάκων

ΠΙΝΑΚΑΣ 1: ΠΑΡΟΥΣΙΑΣΗ ΒΑΣΙΚΩΝ ΣΤΟΙΧΕΙΩΝ HTML .....	30
ΠΙΝΑΚΑΣ 2: ΠΙΝΑΚΑΣ “ΕΘΕΛΟΝΤΗ” .....	54
ΠΙΝΑΚΑΣ 3: ΠΙΝΑΚΑΣ “ΕΚΑΒ” .....	56
ΠΙΝΑΚΑΣ 4: ΠΙΝΑΚΑΣ “ΠΕΡΙΣΤΑΤΙΚΟ” .....	57
ΠΙΝΑΚΑΣ 5: ΠΙΝΑΚΑΣ “ΑΠΙΝΙΔΩΤΗΣ” .....	58
ΠΙΝΑΚΑΣ 6: ΠΙΝΑΚΑΣ “ΑΣΘΕΝΗΣ” .....	60
ΠΙΝΑΚΑΣ 7: ΜΕΤΡΙΚΕΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ .....	106
ΠΙΝΑΚΑΣ 8: ΜΕΤΡΙΚΕΣ ΚΙΝΗΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	106
ΠΙΝΑΚΑΣ 9: ΜΕΤΡΙΚΕΣ BACKEND .....	106



## Περίληψη

Στην σημερινή εποχή, βιώνουμε μια ταχύτατη ανάπτυξη της τεχνολογίας σε αρκετούς τομείς της πληροφορικής και των τηλεπικοινωνιών. Με την συνεχή βελτίωση του διαδικτύου και των υπηρεσιών που παρέχει, ερχόμαστε σε θέση να τις χρησιμοποιήσουμε προς όφελος μας για να κάνει τις ζωές μας πιο εύκολες ή ακόμα και να σώσουν μια ζωή. Στόχος της παρούσας διπλωματικής είναι να βοηθήσει στην αντιμετώπιση ενός περιστατικού, ιδιαίτερα σε περιοχές στις οποίες τίθεται δύσκολο το να σπεύσει κάποιο ασθενοφόρο. Με αυτό τον τρόπο το πληροφοριακό σύστημα διαχείρισης θα συντελέσει στην προστασία των ασθενών, θα παρέχει επίσης αίσθημα ασφάλειας προς τους πολίτες, ενδυναμώνοντας την αλληλεγγύη και την προσφορά στο κοινωνικό σύνολο, που είναι τόσο σημαντικές σε μια κοινωνία.

Στην παρούσα διπλωματική εργασία πρόκειται να σχεδιαστεί και να υλοποιηθεί ένα πληροφοριακό σύστημα διαχείρισης απινιδωτών. Το πληροφοριακό σύστημα θα αποτελείται από μια διαδικτυακή εφαρμογή και από 2 εφαρμογές σε iOS και Android αντίστοιχα. Ο εκάστοτε χρήστης θα έχει τη δυνατότητα να κάνει αίτηση στην αρμόδια υπηρεσία (Περιφέρεια/ΕΚΑΒ) για να εκπαιδευτεί με μια σειρά σεμιναρίων ως εθελοντής. Αφού τα σεμινάρια έχουν έρθει εις πέρας επιτυχώς τότε θα του δίνεται ένα username και ένα password για να μπορεί να έχει πρόσβαση στην εφαρμογή μας (είτε στην ιστοσελίδα είτε από κινητό). Στην περίπτωση περιστατικού, αφού κάποιος έχει ενημερώσει το ΕΚΑΒ και ο εκάστοτε ανταποκριτής με τη σειρά του αφού ενημερώσει το πλήρωμα του ασθενοφόρου ενημερώνει και τη σελίδα. Επί τόπου, θα έρχεται push notification στους χρήστες που βρίσκονται σε έναν απινιδωτή κοντά στο περιστατικό. Αφού ο χρήστης ανταποκριθεί στο push notification θα εισέλθει στην εφαρμογή και πιο συγκεκριμένα σε έναν χάρτη, ο οποίος θα τον οδηγεί στον πιο κοντινό απινιδωτή με την βέλτιστη διαδρομή. Επιλέγοντας κάποιο συμβάν, ο χρήστης έχει τη δυνατότητα να λάβει παραπάνω πληροφορίες για αυτό. Τέλος ο διαχειριστής, θα έχει τη δυνατότητα να διαγράφει συγκεκριμένα συμβάντα ή και χρήστες, σε περίπτωση που διαπιστωθεί κακόβουλη χρήση.

**Περιβάλλον Ανάπτυξης Κινητής Εφαρμογής:** Το ενσωματωμένο σύστημα υλοποιήθηκε, με χρήση τεχνολογιών λογισμικού και προγραμματισμού διαδικτύου ανοικτού κώδικα κυρίως Javascript, και πιο συγκεκριμένα το ανερχόμενο framework της React Native. Επίσης η εφαρμογή δοκιμάστηκε σε πραγματικές συσκευές με την βοήθεια του προσομοιωτή Expo CLI.

**Περιβάλλον Ανάπτυξης Διαδικτυακής Εφαρμογής:** Η διαδικτυακή εφαρμογή υλοποιήθηκε, με χρήση τεχνολογιών λογισμικού και προγραμματισμού διαδικτύου ανοικτού κώδικα (html5, css3, php, Slim, material-ui, JavaScript (ES2016, ES2017, ES2018), jQuery, Axios, Reactjs, React-router, Redux, Redux-thunk, Pusher API, Mysql, Webpack, Babel), και αναπτύχθηκε μέσω των παρακάτω κειμενογράφων (Visual Studio Code).

**Λέξεις κλειδιά:** Ενσωματωμένο Σύστημα, Διαδικτυακή Εφαρμογή, Διεπαφή προγραμματισμού εφαρμογών (API), ΕΚΑΒ, Εθελοντισμός, Πραγματικός Χρόνος, Απινιδωτής.

## Abstract

In today's times, we are experiencing the rapid development of technology in a number of IT and telecommunications sectors. With the continuous improvement of the internet and the services we offer, we are able to use them for our benefit in making our lives easier or even saving a life. The aim of this diploma is to help deal with an incident, especially in areas where it is difficult to hurry an ambulance quickly. In this way, the IT management system will help to protect patients, it will also provide a sense of security for citizens, enhancing solidarity and offering to the community that are so important in a society.

In this diploma thesis, an information defibrillator management system will be designed and implemented. The information system will consist of a web application and two applications on iOS and Android respectively. The individual user will be able to apply to the relevant department (Region Authority / EKAB) to train with a series of seminars as a volunteer. Once the seminars have been successfully completed, they will be given a username and a password to access our application (either on the website or on a mobile). In the event of an incident, once someone has informed the EKAB and the correspondent in turn after informing the ambulance crew, it also updates the page. On the spot, there will be push notification to users who are in a defibrillator near the incident. Once the user has responded to the push notification he will enter the application and more specifically a map that will lead him to the nearest defibrillator with the optimal route. Once the user has responded to the push notification he will enter the application and more specifically a map that will lead him to the nearest defibrillator with the optimal route. By selecting an event, the user can get more information about it. Finally, the administrator will be able to delete specific events or users if malicious use is detected.

Mobile Embedded Development Environment: The embedded system was implemented using Javascript open source software and programming technologies, and in particular the emerging React Native framework. Also, the application was tested on real devices using the Expo CLI simulator.

Web Application Development Environment: The web application was implemented using open source software and programming technologies (html5, css3, php, slim, material, JavaScript) (ES2016, ES2017, ES2018), jQuery, Axios, Reactjs, React-router , Redux, Redux-thunk, Pusher API, Mysql, Webpack, Babel) and was developed through the following Visual Studio Code.

Keywords: Embedded System, Web Application, Application Programming Interface (API), EMAW, Volunteering, Real Time, Defibrillator.

## Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο

“Σχεδίαση και υλοποίηση πληροφοριακού συστήματος διαχείρισης εθελοντών και απινιδωτών”

καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μηνά Δασυγένη, αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright © Μυρωνίδης Αλέξανδρος, Μηνάς Δασυγένης, 2019, Κοζάνη

## Ευχαριστίες

Με την ευκαιρία αυτής της διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω αρχικά την οικογένεια μου που πίστεψε σε εμένα και μου συμπαραστάθηκε υλικά και ηθικά στις επιλογές μου.

Ακόμη θα ήθελα να ευχαριστήσω θερμά όλους τους φίλους και συμφοιτητές μου, για την αμοιβαία βοήθεια όλα αυτά τα χρόνια σπουδών.

Τέλος θα ήθελα να ευχαριστήσω ιδιαίτερα τον κύριο Μηνά Δασυγένη, επιβλέπων της εργασίας και καθηγητή μου, για την εμπιστοσύνη που έδειξε στο πρόσωπο μου με την ανάθεση της παρούσας διπλωματικής εργασίας καθώς και για τη βοήθεια και καθοδήγηση που πρόσφερε όπου και όταν αυτή χρειάστηκε.



## Κεφάλαιο 1ο – Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζεται αναλυτικά το αντικείμενο της διπλωματικής εργασίας. Αρχικά γίνεται μία ανάλυση της σημερινής κατάστασης και των συνθηκών που προκύπτουν, με την χρήση της τεχνητής νοημοσύνης, αναφέρονται επίσης η ιδέα παράλληλα με τον σκοπό, που οδήγησαν στην ανάληψη και υλοποίηση της παρούσης διπλωματικής εργασίας. Έπειτα αναφέρονται, αντίστοιχες εφαρμογές από άλλους ερευνητές, μαζί με το τελικό συμπέρασμα που προσκομίσθηκε, αναλύοντας και συγκρίνοντας ξεχωριστά κάθε εφαρμογή και τέλος, γίνεται μία συνοπτική παρουσίαση της δομής της εργασίας.

### 1.1 Περιγραφή σημερινής κατάστασης

Στον, εν εξελίξει, 21ο αιώνα η ανθρωπότητα βιώνει μία τεράστια ανάπτυξη της τεχνολογίας, η οποία τείνει να διεισδύσει σε κάθε κλάδο που συσχετίζεται με τον άνθρωπο. Από την υγεία, την εργασία και την μετακίνηση, μέχρι την ψυχαγωγία, την επικοινωνία και την εκπαίδευση, αναπτύσσονται καθημερινά, με την βοήθεια της τεχνολογίας, ολοένα και περισσότερα εργαλεία και υπηρεσίες, που αλληλοεπιδρούν με τον άνθρωπο και την καθημερινότητα του, συμβάλλουν στην βελτίωση της ποιότητας ζωής και γενικότερα διευκολύνουν, καταστάσεις και υποχρεώσεις που βιώνει καθημερινά. Ένας από τους κλάδους που η τεχνολογία έχει κάνει ραγδαία βήματα εξέλιξης, ειδικότερα τα τελευταία χρόνια, είναι ο τομέας της υγείας. Λόγω των αυξανόμενων υποχρεώσεων (επαγγελματικών και μη) του ανθρώπου, υφίσταται πλέον η ανάγκη, για δημιουργία συσκευών και προγραμμάτων, τα οποία θα είναι δομημένα και προγραμματισμένα με τέτοιο τρόπο ώστε με τη σωστή διαχείρισή τους από τον άνθρωπο να είναι ικανά να τον βοηθήσουν, να επικοινωνούν μαζί του με την βοήθεια του διαδικτύου, εξυπηρετώντας του καθημερινά, διάφορες ανάγκες και υποχρεώσεις, συμβάλλοντας έτσι στην βελτίωση της ποιότητας ζωής του, καθώς και στην αποφόρτιση του προγράμματός του, δίνοντας του πολύτιμο χρόνο για διάφορες άλλες ασχολίες.

Ειδικότερα τα τελευταία χρόνια έχει δημιουργηθεί ο εξής όρος: “Διαδίκτυο των πραγμάτων” (Internet of Things ή IoT) που αναφέρεται για την περιγραφή τέτοιων συσκευών, με παρόμοιες ή και ίδιες λειτουργίες με αυτές που προαναφέρθηκαν. Πιο συγκεκριμένα, ο όρος χαρακτηρίζει ουσιαστικά, τις συσκευές που έχουν την δυνατότητα να συνδέονται στο διαδίκτυο και παράλληλα, να είναι προγραμματισμένες σε τέτοιο βαθμό, που να εκτελούν διάφορες λειτουργίες, σύμφωνα με τον λόγο ύπαρξής τους, χωρίς την παρέμβαση του

ανθρώπου. Συνεπώς, δημιουργείται το συμπέρασμα, ότι μπορεί οποιοσδήποτε να ελέγχει τις συσκευές απομακρυσμένα και παράλληλα, να λαμβάνει τις πληροφορίες που η κάθε συσκευή αποστέλλει, ως ανάδραση (feedback) ή ακόμα και να παρακολουθεί σε πραγματικό χρόνο (Real Time) την λειτουργία της κάθε συσκευής.

## 1.2 Ιδέα και σκοπός υλοποίησης

Στην σημερινή εποχή, βιώνουμε μια ταχύτατη ανάπτυξη της τεχνολογίας σε αρκετούς τομείς της πληροφορικής και των τηλεπικοινωνιών. Με την συνεχή βελτίωση του διαδικτύου και των υπηρεσιών που παρέχει, ερχόμαστε σε θέση να τις χρησιμοποιήσουμε προς όφελος μας για να κάνει τις ζωές μας πιο εύκολες ή ακόμα και να σώσουν μια ζωή. Λαμβάνοντας υπόψη, όλα τα παραπάνω και συνδυάζοντας τις πληροφορίες από αντίστοιχες εφαρμογές στο εμπόριο, προέκυψε η ιδέα για την δημιουργία μίας εφαρμογής κινητού τηλεφώνου με σκοπό την άμεση ανταπόκριση εξειδικευμένων εθελοντών σε περιπτώσεις καρδιακών περιστατικών. Η καρδιακή ανακοπή αποτελεί σίγουρα επείγουσα κατάσταση, απειλητική για τη ζωή και κάθε λεπτό έχει σημασία. Η γρήγορη δράση μπορεί να σώσει ζωές για αυτό τον λόγο απαιτεί άμεση βοήθεια, θεωρητικές και πρακτικές γνώσεις, μεθοδικότητα και συντονισμό. Παρά τις προσπάθειες που γίνονται παγκοσμίως για εκπαίδευση με σεμινάρια καρδιοπνευμονικής αναζωογόνησης, τα ποσοστά επιβίωσης δεν είναι ενθαρρυντικά (Περίπου το 80% των καρδιακών ανακοπών – σχεδόν 288.000 άτομα - συμβαίνουν εκτός νοσοκομείου κάθε χρόνο, και λιγότερο από το 10% επιβιώνουν, σύμφωνα με το American Heart Association).

Ο σκοπός της ιδέας όλου αυτού του εγχειρήματος και της υλοποίησής του, που είναι παράλληλα και το αντικείμενο της παρούσης διπλωματικής εργασίας, είναι η δημιουργία ενός ευφυούς συστήματος, υπεύθυνου για την ενημέρωση του χρήστη σε πραγματικό χρόνο, σε περίπτωση περιστατικού δίνοντας τον την δυνατότητα να επέμβει. Εφ' όσον ο εθελοντής δύναται να επέμβει, η εφαρμογή τον ανακατευθύνει στους χάρτες του κινητού του τηλεφώνου με καθοδήγηση αρχικά για τον πλησιέστερο απινιδωτή και μετ' έπειτα στη διεύθυνση του περιστατικού.

## 1.3 Αντίστοιχα συστήματα και συμπέρασμα υλοποίησης

Ύστερα από ενδελεχή αναζήτηση σε διαδικτυακούς τόπους εκπαιδευτικού και τεχνολογικού ενδιαφέροντος, παρακάτω, παρουσιάζεται μία λίστα από πανομοιότυπες εφαρμογές, τα οποία εξυπηρετούν τον σκοπό της παρούσης διπλωματικής εργασίας. Παρουσιάζονται αναλυτικά οι εφαρμογές καθώς και τι προσφέρει η κάθε μια από αυτές ξεχωριστά. Τέλος συγκρίνονται όλες οι εφαρμογές σε σχέση με την εφαρμογή που δημιουργήθηκε στην παρούσα διπλωματική καθώς και ποια είναι τα πλεονεκτήματα έναντι αυτών.

### 1.3.1 AED Locations v2



Εικόνα 1: Εικονίδιο εκτέλεσης αντίστοιχης εφαρμογής

Μία σχετική εφαρμογή είναι το AED Locations v2 [1] από την ομάδα Abletech. Το AED Locations v2 δίνει δυνατότητα στον χρήστη να προσθέσει κάποιον ιδιωτικό απινιδωτή καθώς και την τοποθεσία αυτού στον χάρτη με πρωταρχικό σκοπό και την ελπίδα να σωθούν ζωές στη καθ' εκάστοτε περιοχή. Αυτή η εφαρμογή δίνει επίσης την δυνατότητα στον χρήστη να έχει ανά πάσα στιγμή στο τηλέφωνο του τις τοποθεσίες απινιδωτών όπου και αν βρίσκεται. Πιο αναλυτικά η εφαρμογή προσφέρει:

- Εμφάνιση απινιδωτών δίπλα στην τωρινή τοποθεσία του χρήστη

- Εμφάνιση όλων των απινιδωτών στον χάρτη
- Αναζήτηση απινιδωτών με το όνομα ή τη διεύθυνση του
- Λήψη πληροφοριών της περιοχής, τρόπους επικοινωνίας και διαθεσιμότητα
- Εμφάνιση λίστας με το σύνολο όλων των απινιδωτών
- Εύκολη πληροφόρηση αναβαθμίσεων και προβλημάτων

### 1.3.2 Permis de sauver



Εικόνα 2: Εικονίδιο εκτέλεσης δεύτερης αντίστοιχης εφαρμογής

Μια άλλη εφαρμογή είναι η Permis de sauver [2] από την ομώνυμη ομάδα προγραμματιστών. Η εφαρμογή παρέχει τη δυνατότητα στον χρήστη να γίνει κρίκος της αλυσίδας διάδοσης ενός ασθενή καρδιακής ανακοπής. Προσφέρει την ευκαιρία για άμεση δράση παρέχοντας ταχύτερη υποστήριξη στο περιστατικό, αυξάνοντας έτσι τις πιθανότητες επιβίωσης.

Πιο αναλυτικά :

- Προσφέρει γρήγορη εγγραφή για όλους τους πολίτες εξειδικευμένους ή όχι
- Δυνατότητα ειδοποιήσεων για ταχύτερη υποστήριξη στο περιστατικό
- Δυνατότητα σε κάθε χρήστη να συμμετέχει στη συλλογή πληροφοριών σχετικά με τους πόρους έκτακτης ανάγκης (π.χ. θέση απινιδωτών)
- Παρέχει ειδοποιήσεις με την απόσταση που απέχει ο εθελοντής από το περιστατικό
- Δυνατότητα συνομιλίας σε πραγματικό χρόνο (live chat) μεταξύ των χρηστών
- Παρέχει πλήρη χαρτογράφηση όλων των απινιδωτών σε κοντινή απόσταση

### 1.3.3 AED Alert



Εικόνα 3: Εικονίδιο εκτέλεσης τρίτης αντίστοιχης εφαρμογής

Μια ακόμα εφαρμογή είναι η AED Alert [3] από την ομάδα προγραμματιστών Stan – Standby to help. Με την εφαρμογή AED Alert μπορείτε να συμμετέχετε σε οργανωμένη τοπική βοήθεια, η οποία μπορεί να μειώσει το ποσοστό θνησιμότητας από αιφνίδιες καρδιακές ανακοπές.

Λειτουργία της εφαρμογής :

Σε περίπτωση αιφνίδιας καρδιακής ανακοπής, οι εθελοντές που βρίσκονται κοντά στο περιστατικό λαμβάνουν ειδοποίηση. Ο αριθμός των εθελοντών που λαμβάνουν ειδοποίηση εξαρτάται από τη διαθεσιμότητα τους. Ειδοποιημένοι εθελοντές καλούνται να απαντήσουν στην ειδοποίηση. Με αυτό τον τρόπο παρακολουθούμε τον αριθμό των εθελοντών που θα πάνε στην κλήση. Όταν ένας επαρκής αριθμός εθελοντών έχει αποδεχθεί την ειδοποίηση, οι άλλοι λαμβάνουν ένα μήνυμα ακύρωσης. Με αυτόν τον τρόπο δεν υπάρχουν ποτέ πολλοί εθελοντές στη σκηνή. Οι έτοιμοι εθελοντές λαμβάνουν την τοποθεσία του συμβάντος ή του απινιδωτή, συμπεριλαμβανομένης της περιγραφής της διαδρομής. Οι εθελοντές που ανακτούν τον

απινιδωτή λαμβάνουν έναν κωδικό PIN ή χρησιμοποιούν ένα κλειδί για να ανοίξουν τον πίνακα του απινιδωτή.

Άλλα χαρακτηριστικά εφαρμογής

- Χρονοδιάγραμμα όπου μπορείτε να δει τις πρόσφατες κλήσεις του ο χρήστης
- δελτίο ταυτότητας (αναγνώριση στο περιστατικό)
- Εύρεση πλησιέστερων απινιδωτών
- Λίστα επιτευγμάτων στην περιοχή δράσης του χρήστη της εφαρμογής

### 1.3.4 Staying Alive



Εικόνα 4: Εικονίδιο εκτέλεσης τέταρτης αντίστοιχης εφαρμογής

Επίσης μια ακόμη εφαρμογή είναι η Staying Alive [4] από την ομόθυμη προγραμματιστών Association RMC-BFM & AEDMAP. Η συγκεκριμένη εφαρμογή εκτός την χαρτογράφηση των απινιδωτών διαθέτει επίσης και χαρτογράφηση όλων των ανταποκριτών που είναι ενεργοί. Άλλα κύρια χαρακτηριστικά της συγκεκριμένης εφαρμογής είναι:

- Τοποθεσία απινιδωτών : Εντοπισμός των δέκα πλησιέστερων απινιδωτών στη τωρινή τοποθεσία του χρήστη, χρησιμοποιώντας μια βάση δεδομένων η οποία ενημερώνεται κάθε φορά που γίνεται έναρξη της εφαρμογής
- Αναφορά νέων απινιδωτών. Ο χρήστης έχει τη δυνατότητα να εισάγει νέους απινιδωτές και επιπλέον να επεξεργαστεί τους ήδη υπάρχοντες
- Η εφαρμογή παρέχει ένα δίκτυο ανταποκριτών. Εάν κάποιος χρήστης είναι εκπαιδευμένος στις πρώτες βοήθειες, μπορεί να εγγραφεί ως ανταποκριτής και στη συνέχεια να ενημερώνει το σύστημα σε περίπτωση περιστατικού ώστε να παράσχει τις πρώτες βοήθειες. Η συγκεκριμένη λειτουργία είναι διαθέσιμη σε Γαλλία, Ελβετία, Βέλγιο
- Η εφαρμογή παρέχει επίσης έναν πρακτικό οδηγό χρήσης του αυτόματου εξωτερικού απινιδωτή
- Μια κατάσταση “έκτακτης ανάγκης” που βοηθάει τον χρήστη να κάνει τα σωστή διαδικασία με τη σωστή σειρά

### 1.3.5 DAEdove



Εικόνα 5: Εικονίδιο εκτέλεσης πέμπτης αντίστοιχης εφαρμογής

Μια ακόμη εφαρμογή η οποία μπορεί να παρομοιαστεί με την εφαρμογή της παρούσης διπλωματικής είναι η DAEdove [5] η οποία υλοποιήθηκε από τον Fabio Paracchini. Η DAEdove είναι μια δωρεάν εφαρμογή που δημιουργήθηκε για να παρέχει ένα χάρτη

αυτοματοποιημένων εξωτερικών απινιδωτών στο εθνικό έδαφος. Κάθε απινιδωτής σημειώνεται στο χάρτη υποστηρίζεται από πληροφορίες όπως η θέση του και η διεύθυνση του, τον ιδιοκτήτη του τηλεφωνικού αριθμού, καθώς και τη διαθεσιμότητα των προγραμμάτων εξοπλισμού. Η εφαρμογή δραστηριοποιείται στην Ιταλία.

Άλλα στοιχεία της εφαρμογής είναι:

- Ένας πλήρης ενημερωμένος κατάλογος όλων των πλησιέστερων απινιδωτών
- Πλήρη στοιχεία για κάθε έναν από αυτούς (διεύθυνση, αριθμό τηλεφώνου καθώς και ποιες είναι οι ώρες λειτουργίας του
- Δυνατότητα στον χρήστη για προσθήκη νέου απινιδωτή

### 1.3.6 Responders



Εικόνα 6: Ενδεικτική εικόνα έκτης αντίστοιχης εφαρμογής

Η εφαρμογή Responders [6] είναι επίσης μια εφαρμογή ανάλογου περιεχομένου και δημιουργήθηκε από την AEDMAP. Η εφαρμογή Responders λοιπόν με τη σειρά της χαρτογραφεί απινιδωτές καθώς και τα χαρακτηριστικά αυτών. Άλλα κύρια χαρακτηριστικά της εφαρμογής είναι:

- Εντοπισμός πλησιέστερου απινιδωτή σε περίπτωση έκτακτης ανάγκης
- Εγγραφή νέου χρήστη που μπορεί να συμβάλει στην ανάπτυξη της βάσης δεδομένων των απινιδωτών με την προσθήκη νέων



- Εκμάθηση πρώτων βοηθειών που απεικονίζονται με βίντεο παραδείγματα
- Ύστερα από επιτυχημένη εκμάθηση πρώτων βοηθειών δίνεται η δυνατότητα στο χρήστη να γίνει ανταποκριτής και να σπεύσει σε περίπτωση έκτακτης ανάγκης
- Εγχειρίδιο σχετικά με τα μέτρα που πρέπει να ληφθούν σε ένα έμφραγμα καθώς και τα προειδοποιητικά σημεία και συμπτώματα

### 1.3.7 SAUV Life



Εικόνα 7: Ενδεικτική εικόνα έβδομης αντίστοιχης εφαρμογής

Τέλος μια ακόμη αντίστοιχη εφαρμογή είναι η SAUV Life [7] η οποία δραστηριοποιείται στη Γαλλία. Η SAUV Life έχει ως στόχο να επαγρυπνήσει τους πολίτες να συμμετάσχουν στην διαδικασία της ανταπόκρισης σε ένα καρδιακό περιστατικό. Ένα από τα σημεία που κάνουν την εφαρμογή ιδιαίτερη είναι η αλληλεπίδραση μεταξύ των πολιτών και του κέντρου πρώτων βοηθειών. Η εφαρμογή επιτρέπει τους χρήστες της να επικοινωνήσουν με το κέντρο βοηθειών άμεσα και το ενημερώνουν με τη τοποθεσία τους με σκοπό να παρέμβουν στο περιστατικό αν αυτό βρίσκεται στην ακτίνα δράσης τους.

## 1.4 Ανάλυση και συμπεράσματα αντίστοιχων εφαρμογών

Αφού μελετήθηκαν προσεκτικά όλες οι εφαρμογές, οι δυνατότητες που παρέχουν και υπηρεσίες που χρησιμοποιήθηκαν, παρατηρήθηκαν τα εξής συμπεράσματα:

- Οι περισσότερες από τις εφαρμογές που μελετήθηκαν έχουν δράση σε συγκεκριμένα κράτη, όπως για παράδειγμα η AED Locations τη Νέα Ζηλανδία, η Permis De sauver τη Γαλλία όπως και η SAUV Life, η Staying Alive τη Γαλλία Ελβετία και τη Βέλγιο και η DAEdove την Ιταλία. Η Responders έχει διεθνή δράση παρ' όλα αυτά δεν δίνει τη δυνατότητα στον χρήστη να ενημερωθεί σε πραγματικό χρόνο για κάποιο περιστατικό στην ακτίνα δράσης του
- Όλες οι εφαρμογές δίνουν την δυνατότητα στον χρήστη να δημιουργήσει νέο λογαριασμό
- Κάποιες από αυτές όπως AED Locations, η Staying Alive και η DAEdove δίνουν επίσης τη δυνατότητα στον χρήστη της εφαρμογής να προσθέσει νέον ιδιωτικό απινιδωτή
- Η μόνη εφαρμογή που παρέχει άμεση επικοινωνία με το Κέντρο Άμεσης Βοήθειας είναι η SAUV Life
- Η μόνη εφαρμογή που παρέχει άμεση επικοινωνία μεταξύ των χρηστών είναι η Permis de sauver
- Τέλος κάποιες από αυτές τις εφαρμογές παρέχουν την δυνατότητα εκμάθησης πρώτων βοηθειών καθώς παρέχουν σχετικό ενσωματωμένο εγχειρίδιο. Αυτές είναι η εφαρμογή Staying Alive και η εφαρμογή Responders

Έχοντας συλλέξει τις άνωθεν πληροφορίες σχετικά με παρόμοιες εφαρμογές συμπεραίνουμε ότι το πληροφοριακό σύστημα που αναπτύχθηκε στην παρούσα διπλωματική είναι ικανό να συγκριθεί με τα παραπάνω λογισμικά καθώς καλύπτει κατά μεγάλο ποσοστό έναν σύνολο όλων των δυνατοτήτων τους. Αρχικά το HeartBit App δίνει και αυτό τη δυνατότητα στο χρήστη για άμεση επικοινωνία με το ΕΚΑΒ καθώς και να λαμβάνει σε πραγματικό χρόνο ειδοποιήσεις από αυτό σε περιπτώσεις καρδιακών περιστατικών. Ακόμα ένα πλεονέκτημα είναι ότι η εφαρμογή που αναπτύχθηκε και μπορεί να λειτουργήσει στην Ελλάδα είναι ικανή να εξατομικευτεί ανά περιφέρεια. Επίσης δίνεται η δυνατότητα στο χρήστη να θέσει τον εαυτό του ανενεργό σε περίπτωση απουσίας ή μη διαθεσιμότητας. Τέλος το HeartBit App στοχεύει όλο το φάσμα ηλικιών καθώς έχει φιλική διεπαφή χρήστη (UI friendly), καθώς και αποκριτική σχεδίαση (responsive design) ώστε να φαίνεται σε κάθε είδος συσκευής, οποιωνδήποτε διαστάσεων, με την ίδια ευκρίνεια και απόκριση.

## 1.5 Σύνοψη διπλωματικής εργασίας

Στο κείμενο αυτό αναλύεται η διαδικασία σχεδίασης και υλοποίησης του ενσωματωμένου συστήματος καθώς και της διαδικτυακής εφαρμογής. Αποτελείται από πέντε κεφάλαια.

Το τρέχον κεφάλαιο αποτελεί την εισαγωγή στο θέμα το οποίο πραγματεύεται η διπλωματική εργασία καθώς και τον λόγο για τον οποίο προέκυψε.

Στο δεύτερο κεφάλαιο γίνεται πλήρη καταγραφή και ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν, κατά την διάρκεια της κατασκευής του ενσωματωμένου συστήματος καθώς και της διαδικτυακής εφαρμογής, προκειμένου να προσαρμοστεί ο αναγνώστης στο γνωστικό αντικείμενο.

Το τρίτο κεφάλαιο πραγματεύεται τις απαιτήσεις του συστήματος για το σχεδιασμό της βάσης δεδομένων, την λεπτομερή περιγραφή των οντοτήτων, των δεδομένων και των συσχετίσεων που δημιουργήθηκαν. Επιπλέον, περιγράφονται οι τεχνικές που χρησιμοποιήθηκαν για την ασφάλεια του συστήματος.

Στο τέταρτο κεφάλαιο αναπτύσσεται η δημιουργία του λογισμικού μέρους της διαδικτυακής εφαρμογής. Παρουσιάζονται και αναλύονται οι λειτουργίες και η διεπαφή χρήστη μέσω διαφόρων στιγμιότυπων και πιθανών σεναρίων χρήσης του συστήματος.

Στο πέμπτο, και τελευταίο, κεφάλαιο παρουσιάζονται τα συμπεράσματα από την ανάπτυξη του συστήματος, τα προβλήματα που προέκυψαν κατά την υλοποίηση, και πιθανές μελλοντικές επεκτάσεις και βελτιώσεις



## Κεφάλαιο 2ο – Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο, παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας. Αποσαφηνίζονται θεωρητικές έννοιες που αφορούν τον προγραμματισμό διαδικτύου και αναλύονται οι πλατφόρμες ανάπτυξης λογισμικού που χρησιμοποιήθηκαν. Διότι, ο όγκος των τεχνολογιών, που χρησιμοποιήθηκαν στην διαδικτυακή εφαρμογή, είναι αρκετά μεγάλος και θα ήταν εξαιρετικά χρονοβόρο, να αναλυθούν όλες στον ίδιο βαθμό, θα αναλυθούν ιδιαίτερα οι τεχνολογίες που χρησιμοποιήθηκαν κατά κόρον και σε μικρότερο βαθμό, αυτές που χρησιμοποιήθηκαν ως υποστηρικτικές.

### 2.1 Προγραμματισμός διαδικτύου

Η επιστήμη της πληροφορικής που ασχολείται με τη δημιουργία διαδικτυακών εφαρμογών ιστοχώρου ονομάζεται Προγραμματισμός Διαδικτύου (Web Development) [8]. Η διαδικασία αυτή, της δημιουργίας ενός διαδικτυακού ιστοχώρου ή ιστοσελίδας (ισότιμοι όροι που περιγράφουν το ίδιο στοιχείο), μπορεί να κυμαίνεται από την ανάπτυξη μίας ενιαίας απλής σελίδας έως μία σειρά πολύπλοκων ιστοχώρων. Ο προγραμματισμός διαδικτύου περιλαμβάνει πολλές ενέργειες, μερικές από τις οποίες είναι ο σχεδιασμός του ιστοχώρου, η δημιουργία του περιεχομένου, ο προγραμματισμός, η διαμόρφωση της ασφάλειας του δικτύου, η ανάπτυξη σεναρίων για τον εξυπηρετητή (server) και τον πελάτη (client), καθώς και η δημιουργία συστημάτων για τη διαχείριση του περιεχομένου γνωστά ως Content Management Systems (CMS [33]), τα οποία αποτελούν τον ενδιάμεσο κρίκο μεταξύ του χρήστη και της βάσης δεδομένων.

Ο προγραμματισμός διαδικτύου μπορεί να διαιρεθεί στις εξής τρεις κατηγορίες: στον υπολογιστή εξυπηρετητή (web server), στον υπολογιστή πελάτη (web client) και τέλος στο σύστημα διαχείρισης βάσεων δεδομένων (Database Management System ή DBMS [32]). Ο όρος web server [9] αναφέρεται σε εκείνον τον υπολογιστή ο οποίος είναι υπεύθυνος για την διεκπεραίωση των αιτημάτων που του θέτουν οι web clients. Σε αυτόν τον υπολογιστή εκτελούνται γλώσσες προγραμματισμού εξυπηρετητή (server-side programming) για την παραγωγή HTML εγγράφων. Η έννοια web client αναφέρεται στον υπολογιστή του χρήστη στον οποίο συνήθως εκτελείται ένα πρόγραμμα περιήγησης στο διαδίκτυο (web browser). Ο

περιηγητής δημιουργεί και αποστέλλει αιτήματα προς τον web server και στην συνέχεια παραλαμβάνει τις απαντήσεις τις οποίες εν τέλει εμφανίζει στον χρήστη. Μία ακόμα αρμοδιότητα του browser είναι να εκτελεί γλώσσες προγραμματισμού πελάτη (client-side programming). Το σύστημα βάσης δεδομένων [10] θεωρείται μία ξεχωριστή κατηγορία παρόλο που μπορεί να εκτελείται και στον υπολογιστή εξυπηρετητή. Ναι μεν είναι αναπόσπαστο κομμάτι του προγραμματισμού διαδικτύου, αλλά παρόλα αυτά, είναι σύστημα γενικού σκοπού, το οποίο χρησιμοποιείται σε πολλές άλλες εφαρμογές οι οποίες δεν είναι απαραίτητα διαδικτυακές εφαρμογές. Τα συστήματα βάσεων δεδομένων είναι υπεύθυνα για την αποθήκευση και την ανάκτηση τις δυναμικής πληροφορίας που προέρχεται από τους χρήστες και από συσκευές καταγραφής δεδομένων. Χαρακτηριστικό παράδειγμα, που κάλλιστα θα μπορούσε να περιγράψει και την λειτουργία του συστήματος, παράλληλα με την διαδικτυακή εφαρμογή της διπλωματικής εργασίας, αποτελεί μία συσκευή, οι οποία λαμβάνει μέτρηση για την θερμοκρασία, ανά τακτά χρονικά διαστήματα και στέλνει στον server, την τρέχουσα τιμή, του χώρου στον οποίο βρίσκεται.

### 2.1.1 HTML

Η HTML (HyperText Markup Language ή Γλώσσα Σήμανσης Κειμένου) [11] είναι η βασική γλώσσα σήμανσης στο διαδίκτυο για την δημιουργία ενός ιστοχώρου. Η HTML δεν θεωρείται γλώσσα προγραμματισμού αλλά είναι ένα έγγραφο με οδηγίες προς τους browsers για το πως να εμφανίσουν το περιεχόμενο στους χρήστες. Όπως είναι φανερό η HTML “εκτελείται” στον υπολογιστή πελάτη μέσω του browser. Τα βασικά δομικά συστατικά της HTML είναι τα HTML στοιχεία (HTML elements) τα οποία αποτελούνται από ετικέτες (tags). Τα tags είναι λέξεις ή γράμματα που περικλείονται από γωνιώδεις αγκύλες και υπάρχουν ανά ζεύγη. Δηλαδή, υπάρχει ένα tag που σηματοδοτεί την έναρξη και ένα που σηματοδοτεί την λήξη με μερικές εξαιρέσεις. Το tag λήξης περιέχει μία πλάγια γραμμή ‘/’ για να διαχωρίζεται από το tag έναρξης. Ένα παράδειγμα tag, που είναι και ένα από τα πιο σημαντικά στοιχεία, για την δήλωση ενός HTML εγγράφου είναι το εξής: <html>...</html>. Τα tags μέσα τους περικλείουν κείμενο ή και άλλα εσωτερικά tags (εμφωλευμένα tags). Μερικά βασικά tags παραθέτονται στον πίνακα 1 που ακολουθεί.

HTML στοιχείο	Περιγραφή
<!DOCTYPE>	Αποτελεί οδηγία που καθορίζει την έκδοση της HTML που χρησιμοποιείται
<html>	Περικλείει όλο το HTML έγγραφο
<head>	Παρέχει πληροφορίες σχετικές με το έγγραφο όπως γλώσσα, κωδικοποίηση καθώς και μεταδεδομένα
<title>	Ορίζει τον τίτλο του ιστοχώρου
<body>	Περιγράφει το οπτικό περιεχόμενο της σελίδας, το οποίο θα εμφανιστεί στον χρήστη
<div>	Ομαδοποιεί στοιχεία μεταξύ τους και τα καθορίζει σαν ένα ξεχωριστό τμήμα της σελίδας. (Χρησιμοποιείται σχεδόν πάντα)
<!-- ... -->	Ορίζει σχόλια
<h1>,<h2>,...,<h6>	Ορίζει επικεφαλίδες τμημάτων κειμένου
<p>	Ορίζει παράγραφο
<button>	Ορίζει κουμπί
<form>	Ορίζει φόρμα
<input>	Ορίζει πεδίο εισαγωγής

Πίνακας 1: Παρουσίαση βασικών στοιχείων HTML

Από τις αρχές του διαδικτύου και των ιστοχώρων, υπάρχουν αρκετές εκδόσεις HTML οι οποίες μεταλλάσσονται στον χρόνο, με την πιο πρόσφατη έκδοση που χρησιμοποιείται μέχρι και σήμερα να είναι η HTML5. Η HTML γενικότερα, δημιουργεί τη βασική δομή ενός ιστοχώρου με αποτέλεσμα να υπάρχουν και άλλες τεχνολογίες που τον συμπληρώνουν διαμορφώνοντας έναν πιο ευπαρουσίαστο ιστοχώρο, όπως η CSS. Από την άλλη η Javascript βελτιώνει την λειτουργικότητα του, καθώς δίνει και την δυνατότητα αλληλεπίδρασης με τον χρήστη.

## 2.1.2 CSS

Η CSS (Cascading Style Sheets ή Διαδοχικά Φύλλα Στυλ) [12] χρησιμοποιείται για να περιγράψει τον τρόπο με τον οποίο το HTML έγγραφο θα εμφανιστεί στην οθόνη του χρήστη, σε οθόνες διαφορετικού μεγέθους αλλά και σε διάφορες άλλες περιπτώσεις. Η CSS η οποία δεν θεωρείται ούτε αυτή καθαρή γλώσσα προγραμματισμού, δημιουργήθηκε για να διαχωριστούν οι εντολές της εμφάνισης από τις εντολές του περιεχομένου μίας ιστοσελίδας. Η CSS βοηθάει στο να δημιουργηθεί μία ομορφότερη ιστοσελίδα, οπτικά. Δεν υπάρχει ιστοσελίδα στο διαδίκτυο η οποία να έχει υποστεί στιλιστική επεξεργασία δίχως την χρήση της CSS. “Εκτελείται”, επίσης, από τον browser του χρήστη. Το συντακτικό της είναι σχετικά απλό. Αποτελείται από μία λέξη ή ένα γράμμα τα οποία υποδηλώνουν ένα συγκεκριμένο στοιχείο της HTML και αυτό ονομάζεται ‘επιλογέας’ (css selector). Έπειτα, ακολουθούν μία

ή περισσότερες εντολές που περικλείονται μέσα σε αγκύλες τύπου αγκίστρου (curly braces). Κάθε εντολή βρίσκεται στην δική της γραμμή όπου και τερματίζεται με το ελληνικό ερωτηματικό. Τέλος, η κάθε εντολή αποτελείται από δύο μέρη, την ιδιότητα, η οποία δείχνει ποια ιδιότητα του HTML στοιχείου θα τροποποιηθεί και την τιμή, που θα λάβει η τρέχουσα ιδιότητα. Η ιδιότητα διαχωρίζεται από την τιμή με την άνω-κάτω τελεία. Σήμερα η έκδοση που χρησιμοποιείται είναι η CSS3. Στην παρακάτω εικόνα απεικονίζεται ένα παράδειγμα σύνταξης CSS.

```
background: yellow;
}
body {
width: 80%;
margin: 0 auto;
background: white;
}
header{
height: 50px;
margin-bottom: 1em;
```

Εικόνα 8: Παράδειγμα σύνταξης CSS

### 2.1.3 Javascript

Η Javascript (JS: συντομογραφία, ECMAScript: επίσημη ονομασία, VanillaJS: συνώνυμη ονομασία για Javascript) [13] είναι μία γλώσσα προγραμματισμού σεναρίου (scripting language) η οποία εκτελείται από τον browser στον υπολογιστή πελάτη. Ως scripting γλώσσα απαιτεί την ύπαρξη κάποιας μηχανής εκτέλεσης, κάτι το οποίο αναλαμβάνει να το κάνει ο browser. Πλέον όλοι οι σύγχρονοι browsers μπορούν να εκτελούν την JS χωρίς την ανάγκη κάποιου προσθέτου. Η JS χρησιμοποιείται για να βελτιωθεί η εμπειρία χρήσης και δίνει την δυνατότητα αλληλεπίδρασης με τον χρήστη. Μπορεί να ανανεώνει δυναμικά τα κομμάτια μίας ιστοσελίδας χωρίς να χρειάζεται η πλήρης ανανέωση της, γεγονός που απαιτεί περισσότερο χρόνο και είναι εμφανές στον χρήστη. Επίσης, η JS μπορεί να ζητήσει δεδομένα στο παρασκήνιο από τον web server χωρίς να διαγράψει το περιεχόμενο που κατά πάσα πιθανότητα ο χρήστης διαβάζει και μόλις αυτό είναι διαθέσιμο το εισάγει δυναμικά στην τρέχουσα ιστοσελίδα. Ουσιαστικά η JS έχει την δυνατότητα να αφαιρεί και να προσθέτει HTML στοιχεία, CSS κανόνες, να τροποποιεί συγκεκριμένες ιδιότητες των HTML στοιχείων μεταβάλλοντας τις τιμές τους. Με την πάροδο των χρόνων, έκαναν την εμφάνισή τους,



διάφορες εκδόσεις της JS. Σήμερα, η έκδοση που “καταλαβαίνουν” όλοι οι browsers είναι η ES5 (ECMAScript 5), η οποία εμφανίστηκε το 2009. Η τελευταία μεγάλη αναβάθμιση της γλώσσας έγινε το 2015 με την έλευση της ES6 (ECMAScript 2015). Από τότε μέχρι και σήμερα, ανά χρόνο, γίνεται μία ελαφριά αναβάθμιση της γλώσσας με καινούργιες λειτουργίες. Ωστόσο, από την τελευταία μεγάλη αναβάθμιση της γλώσσας μέχρι και σήμερα, έχει δημιουργηθεί ένα πρόβλημα όσον αφορά την συμβατότητα της γλώσσας με τους browsers και αξίζει να αναφερθεί. Όλοι οι browsers στην ουσία, αποτελούν λογισμικά, υλοποιημένα από διαφορετικές ομάδες προγραμματιστών και εταιρείες, που επιτρέπουν τον χρήστη να εισέλθει στο διαδίκτυο. Ωστόσο, καθώς η γλώσσα ανανεώνεται και προσθέτονται καινούργια στοιχεία, είναι ευθύνη της εκάστοτε εταιρείας να αναβαθμίσει και αυτή το λογισμικό της, ώστε να καταλαβαίνει τις αλλαγές και τις καινούργιες εντολές. Αυτή η διαδικασία σαφώς απαιτεί χρόνο και πραγματοποιείται ταχύτερα από κάποιες εταιρείες, πιο αργά από άλλες ή και καθόλου από ορισμένες. Το βάρος αυτομάτως πέφτει πάνω στα χέρια του προγραμματιστή, ο οποίος, στην δημιουργία ενός ιστοχώρου, θα πρέπει να λάβει υπόψη ότι μπορεί να υπάρχουν χρήστες, που εισέρχονται από διαφορετικούς browser. Βεβαίως αυτό μπορούσε να αποτελέσει ένα μεγάλο και πολλές φορές μοιραίο εμπόδιο διότι, θα πρέπει να συγγράψει κώδικα που να καταλαβαίνουν όλοι οι browsers, πράγμα ουσιαστικά αδύνατο. Την λύση στο πρόβλημα, δίνουν διάφοροι μεταγλωττιστές (compilers), που σαν αποστολή έχουν την μετατροπή της γλώσσας σε μία “κοινή διάλεκτο” που καταλαβαίνουν όλοι οι browsers. Ένας από τους πιο δημοφιλείς μεταγλωττιστές χρησιμοποιήθηκε και στην διαδικτυακή εφαρμογή της διπλωματικής εργασίας (αναλύεται παρακάτω). Τέλος, αξίζει να σημειωθεί ότι, ενώ η JavaScript ήταν καθαρά μία γλώσσα προγραμματισμού που εκτελούνταν αυστηρά στον client, πλέον έχει αρχίσει να χρησιμοποιείται και στην πλευρά των υπολογιστών εξυπηρετητών. Παράδειγμα κώδικα σε JS παρουσιάζει η εικόνα παρακάτω.

```
function addSubmitEvent(form, func) {  
  var oldSubmit = form.onsubmit;  
  if (typeof oldSubmit !== 'function') {  
    form.onsubmit = func;  
  }  
  else {  
    form.onsubmit = function() {  
      return oldSubmit() && func();  
    }  
  }  
}
```

Εικόνα 9: Παράδειγμα σύνταξης Javascript

### 1.1.1 2.1.4 MYSQL

Η MySQL [14] είναι ένα σύστημα το οποίο συγκαταλέγεται στην κατηγορία των σχεσιακών συστημάτων διαχείρισης βάσεων δεδομένων (RDBMS). Μία βάση δεδομένων αποτελεί τον πυρήνα κάθε συστήματος, μιας και εκεί αποθηκεύονται όλες οι πληροφορίες. Η βάση δεδομένων είναι αναπόσπαστο κομμάτι μίας εφαρμογής που έχει σαν αποστολή, να αποθηκεύει, να ανακτά και να εμφανίζει δεδομένα. Παρόλα αυτά συνιστά ξεχωριστή κατηγορία γιατί είναι πολύπλοκη, και επειδή σχετίζεται με άλλες εφαρμογές εκτός του προγραμματισμού διαδικτύου. Ένα DBMS είναι υπεύθυνο για την αποδοτική αποθήκευση των δεδομένων σε κάποιο μέσο μόνιμης αποθήκευσης με σκοπό την όσο το δυνατόν πιο γρήγορη ανάκτηση τους όταν αυτή ζητηθεί. Η επικοινωνία με μία βάση δεδομένων επιτυγχάνεται μέσω ενδιάμεσων προγραμμάτων, τους οδηγούς (drivers). Οι τελευταίοι λαμβάνουν μία συγκεκριμένη μορφή εντολών, την επεξεργάζονται κατάλληλα και την μεταφράζουν σε γλώσσα κατανοητή από την βάση δεδομένων. Στην συνέχεια την αποστέλλουν και έπειτα αναμένουν την πληροφορία την οποία μεταφέρουν στον αιτούντα. Οι εντολές αυτές ανήκουν στην γλώσσα SQL (Structured Query Language). Υπάρχουν δύο ειδών συστήματα διαχείρισης βάσεων δεδομένων, τα σχεσιακά και μη σχεσιακά. Η επιλογή του συστήματος έχει να κάνει με την πολυπλοκότητα της εφαρμογής που πρόκειται να χτιστεί, την πολυπλοκότητα στις σχέσεις μεταξύ των διάφορων οντοτήτων, καθώς και στον συνδυασμό των τεχνολογιών που πρόκειται να χρησιμοποιηθούν. Παράδειγμα εντολής SQL παρουσιάζει η εικόνα παρακάτω.

```
CREATE TABLE public."bridge.customeractions" AS
SELECT
  e."id",
  e."cust_application_id" AS "customer_object_id",
  e."user_id" AS "customer_consolidated_id",
  e."created_at" AS "activity_date",
  CASE
    WHEN e."code_action" = 'new' THEN 'Registered'
    WHEN e."code_action" = 'rejected' THEN 'Rejected'
    WHEN e."code_action" = 'canceled' THEN 'Cancelled'
    ELSE 'XNA'
  END AS "status",
  sys."customer_consolidated_id" AS "customerid"
FROM inbound."in.event" e
```

Εικόνα 10: Παράδειγμα SQL εντολής

## 2.1.5 PHP

Η PHP [15] είναι μία σεναριακή γλώσσα προγραμματισμού εξυπηρετητή (scripting server-side language), που είναι κατάλληλη για τη δημιουργία ιστοχώρων με δυναμικό περιεχόμενο. Πρόκειται ίσως για την πιο διαδεδομένη γλώσσα προγραμματισμού εξυπηρετητή, του κλάδου του προγραμματισμού διαδικτύου, λόγω της εύκολης και γρήγορης διαμόρφωσης που προσφέρει (configuration). Πάνω σε αυτή, έχει υλοποιηθεί το μεγαλύτερο ποσοστό ιστοχώρων στο διαδίκτυο, το οποίο κατέχει μέχρι και σήμερα και μάλιστα, την επέλεξαν ιστοχώροι οι οποίοι εξελίχθηκαν σε διαδικτυακούς κολοσσούς όπως είναι το Facebook, το Wikipedia, το Yahoo κ.α. Ένα αρχείο PHP μπορεί να περιέχει κείμενο, κώδικα HTML, CSS, JavaScript και PHP, με τον κώδικα να εκτελείται στον server και το αποτέλεσμα να επιστρέφει στον browser ως απλές σελίδες HTML. Πιο συγκεκριμένα, η PHP μπορεί να δημιουργήσει, να ανοίξει, να κλείσει, να διαβάσει, να γράψει και να διαγράψει αρχεία σε έναν server. Αποτελεί τον συνδετικό κρίκο μεταξύ του ιστοχώρου και της βάσης δεδομένων, αφού μόνο μέσω της PHP πραγματοποιούνται ενέργειες δημιουργίας, επεξεργασίας και διαγραφής δεδομένων σε αυτή. Ακόμη μπορεί να κρυπτογραφήσει δεδομένα. Η PHP, ως γλώσσα σεναριακού προγραμματισμού, ενσωματώνεται πολύ εύκολα στον κώδικα της HTML. Ο κώδικας της πρώτης πρέπει να βρίσκεται εντός της ετικέτας `<? php . . .?>` και τα αρχεία που περιέχουν κώδικα PHP και HTML πρέπει να έχουν υποχρεωτικά την κατάληξη .php. Η PHP τρέχει σε όλες τις πλατφόρμες, όπως των Windows, Linux, Unix, Mac OS X κ.λπ. Είναι συμβατή με όλους σχεδόν τους server που χρησιμοποιούνται σήμερα (Apache, IIS, Enginx κ.λπ.). Υποστηρίζει ένα ευρύ φάσμα από Βάσεις Δεδομένων, είναι δωρεάν και είναι εύκολη στην εκμάθηση. Επίσης, η PHP, ως γλώσσα δομημένου προγραμματισμού, είναι ιδιαίτερα επεκτάσιμη και παρέχει πληθώρα έτοιμων βιβλιοθηκών συναρτήσεων (Functions) προς χρήση, ενώ ταυτόχρονα δίνει τη δυνατότητα στον προγραμματιστή να δημιουργήσει και τις δικές του συναρτήσεις. Τέλος, η PHP επιτρέπει την χρήση αντικειμενοστραφούς προγραμματισμού (Object Oriented Programming ή OOP), που πλέον στην σημερινή εποχή χρησιμοποιείται σαν μοτίβο για την ανάπτυξη οποιασδήποτε εφαρμογής, σε οποιαδήποτε γλώσσα προγραμματισμού. Η σημερινή έκδοση που χρησιμοποιείται, είναι η PHP 7.2. Παράδειγμα κώδικα σε JS παρουσιάζει η εικόνα παρακάτω.

```

<?php
if (!isset($_POST['categorie']))
{
    include("formulaire_recherche.php");
}
// Pour une recherche sans mot_cle
elseif (!isset($_POST['mot_cle1']) AND !isset($_POST['mot_cle2']) AND !isset($_POST['mot_cle3']))
{
    $reponse = $bdd->prepare('
        SELECT * FROM document WHERE categorie = ?') or die(print_r($bdd->errorInfo()));

    $reponse->execute(array(
        $_POST['categorie']));
}

```

Εικόνα 11: Παράδειγμα σύνταξης PHP

## 2.2 Δομές (frameworks), βιβλιοθήκες (libraries), μεταγλωττιστές (compilers), εργαλεία

Στην υποενότητα που ακολουθεί, θα γίνει μία περιεκτική αναφορά στις βιβλιοθήκες (libraries), στις δομές (frameworks), στους μεταγλωττιστές (compilers) και σε διάφορα άλλα εργαλεία που χρησιμοποιήθηκαν στην υλοποίηση της διαδικτυακής εφαρμογής. Αποτελεί ένα πολύ σημαντικό κομμάτι αναφοράς της διπλωματικής διότι, ουσιαστικά, η υλοποίηση της εφαρμογής οφείλεται κατά κόρον σε αυτά τα στοιχεία. Το μεγαλύτερο ποσοστό, αποτελείται από τεχνολογίες “του σήμερα”, οι οποίες χρησιμοποιούνται για την δημιουργία σύγχρονων διαδικτυακών εφαρμογών και όχι μόνο.

### 2.2.1 Slim Framework

Το Slim [16] αποτελεί ένα δωρεάν, ανοικτού κώδικα, framework της PHP. Ως framework [17] ορίζεται μία δομή λογισμικού, υλοποιημένη σε μία συγκεκριμένη γλώσσα προγραμματισμού και περιλαμβάνει βιβλιοθήκες καθώς και βοηθητικά προγράμματα, με σκοπό ένας προγραμματιστής να γράφει πιο γρήγορα εφαρμογές. Μερικά γνωρίσματα των frameworks είναι:

- Ενσωμάτωση συχνών χρησιμοποιούμενων «ρουτίνων» (Functions), για αξιοποίηση χρόνου και αποφυγή διπλότυπου κώδικα.
- Προβλεπόμενη και πρότυπη ακολουθία λύσεων (συνήθως) σε συχνά προβλήματα όπως (σύνδεση χρηστών, έλεγχος δεδομένων φόρμας κλπ.).
- Έτοιμες λύσεις για ασφάλεια της εφαρμογής.

- Συχνή προτροπή για καλύτερη δομή της εφαρμογής με τελικό σκοπό ο προγραμματιστής να μπορεί τελικά να την διαχειριστεί ευκολότερα όσο μεγαλώνει.
- Αποτελεί μία πιο «καλογραμμένη» έκδοση της βασικής γλώσσας και απλοποιεί σημαντικά την ανάγνωση/κατανόηση της.

Το Slim, δημιουργήθηκε το 2013, από τον Josh Lockhart. Το Slim Framework είναι ένα PHP micro framework που βοηθά τους προγραμματιστές της PHP να γράφουν γρήγορα και εύκολα τις εφαρμογές web και τα API. Σκεφτείτε το ως ένα βασικό σύνολο εργαλείων με τα οποία ένας προγραμματιστής μπορεί να δημιουργήσει καταπληκτικά πράγματα.

Το Slim, ήταν η ο πρόδρομος της ανάπτυξης των micro framework που ξεκίνησε το 2013 και μέχρι και σήμερα αποτελεί καινοτόμα ιδέα. Ένα “micro” framework είναι μια συλλογή αναγκών που απαιτούνται για την συλλογή μιας διαδικτυακής εφαρμογής. Συνήθως, ένα micro framework αναλαμβάνει και διευκολύνει την λήψη ενός HTTP αιτήματος (request), δρομολογώντας το HTTP αίτημα στον κατάλληλο ελεγκτή (controller), ο οποίος είναι υπεύθυνος για να επιστρέψει μια HTTP απόκριση (response). Το Slim πηγαίνει αυτή τη διαδικασία κάποια βήματα μπροστά, αφού διαθέτει εργαλεία τα οποία διαχειρίζονται την HTTP απόκριση (caching HTTP, ενημερώσεις κατάστασης, ανακατευθύνσεις κ.ο.κ.) καθώς και μια εκτεταμένη εφαρμογή middleware.

Αντίθετα, πολλοί προγραμματιστές είναι περισσότερο εξοικειωμένοι με μεγαλύτερα full-stack frameworks όπως είναι το Laravel [18]. Αυτά τα εξαιρετικά πλαίσια έχουν μεγαλύτερο κώδικα καθώς και μεγαλύτερες “boiler-plate” λειτουργίες οι οποίες είναι πιο δυσνόητες στην εκμάθησή τους. Όμως υπάρχουν projects και εργασίες που δεν απαιτούν πρόσθετη επιβάρυνση καθώς διαχειρίζεται η πλευρά του πελάτη (client-side) από άλλα frameworks. Για αυτό ακριβώς το σκοπό δημιουργήθηκαν τα micro frameworks όπως το Slim.

Το Slim, προτιμά την απλότητα έναντι της πολυπλοκότητας. Αυτό ισχύει στο μέγιστο στην αρχιτεκτονική του Slim, τον πηγαίο του κώδικα, καθώς και στο σετ χαρακτηριστικών του.

## 2.2.2 Material-ui και responsive design

Το Material-ui [19] είναι ένα framework HTML και CSS με επεκτάσεις Javascript και διευθετεί πολλά θέματα εμφάνισης για φόρμες, κουμπιά και άλλα. Το Material-ui περιέχει σύνολο κανόνων CSS για διάφορα HTML στοιχεία γνωστά ως Material design προσδίδοντας τους μία καλύτερη εμφάνιση. Το Material design (κωδικό όνομα: quantum paper) είναι ένα σύνολο σχεδιαστικών κανόνων (design language) που δημιουργήθηκε από τη Google και ανακοινώθηκε για πρώτη φορά στο συνέδριο Google I/O στις 25 Ιουνίου, 2014. Επεκτείνει το μοντέλο των “καρτών” (πρωτοεμφανίστηκε στο Google Now) που είναι σχεδιασμός, βασισμένος σε διάταξη πίνακα (grid layout), με κινούμενα σχέδια και μεταβάσεις που ανταποκρίνονται στις κινήσεις των χρηστών, χρήση padding και εφέ βάθους με κατάλληλο φωτισμό και σκίαση των αντικειμένων. Ο σχεδιαστής Matías Duarte εξήγησε ότι “σε αντίθεση με το πραγματικό χαρτί, το ψηφιακό υλικό μπορεί να επεκταθεί και να μορφοποιηθεί έξυπνα. Το material έχει φυσικές επιφάνειες και άκρα. Οι σκιές δίνουν νόημα σε αυτό που αγγίζεις”. Η Google υποστηρίζει ότι η καινούρια σχεδιαστική τους γλώσσα βασίζεται στο χαρτί και το μολύβι. Επίσης περιέχει βιβλιοθήκες Javascript για την δυναμική αλλαγή της εμφάνισης των ιστοσελίδων. Το Material-ui είναι μία πολύ καλή επιλογή για μία καινούρια εφαρμογή διαδικτύου διότι μπορεί να δώσει μία ωραία εμφάνιση στην ιστοσελίδα χωρίς να χρειάζεται η συγγραφή καθόλου CSS. Επίσης διαθέτει έτοιμα μοτίβα κανόνων παρουσίασης και εναλλαγής της διεπαφής χρήστη (UI), που ποικίλουν ανά μέγεθος, διαστάσεις οθόνης ή είδος συσκευής. Δεδομένο που από μόνο του προσδίδει μία “ασφάλεια” στο γεγονός ότι, σε οποιαδήποτε αλλαγή διαστάσεων της εφαρμογής, η ίδια θα παραμείνει το ίδιο ελκυστική προς τον χρήστη και παράλληλα λειτουργική (το λεγόμενο responsive design). Η σημερινή του έκδοση είναι η 3.9.3.

## 2.2.3 jQuery

Η jQuery [20] είναι μία βιβλιοθήκη (library) της Javascript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. Πιο συγκεκριμένα, παρέχει έτοιμες εντολές που πραγματοποιούν τις ίδιες λειτουργίες με την Javascript με την χρήση λιγότερου κώδικα. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη

επισκευσιμότητα. Είναι προαπαιτούμενο για την λειτουργία των Javascript πακέτων του Bootstrap. Η σημερινή του έκδοση είναι η 3.3.1. Παράδειγμα εντολής jQuery παρουσιάζει η εικόνα παρακάτω.

```
$(document).ready(function() {
    changeColor();
});

function changeColor() {
    var rndColors = ["#00FF00", "#CCCCCC", "#995499", "#FFFFFF", "#FF9900"];
    var selColor = Math.floor(Math.random() * rndColors.length);
    $('div').css("background-color", rndColors[selColor]);
    setTimeout(changeColor, 1000);
}
```

Εικόνα 12: Παράδειγμα εντολής jQuery

## 2.2.4 React

Η React [21] (ReactJS ή React.js) είναι μία ανοικτού κώδικα (open source) βιβλιοθήκη της Javascript για τη δημιουργία διεπαφών χρήστη (UI) όσον αφορά το μπροστινό κομμάτι αλληλεπίδρασης (frontend). Είναι το στρώμα προβολής (view layer) για εφαρμογές ιστού. Δημιουργήθηκε (2013), αναπτύσσεται και συντηρείται μέχρι και σήμερα, ως επί το πλείστον, από την ομάδα προγραμματιστών του Facebook. Πρόκειται για ένα από τα πιο σύγχρονα και τεχνολογικά εξελιγμένα εργαλεία για δημιουργία σύνθετων εφαρμογών, που δεν περιορίζονται μόνο στο διαδικτυακό κομμάτι. Η σαφής λειτουργία της είναι, η δυνατότητα που προσφέρει για ταυτόχρονη αλληλεπίδραση των δεδομένων με τον χρήστη (rendering), η οποία συμβαίνει ταχύτατα, παράλληλα με την δυνατότητα για το χτίσιμο html στοιχείων εντός Javascript. Ωστόσο, το σημαντικό στοιχείο είναι ότι, διαθέτει παράπλευρα πακέτα προς εγκατάσταση, που το καθένα από αυτά αποτελεί ξεχωριστή λειτουργία. Αν συνδυαστούν, δημιουργούν ένα τελειοποιημένο εργαλείο για την δημιουργία ακόμα και της πιο σύνθετης εφαρμογής. Ένα πολύ σημαντικό στοιχείο που προκύπτει είναι το εξής: δύναται η δημιουργία ενός διαφορετικού είδους εφαρμογής, η λεγόμενη εφαρμογή μίας σελίδας (Single Page Application ή SPA) [22], η οποία δεν αποτελεί τίποτα παραπάνω από ένα HTML έγγραφο, το οποίο περιέχει σαν εξάρτηση ένα Javascript έγγραφο, που περιέχει όλο τον απαραίτητο κώδικα. Αυτό πρακτικά οδηγεί στην δυνατότητα αποφόρτισης του εξυπηρετητή και στην μετατροπή του από ένα πολυσύνθετο σύστημα που ευθύνεται για πολλές λειτουργίες παράλληλα (αυθεντικοποίηση, φόρτωση html, ασφάλεια, αλληλεπίδραση με την βάση δεδομένων, κ.α.) σε ένα αρκετά πιο βασικό και πιο εύκολο στην δημιουργία του, το λεγόμενο Application Programming Interface ή API [23]. Ένας από τους λόγους που γίνεται αυτή η

μετάβαση είναι διότι, ο εξυπηρετητής, πλέον, έχει αρκετά λιγότερες υποχρεώσεις και παράλληλα μπορεί να εξυπηρετεί περισσότερα αιτήματα. Τέλος, αξίζει να αναφερθεί ότι υπάρχουν και βιβλιοθήκες τρίτων μερών (third party libraries) που συνεργάζονται ιδανικά με την React, όπως επίσης η ίδια συνεργάζεται ιδανικά με τις κυριότερες γλώσσες προγραμματισμού εξυπηρετητή καθώς και frameworks. Η σημερινή της έκδοση είναι η 16.6. Το βασικό στοιχείο της React είναι το λεγόμενο “component”, παράδειγμα του οποίου παρουσιάζει η φωτογραφία παρακάτω.

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

Εικόνα 13: Παράδειγμα React Component

### 2.2.5 Fetch

Το Fetch [24], αποτελεί ένα εργαλείο που πραγματοποιεί τα αιτήματα προς τον εξυπηρετητή, το οποίο διαθέτει κάποιες σημαντικές προ εγκατεστημένες λειτουργίες παράλληλα με ορισμένες, που περιέχονται στις καινούργιες εκδόσεις της Javascript. Αντίστοιχο εργαλείο με παρόμοια λειτουργία με το fetch είναι το ευρέως γνωστό Ajax (Asynchronous Javascript and XML) ωστόσο στην σύγκρισή τους, υστερεί σε ορισμένους τομείς. Παράδειγμα εντολής fetch παρουσιάζεται στην επόμενη φωτογραφία.



```

fetch(url, {
  //credentials: 'include',
  method: "POST", // *GET, POST, PUT, DELETE, etc.
  mode: "cors", // no-cors, cors, *same-origin
  cache: "no-cache", // *default, no-cache, reload, force-cache, only-
  // credentials: "same-origin", // include, same-origin, *omit
  headers: {
    // "Content-Type": "application/json; charset=utf-8"
    "Content-Type": "application/x-www-form-urlencoded"
  },
  redirect: "follow", // manual, *follow, error
  referrer: "no-referrer", // no-referrer, *client
  body: JSON.stringify(userData) // body data type must match "Content-
})
.then(response => {
  return response.json();
})
.then(responseJson => {
  // console.log(userData);
  resolve(responseJson);
})
.catch(error => {
  reject(error);
});

```

Εικόνα 14: Παράδειγμα εντολής fetch

## 2.2.6 Redux

Το Redux [25] είναι μία ανοικτού κώδικα (open source) βιβλιοθήκη της Javascript, που λειτουργεί ως διαχειριστής της κατάστασης της εφαρμογής (state management). Πρόκειται για μία τεχνολογία αρκετά σύγχρονη όπως η React, όμως αρκετά πολύπλοκη και δυσνόητη στην αρχή, όσο για τον λόγο ύπαρξής της, καθώς και για εκμάθηση και υλοποίηση, ωστόσο τρομερά σημαντική για την λειτουργία της εφαρμογής. Η λειτουργία της απαιτείται σε μεγάλου μεγέθους εφαρμογές. Στην ουσία μπορεί να παρομοιαστεί σαν ένα “καλάθι” ή σαν μία μικρού μεγέθους βάση δεδομένων, στην οποία επιλέγει να τοποθετήσει η εφαρμογή, πληροφορίες, που ενδέχεται να τις χρειαστεί άμεσα ή αρκετά συχνά. Δεν αντικαθιστά την κανονική βάση δεδομένων του συστήματος, απλώς συνυπάρχει μαζί της. Έτσι επιτυγχάνεται άμεση απόκριση της εφαρμογής, χωρίς την χρήση αιτημάτων προς τον εξυπηρετητή. Συνεργάζεται ιδανικά με την React και αντίστοιχα frontend frameworks και βιβλιοθήκες.

## 2.2.7 Webpack

Το Webpack [26], αποτελεί ένα εργαλείο το οποίο ενώνει πολλαπλά αρχεία (Javascript, CSS, κ.α.) σε ένα ή δύο κύρια αρχεία του ίδιου είδους (static module bundler). Είναι ίσως το κορυφαίο εργαλείο στο είδος του αυτή την στιγμή και τρομερά σημαντικό διότι, καταφέρνει και ελαττώνει σε πολύ μεγάλο βαθμό το μέγεθος των αρχείων που ουσιαστικά είναι η εφαρμογή, αποτέλεσμα με άμεση αντανάκλαση στην ταχύτητα και απόκριση της εφαρμογής. Έρχεται προ εγκατεστημένο πλέον με έτοιμες ρυθμίσεις (pre-build) στα περισσότερα frameworks (και στο Laravel) ή έτοιμα πακέτα για δημιουργία εφαρμογών, διότι ο αριθμός των βιβλιοθηκών που χρησιμοποιούνται είναι δυσανάλογα μεγάλος για το τελικό μέγεθος της εφαρμογής που απαιτείται.

## 2.2.8 Babel

Το Babel [27], αποτελεί έναν μεταγλωττιστή της Javascript (compiler), που μεταφράζει τις καινούργιες εκδόσεις της γλώσσας στην γνωστή σε όλους τους browsers ES5, ώστε η διαδικτυακή εφαρμογή να τρέχει το ίδιο ομαλά σε όλους τους browsers. Αναλύθηκε προηγουμένως στην ενότητα της Javascript, το πρόβλημα που επιλύει. Είναι το κορυφαίο εργαλείο στο είδος του, ακριβώς όπως και με το webpack, έρχεται προ εγκατεστημένο παντού και πιο συγκεκριμένα, υπάρχει έτοιμη η σύνδεση του με το webpack, ώστε ο προγραμματιστής να μην χρειαστεί να επέμβει ριζικά. Αρκετά εργαλεία που αναφέρθηκαν προηγουμένως, χρησιμοποιούν καινούργιες λειτουργίες της Javascript οπότε καθιστά απαραίτητη η προσθήκη του.

## 2.2.9 Pusher API

Το Pusher API [28], αποτελεί μία βιβλιοθήκη τρίτου προσώπου (third-party library), που καθιστά εφικτή την αποστολή διάφορων συμβάντων (events), με σκοπό την ενημέρωση του εκάστοτε χρήστη σε πραγματικό χρόνο (Real-Time). Δημιουργεί μία μοναδική εμπειρία χρήσης (UI experience) καθώς ο χρήστης θα λάβει ειδοποίηση για κάποιο συμβάν χωρίς την ανανέωση της εφαρμογής. Τα events μπορεί να είναι ενέργειες που πραγματοποιούνται στον

εξυπηρετητή και συνήθως τις προκαλούν οι διαχειριστές της εφαρμογής, είτε για να στείλουν μία γενική ενημέρωση, είτε για κάποια επεξεργασία ενός χρήστη κτλ.

## 2.2.10 Google-Maps

Οι Χάρτες Google (Αγγλικά: Google Maps) [29] είναι υπηρεσία χαρτογράφησης στο Διαδίκτυο. Η εφαρμογή και η τεχνολογία της υπηρεσίας παρέχεται από την Google και υποστηρίζει πολλές υπηρεσίες που βασίζονται σε χάρτες, συμπεριλαμβανομένου της ιστοσελίδας “Χάρτες Google”. Προσφέρει χάρτες δρόμων και σχεδιαστή διαδρομών για μεταφορές με τα πόδια, αυτοκίνητο, ποδήλατο (beta) ή μέσα μαζικής μεταφοράς. Περιλαμβάνει επίσης εντοπισμό των επιχειρήσεων που βρίσκονται σε πόλεις σε πολλές χώρες σε όλο τον κόσμο. Οι δορυφορικές εικόνες των Χαρτών Google δεν ανανεώνονται σε πραγματικό χρόνο, ωστόσο η Google προσθέτει δεδομένα στη Κύρια Βάση δεδομένων της σε τακτική βάση και οι περισσότερες από τις εικόνες δεν είναι πάνω από τριών ετών. Αξιοσημείωτο είναι ότι σε αναβάθμιση τον Ιούλιο του 2012, η Ελλάδα ήταν μία από τις 11 χώρες που επιλέχθηκαν.

## 2.2.11 React-Native

Δεν είναι λάθος να λέμε ότι τα κινητά τηλέφωνα είναι η μισή ψυχή του κάθε ατόμου και όταν πρόκειται για ανάπτυξη Android ή iOS, οι επιχειρήσεις συχνά μπερδεύονται αν πρέπει να προσφέρουν στους καταναλωτές τους, μια εφαρμογή για κινητά με εξαιρετική διεπαφή χρήστη (UI) Εμπειρία χρήστη (UX) ή εφαρμογές που είναι συμβατές με πολλές πλατφόρμες και αναπτύσσονται ταχύτερα. Και ως εκ τούτου, έρχεται το REACT Native Development.

Η REACT Native [30] βοηθά να δημιουργηθούν πραγματικές και συναρπαστικές εφαρμογές για κινητά με τη βοήθεια μόνο του JavaScript, το οποίο υποστηρίζεται τόσο για τις πλατφόρμες Android όσο και για iOS. Η συγγραφή του κώδικα γίνεται μόνο μία φορά και οι εφαρμογές REACT Native είναι διαθέσιμες τόσο για πλατφόρμες iOS όσο και για Android που βοηθούν στην εξοικονόμηση χρόνου ανάπτυξης. Βρήκε μεγάλη δημοτικότητα και καθώς υποστηρίζεται από το facebook, η REACT Native βρήκε τεράστια δημοτικότητα από το κοινό

σήμερα.. Αντίστοιχα, η REACT Native είναι χτισμένη πάνω από το ReactJS, το οποίο ενίσχυση έναν τεράστιο ανταγωνισμό με το άλλο γνωστό JS framework, το AngularJS.

Ο πρωταρχικός λόγος για να χρησιμοποιήσει κάποιος το React Native Framework είναι ότι ο προγραμματιστής πρέπει να γράψει μόνο ένα σύνολο κωδικών, δηλαδή κώδικα JavaScript, για να απολαύσει την απόδοση των εφαρμογών Native Mobile REACT. Μερικά από τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης του REACT Native Framework είναι:

#### **Πλεονεκτήματα:**

- **Έχει χρήση Cross-Platform:** Αρχικά αναπτύχθηκε για την ανάπτυξη iOS εφαρμογών, ύστερα όμως από την επιτυχία και τη δημοτικότητα που προέκυψε το Facebook, αποφάσισε να παρέχει και Android υποστήριξη. Έτσι οι εφαρμογές React Native μπορούν να αναπτυχθούν χρησιμοποιώντας έναν ενιαίο κώδικα και για τις δυο πλατφόρμες ταυτόχρονα, γεγονός που αύξησε ακόμα περισσότερο την δημοτικότητά του.
- **Ίδια λειτουργικότητα σε όλες τις πλατφόρμες:** Αρχικά, οι “υβριδικού” ή “κινητού ιστού” εφαρμογές ήταν δημοφιλείς για τις ασυνήθιστες λειτουργίες τους. Ωστόσο, όταν πρόκειται για επιδόσεις, οι εφαρμογές αυτές τόσο περισσότερο καθυστερούσαν. Έτσι η React Native κατάφερε όχι μόνο να λειτουργούν και στα δύο λειτουργικά συστήματα αλλά και να λειτουργούν με τον ίδιο ακριβώς τρόπο σε όλες τις πλατφόρμες χωρίς καμιά καθυστέρηση.

#### **Μειονεκτήματα:**

- **Βελτιώνεται με τον χρόνο:** Καθώς το framework της React Native υπάρχουν κάποιες λειτουργίες σε iOS και Android κώδικες προγραμματισμού που ακόμα στη React Native δεν έχουν δημιουργηθεί. Βέβαια με τη δημοτικότητα αυτού του framework δεν θα αργήσουν να δημιουργηθούν αντίστοιχες λειτουργίες.
- **Η ύπαρξή του είναι αβέβαιη:** Δεδομένου ότι το συγκεκριμένο framework δημιουργήθηκε από το facebook, η ύπαρξή του είναι αβέβαιη καθώς τα δικαιώματα

του ανήκουν εξ' ολοκλήρου σε αυτό, με αποτέλεσμα να έχει την δυνατότητα ανά πάσα στιγμή να καταστρέψει αυτό το έργο. Αν και με την τρέχουσα δημοτικότητα της τεχνολογίας, είναι απίθανο να συμβεί.

**Συμπέρασμα:** Παρά το γεγονός ότι το framework της React Native έχει ορισμένα κινητά περνά όλες τις δοκιμές και τα εμπόδια και γίνεται η νούμερο ένα επιλογή για σχεδόν κάθε επιχείρηση ανάπτυξης εφαρμογών για κινητά.

## 2.4 Πλατφόρμες Ανάπτυξης Λογισμικού

Στα πλαίσια της υλοποίησης της παρούσας διπλωματικής εργασίας ήταν απαραίτητη και η ανάπτυξη πολύ σύνθετου και πολύπλοκου λογισμικού. Χρησιμοποιήθηκαν, επομένως, εργαλεία λογισμικού που διευκόλυναν την συγγραφή αυτού. Μία βασική ενότητα λογισμικών ανάπτυξης είναι το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment ή IDE). Το IDE είναι πρόγραμμα λογισμικού. Διαθέτει στον προγραμματιστή ένα περιβάλλον στο οποίο παρέχονται συγκεντρωμένες δυνατότητες οι οποίες είναι απαραίτητες για την συγγραφή λογισμικού. Τα βασικά μέρη ενός IDE είναι ο επεξεργαστής πηγαίου κώδικα, εργαλεία για την αυτοματοποιημένη μεταγλώττιση και αποσφαλματωτές. Τα περισσότερα σύγχρονα IDE διαθέτουν επίσης μηχανισμούς για την αυτόματη συμπλήρωση εντολών κατά την συγγραφή κώδικα, μηχανισμούς για την εύρεση λανθασμένης σύνταξης, περιηγητές κλάσεων και αντικειμένων ακόμα και σύστημα διαχείρισης εκδόσεων (Version Control System ή VCS).

### 2.4.1 Visual Studio Code

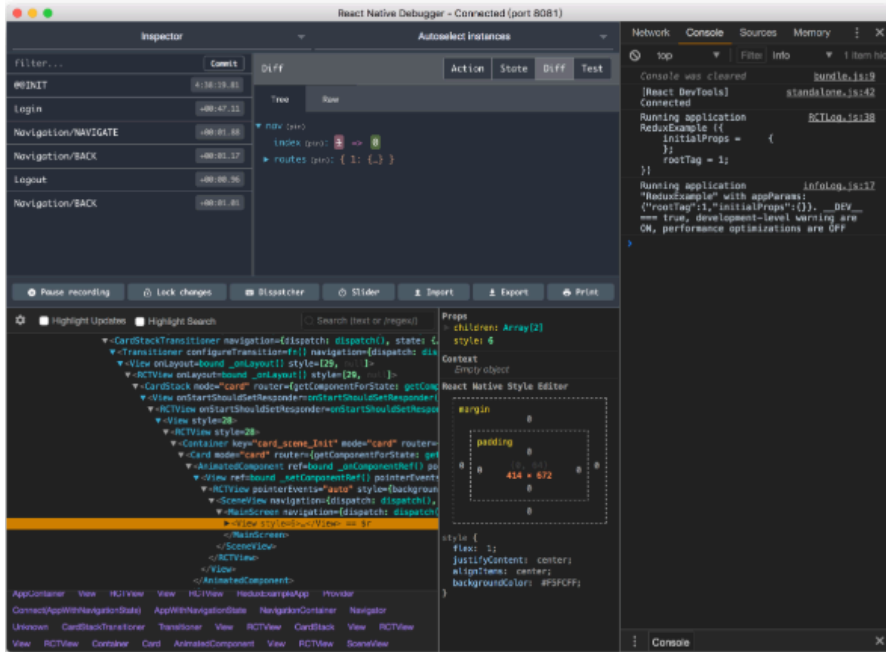
Το Visual Studio Code (VSCode) είναι ένα σύγχρονο IDE, υλοποιημένο από την Microsoft, ανοικτού κώδικα, δωρεάν και με τρομερή απήχηση μεταξύ των προγραμματιστών. Πρόκειται για ένα από τα ελάχιστα δωρεάν IDE's με την ποιότητα και τις λειτουργίες που προσφέρει. Επίσης διαθέτει στο διαδίκτυο μία τεράστια ποικιλία από εξωτερικά εργαλεία και συνδέσεις (plugins). Χρησιμοποιήθηκε στο πρώιμο στάδιο της υλοποίησης της διαδικτυακής εφαρμογής.

## 2.4.2 Expo

Το Expo μια δωρεάν εργαλειοθήκη ανοιχτού κώδικα που βασίζεται στη React Native για να βοηθήσει να δημιουργηθούν native εφαρμογές iOS και Android χρησιμοποιώντας JavaScript και React. Για να δοκιμάσει κάποιος σε μια πραγματική συσκευή iOS χρειάζεται μόνο έναν λογαριασμό προγραμματιστή της Apple (\$ 99 / έτος). Αντίθετα με το Expo μπορεί να το πετύχει αυτό χρησιμοποιώντας το Expo και για τις δύο πλατφόρμες: iOS και Android. Το Expo παρέχει μια εφαρμογή πελάτη και με τη λήψη του από τα αντίστοιχα καταστήματα με βάση την κινητή πλατφόρμα που εκτελεί η συσκευή του κάθε χρήστη, μπορεί εύκολα ο καθένας να τεστάρει τις εφαρμογές με πολύ απλά και εύκολα βήματα.

## 2.4.3 React Native Debugger

Τέλος, χρησιμοποιήθηκε και η εφαρμογή React Native Debugger. Εφ' όσον δεν είναι δυνατόν κατά τη διάρκεια ανάπτυξης μιας React Native εφαρμογής να μπορεί ο προγραμματιστής να έχει στην διαχείριση του τα Εργαλεία Προγραμματιστή (Developer Tools), που παρέχουν οι φυλλομετρητές, δημιουργήθηκε η ανάγκη ανάπτυξης μιας τέτοιας εφαρμογής που θα εκπληρώνει αυτές τις ανάγκες. Η λειτουργία του φαίνεται στην εικόνα:



Εικόνα 15: Παράδειγμα λειτουργίας React Native Debugger

## 2.5 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο, παρουσιάστηκαν όλες οι πληροφορίες και τα εργαλεία που χρειάστηκαν για την ανάπτυξη του ενσωματωμένου συστήματος καθώς και της διαδικτυακής εφαρμογής. Πιο συγκεκριμένα, αναλύθηκαν οι έννοιες του θεωρητικού υπόβαθρου, παρουσιάστηκαν οι γλώσσες προγραμματισμού, οι τεχνολογίες και τα εργαλεία του λογισμικού που χρησιμοποιήθηκαν. Επίσης, δόθηκαν κάποια στιγμιότυπα (screenshot) για την καλύτερη κατανόηση των εννοιών, όπως αυτές εφαρμόστηκαν στην πράξη. Στο επόμενο κεφάλαιο, παρουσιάζονται οι απαιτήσεις του συστήματος, οι λεπτομέρειες για το σχεδιασμό και την ανάπτυξη της βάσης δεδομένων της εφαρμογής, καθώς και τεχνικές που εφαρμόστηκαν για την ασφάλεια του συστήματος.





## Κεφάλαιο 3ο – Δομικά μέρη συστήματος

Σε αυτό το κεφάλαιο, γίνεται παρουσίαση των απαιτήσεων του συστήματος καθώς και των δομικών μερών του. Αρχικά, γίνεται αναλυτική αναφορά στον σχεδιασμό και την υλοποίηση της βάσης δεδομένων. Επιπλέον, αναλύονται, ο τρόπος επίτευξης επικοινωνίας ενσωματωμένου συστήματος και διαδικτυακής εφαρμογής, περιγράφονται οι ρόλοι των χρηστών στην εφαρμογή και τέλος, παρουσιάζονται τεχνικές και εργαλεία που χρησιμοποιούνται για την επίτευξη της ασφάλειας του συστήματος.

### 3.1 Απαιτήσεις συστήματος

Για την επιτυχή ολοκλήρωση κάθε έργου, ένας πολύ σημαντικός παράγοντας είναι ο σωστός καθορισμός των απαιτήσεων από την αρχική περιγραφή του, μιας και αυτές καθορίζουν την πορεία του έργου. Συγκεκριμένα, για το παρόν έργο, στόχος ήταν η δημιουργία ενός ενσωματωμένου συστήματος, παράλληλα με μία διαδικτυακή εφαρμογή και ταυτόχρονα, η ακριβής καταγραφή και αποστολή μετρήσεων μεταξύ των δύο. Παρακάτω παρατίθενται αναλυτικά οι απαιτήσεις που εξήχθησαν συνοδευόμενες από επεξήγηση.

- **Το σύστημα διαχειρίζεται δύο κατηγορίες χρηστών.** Αυτές αποτελούνται από: τους εθελοντές και τους διαχειριστές. Οι εθελοντές έχουν δυνατότητες ανάκτησης κωδικού μέσω email και επεξεργασίας προσωπικών στοιχείων, κλήσης έκτακτης ανάγκης στο ΕΚΑΒ, έχουν επίσης τη δυνατότητα να θέσουν τον εαυτό τους ανενεργό έτσι ώστε να μη δέχονται ειδοποιήσεις και τέλος, έχουν δυνατότητα απόρριψης περιστατικού ή ανταπόκρισης με ανακατεύθυνση στους χάρτες του κινητού. Οι διαχειριστές αποτελούν στην ουσία χρήστες, με σαφώς αυξημένες δυνατότητες επεξεργασίας, δημιουργίας και ελέγχου, καθώς τους δίνονται οι δυνατότητες για δημιουργία νέων εθελοντών, επεξεργασία των ήδη υπαρχόντων εθελοντών, δημιουργία και επεξεργασία νέων διαχειριστών, τοποθέτηση νέων απινιδωτών και επεξεργασία των στοιχείων τους και τέλος, το πιο σημαντικό, εισαγωγή νέου περιστατικού ύστερα από αναζήτηση διεύθυνσης στο χάρτη.
- **Το σύστημα παρέχει στον εθελοντή λειτουργία ανάκτησης κωδικού πρόσβασης.** Σε περίπτωση απώλειας κωδικού ο εθελοντής έχει τη δυνατότητα να εισάγει το email του με σκοπό να του αποσταλεί σύνδεσμος μιας χρήσης για εισαγωγή νέου κωδικού.

- **Ειδοποίηση και ενημέρωση διαχειριστή.** Ο εκάστοτε διαχειριστής του συστήματος δέχεται ενημέρωση για διάφορες λειτουργίες και ενέργειες σε πραγματικό χρόνο όπως το ξεκλείδωμα, την απώλεια, ή την μεγάλη μεταβολή της θερμοκρασίας των απινιδωτών. Οι αλλαγές στο UI γίνονται ασύγχρονα, γεγονός που βοηθάει στην απόκριση και ποιότητα της εφαρμογής, καθώς ο χρήστης δεν απαιτείται να ανανεώνει χειροκίνητα την σελίδα, για κάθε αλλαγή που συμβαίνει.
- **Απλό και φιλικό περιβάλλον χρήσης καθώς και αποκριτική σχεδίαση της εφαρμογής (UI friendly and responsive design).** Το περιβάλλον χρήσης της εφαρμογής είναι δομημένο και υλοποιημένο με τέτοιο τρόπο, ώστε να γίνεται κατανοητό και εύκολα διαχειρίσιμο, ακόμα και από άτομα με ελάχιστες γνώσεις ή εμπειρία στην χρήση υπολογιστή. Η απλότητά του έχει ως αποτέλεσμα όχι μόνο την ευκολία στην χρήση, αλλά και να καθοδηγεί τον χρήστη στην εύρεση των λειτουργιών που αναζητεί. Τέλος η αποκριτική του σχεδίαση επιτρέπει την διατήρηση της ποιότητας, της εμφάνισης και των λειτουργιών της εφαρμογής, από οποιαδήποτε συσκευή χρησιμοποιείται, οποιωνδήποτε διαστάσεων (κινητό τηλέφωνο, tablet ή Η/Υ).
- **Απαιτείται πρόσβαση στο διαδίκτυο.** Η επικοινωνία διαχειριστή και εθελοντή τόσο για την ενημέρωση περιστατικών ή μηνυμάτων απαιτεί πρόσβαση στο διαδίκτυο. Το ίδιο απαιτεί και η αμφίδρομη επικοινωνία μεταξύ διαχειριστή και απινιδωτή για την απασφάλιση του όπως και από τον απινιδωτή στον διαχειριστή σε μεγάλες μη επιθυμητές μεταβολές τις θερμοκρασίας ή απώλεια.
- **Κρίνεται αναγκαία η εφαρμογή της ασφάλειας σε κάθε πτυχή του συστήματος.** Αυτό σημαίνει πως η επικοινωνία μεταξύ της εφαρμογής και των σταθμών καθώς και του χρήστη με την διαδικτυακή εφαρμογή, υποστηρίζει πολλές μορφές ασφαλείας, κάθε μορφή υπεύθυνη για τον τομέα ευθύνης της.
- **Επεκτάσιμο και συντηρήσιμο.** Οι έννοιες αυτές ορίζουν πως το σύστημα θα πρέπει να κατασκευαστεί, είτε σε επίπεδο κώδικα είτε σε επίπεδο υλικής κατασκευής, με τρόπο εύκολα κατανοητό από κάποιον τρίτο. Στόχος αυτού είναι ο κάθε εμπλεκόμενος με το σύστημα να μπορεί εύκολα να κατανοεί τη ροή των γεγονότων και να επεμβαίνει όπου χρειάζεται. Τέλος πρέπει να τονιστεί πως μία από τις σημαντικές αρχές για την δημιουργία

επεκτάσιμου και συντηρήσιμου κώδικα είναι η αποφυγή της επανάληψης κομματιών κώδικα.

## 3.2 Περιπτώσεις χρήσης (use cases)

Το συγκεκριμένο κεφάλαιο περιέχει τις λειτουργίες που θα μπορεί να υλοποιήσει ο χρήστης της ιστοσελίδας. Η σελίδα που θα δημιουργήσουμε θα υποστηρίζει 3 ειδών διαφορετικούς χρήστες. Κάθε χρήστης θα εισέρχεται στο σύστημα μέσω ελεγχόμενης εισόδου του ονόματος χρήστη και του κωδικού που του έχουν παραχωρηθεί από την περιφέρεια. Ο τύπος αυτών των χρηστών θα είναι: ο διαχειριστής, ο εθελοντής, το προσωπικό ΕΚΑΒ.

- Εθελοντής

Ο εθελοντής θα μπορεί να υλοποιήσει τις παρακάτω λειτουργίες:

**Περίπτωση χρήσης 1:** Είσοδος στο σύστημα . Ο εθελοντής θα εισέρχεται στο σύστημα με τη χρήση των προσωπικών του στοιχείων username και password τα οποία θα εισάγει στη φόρμα συμπλήρωσης που θα εμφανίζεται στην αρχική σελίδα. Τα συγκεκριμένα στοιχεία θα του παρέχονται ύστερα από επιτυχή παρακολούθηση κάποιων σεμιναρίων. Η είσοδος θα επιτρέπεται μόνο όταν ο συνδυασμός των πεδίων είναι σωστός.

**Περίπτωση χρήσης 2:** Εμφάνιση του χάρτη και της τοποθεσίας των απινιδωτών και πιθανών περιστατικών.

**Περίπτωση χρήσης 3:** Εμφάνιση ιστορικού περιστατικών.

**Περίπτωση χρήσης 4:** Επικοινωνία με τον διαχειριστή μέσω μιας ειδικής φόρμας.

**Περίπτωση χρήσης 5:** Σε περίπτωση περιστατικού δυνατότητα χρησιμοποίησης του πλησιέστερου απινιδωτή για την άμεση βοήθεια.

**Περίπτωση χρήσης 6:** Έξοδος από το σύστημα θα υπάρχει πεδίο "αποσύνδεση" με το οποίο θα μπορεί να αποσυνδεθεί από τη σελίδα.

- Προσωπικό ΕΚΑΒ

Θα εισέρχεται στο σύστημα με το δικό όνομα χρήστη και κωδικό ώστε να μπορεί να έχει πρόσβαση στις λειτουργίες της σελίδας τα οποία θα του παραχωρηθούν. Θα μπορεί να υλοποιήσει τις ίδιες λειτουργίες με το εθελοντή όπως αυτές ορίστηκαν παραπάνω.

- Διαχειριστής

Ο διαχειριστής της σελίδας θα έχει δικαιώματα πλήρους χρήσης της σελίδας. Αυτό σημαίνει πως θα είναι σε θέση να δει, να επεξεργαστεί και να διαγράψει καταχωρήσεις όπως συμβαίνει με τον εθελοντή ή το προσωπικό. Θα είναι υπεύθυνος ώστε να εξασφαλίζεται η αποδοτικότητα και η αποτελεσματική λειτουργία της σελίδας. Για αυτό το λόγο θα υποστηρίζει όλες τις λειτουργίες που μπορούν να υλοποιήσουν οι εθελοντές και το ΕΚΑΒ. Επίσης ο διαχειριστής θα έχει ακόμα την δυνατότητα να κάνει άρση των δικαιωμάτων των παραπάνω χρηστών σε περίπτωση κακόβουλης χρήσης ή μη παρακολούθησης των σεμιναρίων για ένα περιορισμένο διάστημα ή μόνιμως. Με αυτό τον τρόπο θα μπορεί να έχει άμεση πρόσβαση στη σελίδα και να διορθώσει πιθανές δυσλειτουργίες που θα εμφανιστούν. Οι δυσλειτουργίες αυτές μπορεί να εμφανίζονται είτε σε επίπεδο εμφάνισης (front-end) είτε σε επίπεδο υλοποίησης εργασιών σε συνεργασία με τη βάση δεδομένων(back-end).

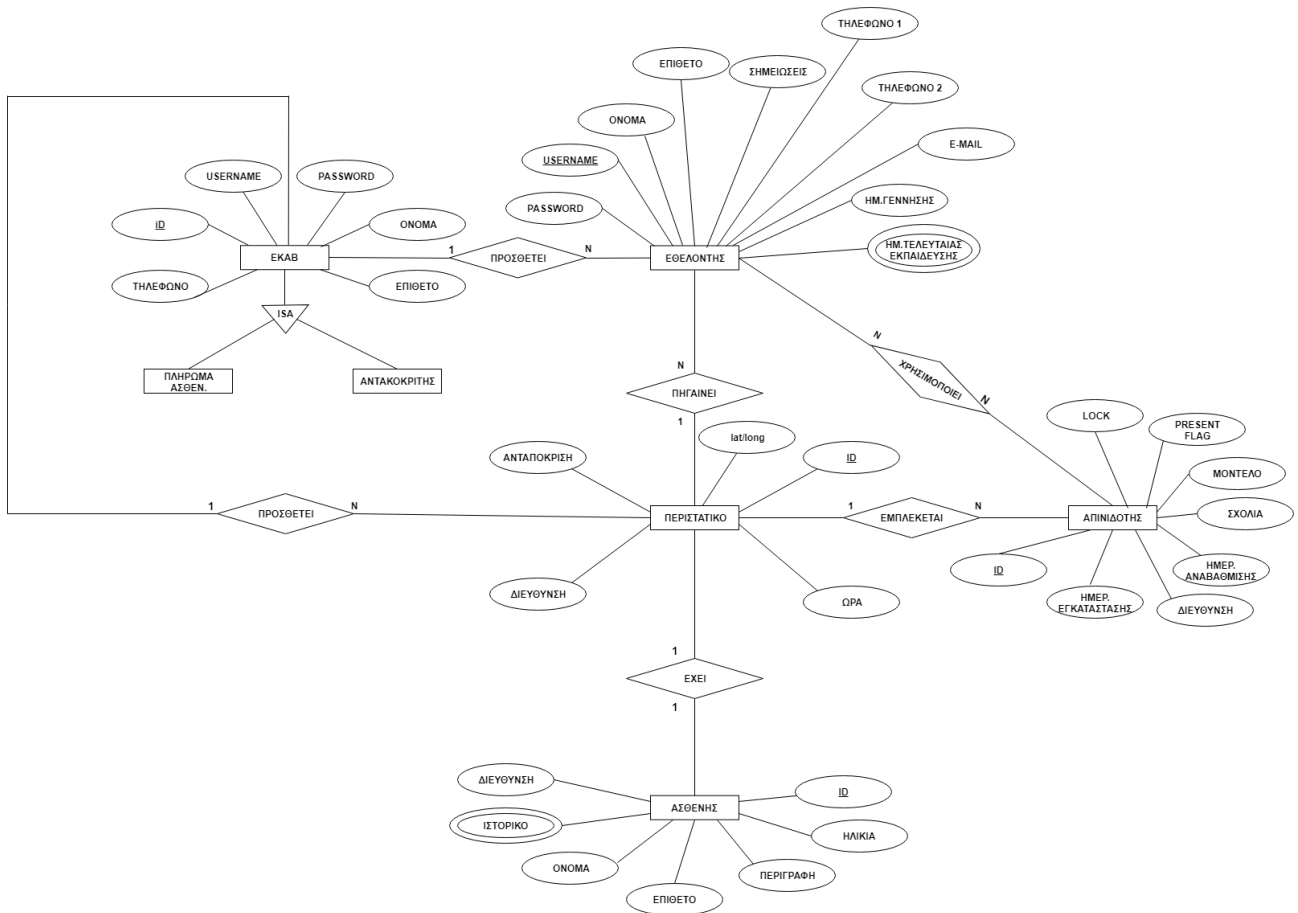
### 3.3 Ανάλυση βάσης δεδομένων

Μετά την ανάλυση και τον προσδιορισμό των απαιτήσεων του συστήματος, το επόμενο βήμα για τον προγραμματιστή είναι ο σχεδιασμός της βάσης δεδομένων. Ο κατά το δυνατόν καλύτερος σχεδιασμός της βάσης δεδομένων αποτελεί υψίστης σημασίας εργασία και χτίζεται με έμφαση στον τρόπο που οι πίνακες συνδέονται μεταξύ τους, με σχέσεις ανάμεσα στις εγγραφές τους. Ο βέλτιστος σχεδιασμός μπορεί να επιτευχθεί με την κανονικοποίηση (normalization) της βάσης δεδομένων, χάρη στην οποία αφαιρείται η πλεονάζουσα πληροφορία, αυξάνεται η σαφήνεια της οργάνωσης της βάσης και

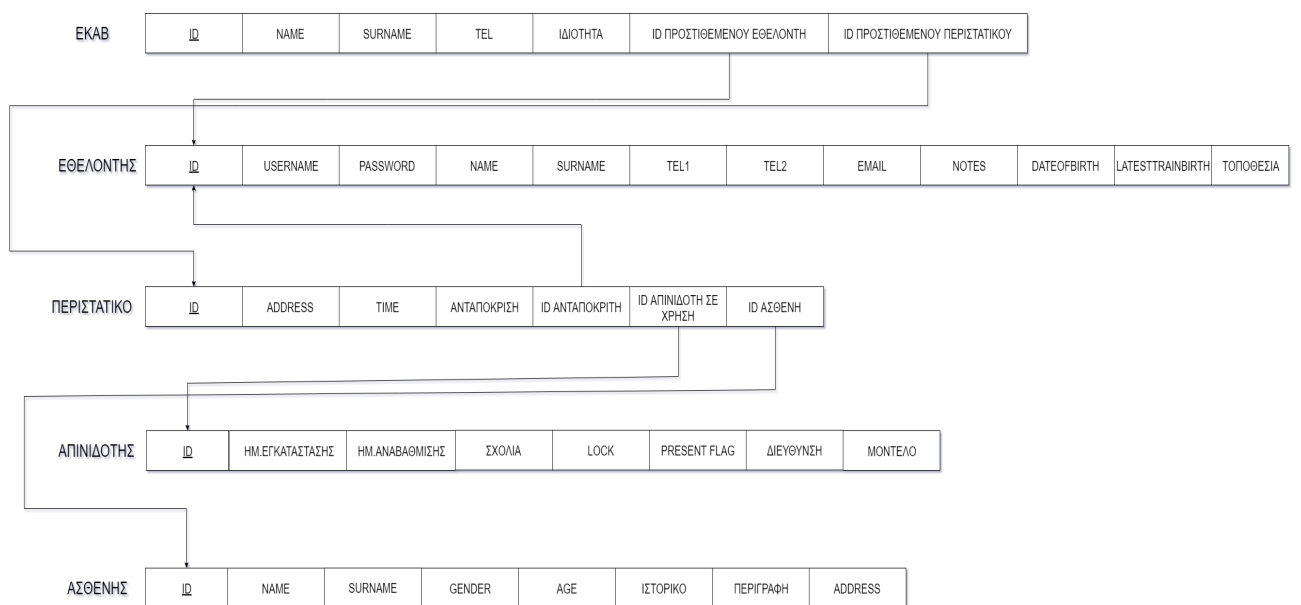
απλοποιείται η διαχείρισή της. Επιπλέον, η κανονικοποίηση της βάσης δεδομένων επιτρέπει την ευκολότερη και αποδοτικότερη τροποποίησή της στο μέλλον, έτσι ώστε να καλύπτει τις προδιαγραφές μίας συνεχώς αναβαθμιζόμενης και λειτουργικά επεκτεινόμενης εφαρμογής. Παρακάτω παρατίθεται αρχικά, το διάγραμμα οντοτήτων συσχετίσεων, ώστε ο αναγνώστης να λάβει μία πλήρη εικόνα για το πως είναι δομημένη η βάση με τους πίνακες και τι συσχετίσεις υπάρχουν μεταξύ των πινάκων και έπειτα, ακολουθεί η παρουσίαση κάθε πίνακα ξεχωριστά, μαζί με μία αναλυτική περιγραφή για κάθε στήλη, κάτω από κάθε πίνακα, για να αποσαφηνιστεί ο λόγος ύπαρξης της.

### 3.3.1 E\_R και σχεσιακό διάγραμμα βάσης δεδομένων

Στην εικόνα που ακολουθεί, αποτυπώνεται μία ολοκληρωμένη αναπαράσταση της δομής της βάσης δεδομένων. Πιο συγκεκριμένα, φαίνονται όλοι οι πίνακες με τα πεδία τους, τους τύπους δεδομένων και τις εξαρτήσεις τους.



Εικόνα 16: E\_R βάσης δεδομένων HeartBit



Εικόνα 17: Σχισιακό διάγραμμα βάσης δεδομένων HeartBit

### 3.3.2 Δομή πίνακα “Εθελοντής”

Στον πίνακα 2 φαίνεται η δομή του πίνακα «εθελοντής» ο οποίος περιέχει τις βασικές πληροφορίες των χρηστών. Στη συνέχεια ακολουθεί ο πίνακας του ΕΚΑΒ, που είναι και ο κύριος διαχειριστής της πλατφόρμας και έχει πλήρη συσχέτιση με τον εθελοντή. Έπειτα ακολουθεί ο πίνακας του εκάστοτε περιστατικού με τα απαραίτητα στοιχεία και του απινιδωτή που αλληλοσυνδέονται μεταξύ τους με το πίνακα εθελοντής. Τέλος έχουμε τον πίνακα Ασθενής με τα στοιχεία του.

Όνομα	Τύπος	Κενό	Προεπιλογή	Πρόσθετα
<u>Id</u>	Int(11)	Όχι	Καμία	AUTO_INCREMENT
Username	Varchar(25)	Όχι	Καμία	UNIQUE
Password	Varchar(64)	Όχι	Καμία	-
Name	Varchar(25)	Ναι	Καμία	-
Surname	Varchar(25)	Ναι	Καμία	-
Tel1	Bigint(10)	Ναι	Καμία	-
Tel2	Bigint(10)	Ναι	Καμία	-
E-mail	Varchar(30)	Όχι	Καμία	UNIQUE
Notes	Varchar(100)	Ναι	Καμία	-
Latest Training	Date()	Ναι	Καμία	-
Date of Birth	Date()	Ναι	Καμία	-
Address	Varchar(30)	Ναι	Καμία	-

Πίνακας 2: Πίνακας “Εθελοντή”

Ανάλυση στηλών του πίνακα “Εθελοντής”:

- **id:** Αποτελεί το αναγνωριστικό του χρήστη και είναι το πρωτεύον κλειδί του πίνακα. Έχει τη δυνατότητα να λάβει πολύ μεγάλες τιμές ώστε να καλύπτει τις ανάγκες έπειτα από χρόνια.
- **username:** Αποτελεί το όνομα του χρήστη και χρησιμοποιείται για τη σύνδεσή του στο σύστημα. Αποτελεί συμβολοσειρά και είναι μοναδική για κάθε χρήστη νοουμένου πως είναι το εκ των δύο στοιχείων σύνδεσής του στο σύστημα. Το username θα δίδεται απο τον διαχειριστή.

- **password:** Είναι ο κωδικός του χρήστη, αποτελεί συμβολοσειρά μήκους 64 χαρακτήρων με σκοπό την συμβατότητα στην κρυπτογράφηση (hash). Το password θα δίδεται και αυτό από τον διαχειριστή.
- **Name:** Συμβολοσειρά μήκους 25 χαρακτήρων, αποτελεί το όνομα του χρήστη.
- **Surname:** Συμβολοσειρά μήκους 25 χαρακτήρων, αποτελεί το επώνυμο του χρήστη.
- **email:** Είναι το email του χρήστη. Αποτελεί επίσης μοναδικό στοιχείο διότι αφενός αποστέλλονται ειδοποιήσεις μέσω emails και αφετέρου είναι το πεδίο το οποίο χρησιμοποιείται για την επαναφορά κωδικού.
- **Telephone 1:** Είναι το κινητό τηλέφωνο επικοινωνίας του χρήστη. Σε αυτό το νούμερο, το σύστημα αποστέλλει αυτόματα ειδοποιήσεις με SMS. Αποτελείται από 10 ακέραιους αριθμούς όπως όλοι οι τηλεφωνικοί αριθμοί.
- **Telephone 2:** Είναι το κινητό τηλέφωνο επικοινωνίας του χρήστη. Σε αυτό το νούμερο, το σύστημα αποστέλλει αυτόματα ειδοποιήσεις με SMS. Αποτελείται από 10 ακέραιους αριθμούς όπως όλοι οι τηλεφωνικοί αριθμοί.
- **Notes:** Συμβολοσειρά 100 χαρακτήρων, δίνει τη δυνατότητα στον διαχειριστή για προσθήκη σχολίων στον εκάστοτε εθελοντή.
- **Latest Training:** Τύπος δεδομένων ημερομηνίας της μορφής YYYY-MM-DD, αποτελεί την ημερομηνία της τελευταίας εκπαίδευσης του εθελοντή.
- **Date of Birth:** Τύπος δεδομένων ημερομηνίας της μορφής YYYY-MM-DD, αποτελεί την ημερομηνία γέννησης του εθελοντή.
- **Address:** Συμβολοσειρά 30 χαρακτήρων, αποτελεί την διεύθυνση του χρήστη.



### 3.3.3 Δομή πίνακα “EKAB”

Στον Πίνακα 3 παρουσιάζεται η δομή του πίνακα «EKAB » ο οποίος περιέχει και τα βασικά του στοιχεία. Ο συγκεκριμένος πίνακας αποτελεί τα στοιχεία από τον διαχειριστή της πλατφόρμας.

Όνομα	Τύπος	Κενό	Προεπιλογή	Πρόσθετα
<u>Id</u>	Int(11)	Όχι	Καμία	AUTO_INCREMENT
Type	Varchar(25)	Όχι	Καμία	NOT NULL
Name	Varchar(25)	Ναι	Καμία	-
surname	Varchar(25)	Ναι	Καμία	-
Tel	Bigint(10)	Ναι	Καμία	-
e-mail	Varchar(30)	Όχι	Καμία	UNIQUE
Address	Varchar(30)	Ναι	Καμία	-
username	Varchar(25)	Όχι	Καμία	UNIQUE
password	Varchar(64)	Όχι	Καμία	-

Πίνακας 3: Πίνακας “EKAB”

Ανάλυση στηλών του πίνακα “EKAB”:

- **id:** Αποτελεί το αναγνωριστικό του χρήστη και είναι το πρωτεύον κλειδί του πίνακα. Έχει τη δυνατότητα να λάβει πολύ μεγάλες τιμές ώστε να καλύπτει τις ανάγκες έπειτα από χρόνια.
- **Type:** Αποτελεί την ιδιότητα του χρήστη που μπορεί να είναι και ο διαχειριστής. Στην προκειμένη περίπτωση μπορεί να είναι το πλήρωμα ασθενοφόρου του EKAB ή ο τηλεφωνικός ανταποκριτής.
- **Name:** Συμβολοσειρά μήκους 25 χαρακτήρων, αποτελεί το όνομα του χρήστη.
- **Surname:** Συμβολοσειρά μήκους 25 χαρακτήρων, αποτελεί το επώνυμο του χρήστη.
- **email:** Είναι το email του χρήστη. Αποτελεί επίσης μοναδικό στοιχείο διότι αφενός αποστέλλονται ειδοποιήσεις μέσω emails και αφετέρου είναι το πεδίο το οποίο χρησιμοποιείται για την επαναφορά κωδικού.

- **telephone:** Είναι το κινητό τηλέφωνο επικοινωνίας του χρήστη. Σε αυτό το νούμερο, το σύστημα αποστέλλει αυτόματα ειδοποιήσεις με SMS. Αποτελείται από 10 ακέραιους αριθμούς όπως όλοι οι τηλεφωνικοί αριθμοί.
- **Address:** Συμβολοσειρά 30 χαρακτήρων, αποτελεί την διεύθυνση του χρήστη.
- **username:** Αποτελεί το όνομα του χρήστη και χρησιμοποιείται για τη σύνδεσή του στο σύστημα. Αποτελεί συμβολοσειρά και είναι μοναδική για κάθε χρήστη νοουμένου πως είναι το εκ των δύο στοιχείων σύνδεσής του στο σύστημα. Το username θα δίδεται απο τον διαχειριστή.
- **password:** Είναι ο κωδικός του χρήστη, αποτελεί συμβολοσειρά μήκους 64 χαρακτήρων με σκοπό την συμβατότητα στην κρυπτογράφηση (hash). Το password θα δίδεται και αυτό από τον διαχειριστή.

### 3.2.4 Δομή πίνακα “Περιστατικό”

Στον Πίνακα 4 παρουσιάζεται η δομή του πίνακα «περιστατικό » ο οποίος περιέχει και τα βασικά του στοιχεία. Ο συγκεκριμένος πίνακας συνδέεται άμεσα με τον πίνακα εθελοντή. Αφού σε περίπτωση περιστατικού ο εθελοντής σπεύδει να επέμβει.

Όνομα	Τύπος	Κενό	Προεπιλογή	Πρόσθετα
<u>Id</u>	Int(11)	Όχι	Καμία	AUTO_INCREMENT
correspondence	Varchar(3)	Ναι	Καμία	NOT NULL
e-mail	Varchar(30)	Όχι	Καμία	UNIQUE
address	Varchar(30)	Ναι	Καμία	-
lat/long	Decimal(9,6)	Όχι	Καμία	-

Πίνακας 4: Πίνακας “Περιστατικό”

Ανάλυση στηλών του πίνακα “Περιστατικό”:

- **id:** Αποτελεί το αναγνωριστικό του χρήστη και είναι το πρωτεύον κλειδί του πίνακα και αυξάνεται κατά ένα για την καταμέτρηση του πλήθους των περιστατικών. Έχει τη δυνατότητα να λάβει πολύ μεγάλες τιμές ώστε να καλύπτει τις ανάγκες έπειτα από χρόνια.

- **Correspondence:** Αποτελεί συμβολοσειρά 3 χαρακτήρων και εκπληρώνει την ανάγκη για να γνωρίζουμε αν υπάρχει ανταπόκριση στο περιστατικό ή όχι.
- **email:** Είναι το email του χρήστη. Αποτελεί επίσης μοναδικό στοιχείο διότι αφενός αποστέλλονται ειδοποιήσεις μέσω emails και αφετέρου είναι το πεδίο το οποίο χρησιμοποιείται για την επαναφορά κωδικού.
- **Address:** Συμβολοσειρά 30 χαρακτήρων, αποτελεί την διεύθυνση του χρήστη.
- **lat/long:** Αποτελεί τις συντεταγμένες του περιστατικού στον χάρτη, πιο συγκεκριμένα latitude το γεωγραφικό πλάτος και longitude το γεωγραφικό μήκος.

### 3.3.5 Δομή πίνακα “Απινιδωτής”

Ο πίνακας 5 περιέχει τη δομή του πίνακα περιστατικό καθώς και τα βασικά του στοιχεία. Συνδέεται άμεσα με τον εθελοντή αφού χρησιμοποιεί έναν απινιδωτή σε περίπτωση περιστατικού.

Όνομα	Τύπος	Κενό	Προεπιλογή	Πρόσθετα
<u>Id</u>	Int(11)	Όχι	Καμία	AUTO_INCREMENT
installationdate	Date()	Όχι	Καμία	-
upgradedate	Date()	Ναι	Καμία	-
Notes	Varchar(100)	Ναι	Καμία	-
location	Varchar(30)	Ναι	Καμία	-
Model	Varchar(30)	Όχι	Καμία	-
presentflag	Boolean()	Όχι	0	NULL
Locker	Boolean()	Όχι	0	NULL

Πίνακας 5: Πίνακας “Απινιδωτής”

Ανάλυση στηλών του πίνακα “Απινιδωτής”:

- **id:** Αποτελεί το αναγνωριστικό του χρήστη και είναι το πρωτεύον κλειδί του πίνακα. Έχει τη δυνατότητα να λάβει πολύ μεγάλες τιμές ώστε να καλύπτει τις ανάγκες έπειτα από χρόνια.
- **Installation Date:** την ημέρα την οποία εγκαταστάθηκε ο κάθε απινιδωτής.
- **Upgrade Date:** Τύπος δεδομένων ημερομηνίας της μορφής YYYY-MM-DD , και δηλώνει την ημέρα την οποία αναβαθμίστηκε ή ελέγχθηκε ένας απινιδωτής έτσι ώστε να γνωρίζουμε ότι βρίσκεται πάντοτε σε λειτουργία.
- **Notes:** Αποτελεί συμβολοσειρά 30 χαρακτήρων και δηλώνει την γενική περιγραφή του περιστατικού.
- **Location:** Αποτελεί συμβολοσειρά 30 χαρακτήρων και δηλώνει την τοποθεσία του περιστατικού.
- **Model:** Αποτελεί συμβολοσειρά 30 χαρακτήρων και αποτελεί το όνομα του μοντέλου του απινιδωτή.
- **Present Flag:** Αποτελεί τύπο δεδομένων boolean, δηλαδή παίρνει δύο τιμές 0 και 1. 0 εάν ο απινιδωτής είναι στη θέση του, 1 εάν έχει κλαπεί ή μετακινηθεί.
- **Lock:** Αποτελεί τύπο δεδομένων boolean, δηλαδή παίρνει δύο τιμές 0 και 1. 0 εάν ο απινιδωτής είναι κλειδωμένος, 1 εάν έχει ξεκλειδωθεί.

### 3.3.6 Δομή πίνακα “ασθενής”

Ο πίνακας 6 περιέχει τη δομή του πίνακα ασθενής καθώς και τα βασικά του στοιχεία. Συνδέεται άμεσα με το περιστατικό αφού το περιστατικό προϋποθέτει την ύπαρξη ασθενή.

Όνομα	Τύπος	Κενό	Προεπιλογή	Πρόσθετα
<u>Id</u>	Int(11)	Όχι	Καμία	AUTO_INCREMENT
history	Varchar(1000)	Όχι	Καμία	NOT NULL
description	Varchar(100)	Όχι	Καμία	NOT NULL
Name	Varchar(25)	Ναι	Καμία	-
surname	Varchar(25)	Ναι	Καμία	-

birthdate	Bigint(4)	Όχι	Καμία	-
gender	Varchar(1)	Όχι	Καμία	NOT NULL
address	Varchar(30)	Ναι	Καμία	-

Πίνακας 6: Πίνακας “Ασθενής”

Ανάλυση στηλών του πίνακα “Ασθενής”:

- **id:** Αποτελεί το αναγνωριστικό του χρήστη και είναι το πρωτεύον κλειδί του πίνακα. Έχει τη δυνατότητα να λάβει πολύ μεγάλες τιμές ώστε να καλύπτει τις ανάγκες έπειτα από χρόνια.
- **history:** Αποτελεί συμβολοσειρά 1000 χαρακτήρων και δηλώνει το ιστορικό ασθενειών του ασθενή στη βάση.
- **description:** Αποτελεί συμβολοσειρά 100 χαρακτήρων και δηλώνει την γενική περιγραφή του περιστατικού.
- **name:** Συμβολοσειρά μήκους 25 χαρακτήρων, αποτελεί το όνομα του χρήστη.
- **surname:** Συμβολοσειρά μήκους 25 χαρακτήρων, αποτελεί το επώνυμο του χρήστη.
- **Birth Date:** Δηλώνει την χρονολογία γέννησης του ασθενή του ασθενή
- **gender:** Συμβολοσειρά μήκους 1 χαρακτήρα, πιο συγκεκριμένα m για male και f για female, αποτελεί το γένος του χρήστη.
- **address:** Συμβολοσειρά 30 χαρακτήρων, αποτελεί την διεύθυνση του χρήστη.

### 3.4 Ασφάλεια συστήματος

Τα τελευταία χρόνια η επίτευξη της ασφάλειας έχει γίνει άκρως απαραίτητη προϋπόθεση και δεν αποτελεί προαιρετικό χαρακτηριστικό. Ο όρος ασφάλεια καλύπτει πολλά θέματα, από την επίθεση στο σύστημα (hardware/software) από έναν hacker, ως τη δυσλειτουργία του συστήματος. Η αναφορά σε ζητήματα ασφάλειας διαδικτυακών εφαρμογών φέρνει στο μυαλό μας εικόνες επιτιθέμενων σε ιστότοπους, οι οποίοι «αρπάζουν» στοιχεία πιστωτικών καρτών ή εκτελούν επιθέσεις άρνησης εξυπηρέτησης (denial of service). Ωστόσο, αυτό είναι μόνο ένα μέρος του συνολικότερου προβλήματος, το οποίο καλούμαστε να

αντιμετωπίσουμε, όταν επιθυμούμε να προστατεύσουμε μία διαδικτυακή εφαρμογή. Οι κώδικας που αναπτύσσει ο προγραμματιστής, εκτός από τη σωστή λειτουργικότητα, θα πρέπει να είναι και ασφαλής (με κάθε έννοια του όρου). Ως εκ τούτου, θα πρέπει να χρησιμοποιούνται όλες οι βέλτιστες πρακτικές ασφαλείας και οι συνιστώμενες τεχνικές. Σε αυτό το σημείο μπαίνουν στο παιχνίδι, τα διάφορα frameworks και οι έτοιμες λειτουργίες που προσφέρουν περί ασφαλείας. Λόγω των πολλών κατηγοριών επιθέσεων που μπορεί να προκύψουν, οι οποίες ολοένα και αυξάνονται με τα χρόνια και γίνονται ακόμα πιο πολύπλοκες, ο προγραμματιστής θα πρέπει να διαθέτει και άριστες γνώσεις δικτύων, servers, πρωτοκόλλων κτλ. πράγμα το οποίο φαντάζει αρκετά δύσκολο. Επίσης λόγω ημιμάθειας, παρατηρείται συχνά το φαινόμενο, όπου ένας προγραμματιστής “νομίζει” ότι έχει λάβει όλα τα απαραίτητα μέτρα ασφαλείας αντιγράφοντας κομμάτια κώδικα που χρησιμεύουν για αυτό τον λόγο, δίχως ωστόσο να τεστάρει την εφαρμογή του δημιουργώντας στο τέλος, αρκετά κενά ασφαλείας, αφήνοντας έτσι την εφαρμογή του εκτεθειμένη. Για να εξαλειφθούν τέτοιες καταστάσεις, μία ασφαλής και σίγουρη τεχνική είναι, ο προγραμματιστής να “αφήνει” κατά κάποιον τρόπο την ασφάλεια στα “χέρια” του framework ώστε αυτός να παραμένει προσηλωμένος στην ανάπτυξη της εφαρμογής. Το Slim ως θερμός οπαδός της ασφάλειας, παρέχει όλα τα εργαλεία που χρειάζεται μία σύγχρονη διαδικτυακή εφαρμογή για να θεωρείται ασφαλής. Παρακάτω παρουσιάζονται διάφορες τεχνικές ασφαλείας που χρησιμοποιήθηκαν στην διαδικτυακή εφαρμογή.

- **Ασφάλεια από επιθέσεις ψεκασμού εντολών SQL (SQL injection) [37].** Αποτελεί επίθεση που στοχεύει στην αλλαγή των εντολών SQL που επεξεργάζονται την βάση δεδομένων. Χωρίς την χρήση κάποιου είδους προστασίας, μπορεί ο χρήστης να περάσει μαζί με μία προκαθορισμένη εντολή SQL και μία δεύτερη που θα “ρίχνει” (drop) έναν πίνακα της βάσης. Μία λύση στην επίθεση αυτή είναι η χρήση της λειτουργίας PDO [38], η οποία προτείνεται από την γλώσσα PHP σαν ένας τρόπος προστασίας στην συγκεκριμένη επίθεση. Στην ουσία “σπάει” σε τμήματα (bindings) τα δεδομένα που δίνονται σε μία εντολή SQL και τα ελέγχει ατομικά.
- **Λειτουργία συνοδού (session) και στρωμάτων ασφαλείας (middlewares) στο κομμάτι αυθεντικοποίησης, λειτουργία κλειδιού (api\_token) στο κομμάτι του api.** Η διαδικτυακή εφαρμογή χωρίζεται σε δύο κομμάτια. Στο κομμάτι αυθεντικοποίησης και στο api. Στο πρώτο κομμάτι, αφού ο χρήστης συνδεθεί, του δίνεται συνοδός (session) 120

λεπτών για πλοήγηση στην εφαρμογή (με το πέρας των 120 λεπτών, η εφαρμογή αυτομάτως τον αποσυνδέει). Παράλληλα του δημιουργείται ένα `api_token` που τον βοηθάει στην ταυτοποίησή του από τον server, στα requests που θα κάνει στο api. Ο χρήστης δεν γνωρίζει το `api_token` ούτε του δίνεται κάποια πληροφορία για το που βρίσκεται. Έπειτα ο χρήστης οδηγείται στο κεντρικό κομμάτι της εφαρμογής, όπου αναλαμβάνει η React μαζί με το React-router και το Redux. Εκεί πλοηγείται στις ενότητες της εφαρμογής, με μοναδικό απαιτούμενο στοιχείο, το `api_token`. Με την βοήθεια των στρωμάτων ασφαλείας (middlewares), ο εκάστοτε χρήστης σε κάθε ανακατεύθυνση της σελίδας, ελέγχεται για την ιδιότητά του (διαχειριστής, χρήστης) και του επιτρέπεται ή του απορρίπτεται αντίστοιχα η πλοήγηση σε διάφορα τμήματα της εφαρμογής. Τα middlewares αποτελούν κομμάτι του Laravel, δημιουργούνται και ορίζονται από τον χρήστη.

- **Post request σε συγκεκριμένο api endpoint με την χρήση μοναδικού κωδικού (unique) για τον σταθμό.** Κατά την διαδικασία δημιουργίας ενός σταθμού στην διαδικτυακή εφαρμογή, δίνονται οδηγίες στον χρήστη, για το πως θα προγραμματίσει την πραγματική του συσκευή (ενσωματωμένο σύστημα), να στέλνει σε συγκεκριμένο api endpoint τις μετρήσεις που θα λαμβάνει, ώστε να υπάρχει ταυτοποίηση της πραγματικής συσκευής και της εικονικής (σταθμός στην εφαρμογή).

### 3.5 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό, δόθηκε αναλυτικά η περιγραφή του τρόπου σχεδίασης και ανάπτυξης του συστήματος. Παρουσιάστηκαν οι απαιτήσεις του συστήματος και σχεδιασμός της βάσης δεδομένων και όλων των πινάκων που την απαρτίζουν. Τέλος, αναλύθηκαν λεπτομερώς όλες οι τεχνικές που χρησιμοποιήθηκαν για την επίτευξη της ασφάλειας του συστήματος. Στο επόμενο κεφάλαιο ακολουθεί, η ανάλυση του λογισμικού μέρους του συστήματος.





## Κεφάλαιο 4ο – Λειτουργικότητα συστήματος

Όπως έχει ήδη αναφερθεί στην σύνοψη κεφαλαίων, το κεντρικό θέμα αυτού του κεφαλαίου αποτελεί το η λειτουργικότητα του συστήματος, με το οποίο και ολοκληρώνεται. Αναλύονται οι λειτουργίες και οι διεπαφές χρήστη της διαδικτυακής εφαρμογής, καθώς και οι περιπτώσεις χρήσης που παρουσιάζονται, ανάλογα την ιδιότητα του. Επίσης θα δοθεί μία περιγραφή για την συνολική λειτουργία της εφαρμογής κινητού, την ροή του προγράμματος που ακολουθεί, τις ενέργειες που πραγματοποιούνται από την πλευρά του χρήστη εθελοντή και οι περιπτώσεις χρήσης.

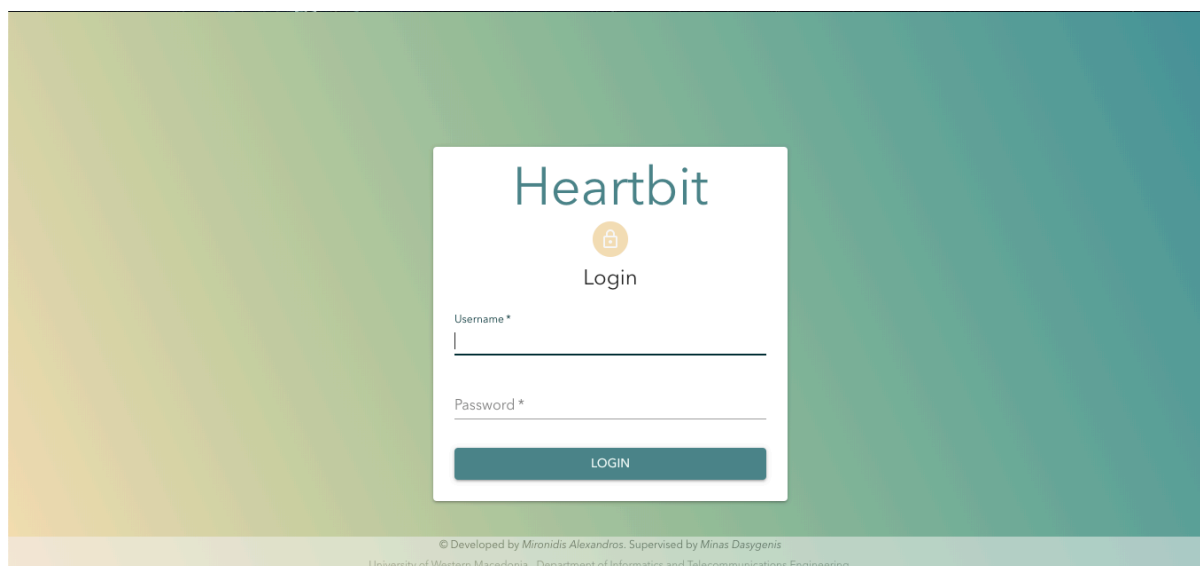
### 4.1 Ανάλυση λειτουργιών διαδικτυακής εφαρμογής και διεπαφές χρήστη

Ο σχεδιασμός της διεπαφής χρήστη σε ένα σύστημα αποτελεί ένα από τα βασικότερα στοιχεία που θα καθορίσουν την επιτυχία μίας εφαρμογής. Μία δύσχρηστη διεπαφή μπορεί να κάνει τον χρήστη να απορρίψει την εφαρμογή, παρά το γεγονός ότι αυτή μπορεί να λειτουργεί σωστά. Για αυτό τον λόγο η εφαρμογή παρέχει όλους τους απαραίτητους μηχανισμούς που θα συντελέσουν στην καλύτερη και ευκολότερη διαχείριση της από τους χρήστες. Παράλληλα με την διεπαφή, η εφαρμογή θα πρέπει να παρέχει όλες τις απαραίτητες λειτουργίες για τις οποίες προορίζεται, δίνοντας τον χρήστη πληθώρα επιλογών για ένα μεγάλο εύρος λειτουργιών. Στις υποενότητες που ακολουθούν παρουσιάζονται όλες οι διεπαφές χρήστη και οι λειτουργίες τους. Αξίζει να σημειωθούν και να αναλυθούν σε αυτό το σημείο ορισμένα στοιχεία της εφαρμογής, ώστε να μην αναφέρονται ξανά σε κάθε διεπαφή. Αρχικά, η εφαρμογή ακούει και πραγματοποιεί ότι αλλαγή συμβαίνει σε πραγματικό χρόνο χάρις στην βοήθεια του Pusher API. Αυτό πρακτικά σημαίνει πως ο διαχειριστής που βρίσκεται εντός της εφαρμογής, “ακούει” στο παρασκήνιο πιθανά συμβάντα (events) που μπορεί να λάβουν χώρα όσον αφορά τους απινιδωτές (π.χ. ξεκλείδωμα ενός απινιδωτή, απώλεια, πολύ χαμηλή ή υψηλή θερμοκρασία.). Έτσι με την πραγματοποίηση ενός συμβάντος, αυτομάτως γίνεται ανανέωση της διεπαφής χρήστη σε επιλεγμένα σημεία που επηρεάζονται από την ενέργεια. Έτσι ο χρήστης θα δει αυτομάτως την ανανεωμένη πληροφορία καθώς και μία ειδοποίηση και δεν θα χρειαστεί να πραγματοποιήσει ανακατεύθυνση της σελίδας. Ένα εξίσου σημαντικό στοιχείο αφορά την επικύρωση των δεδομένων (validation). Σε όλη την εφαρμογή τα δεδομένα που

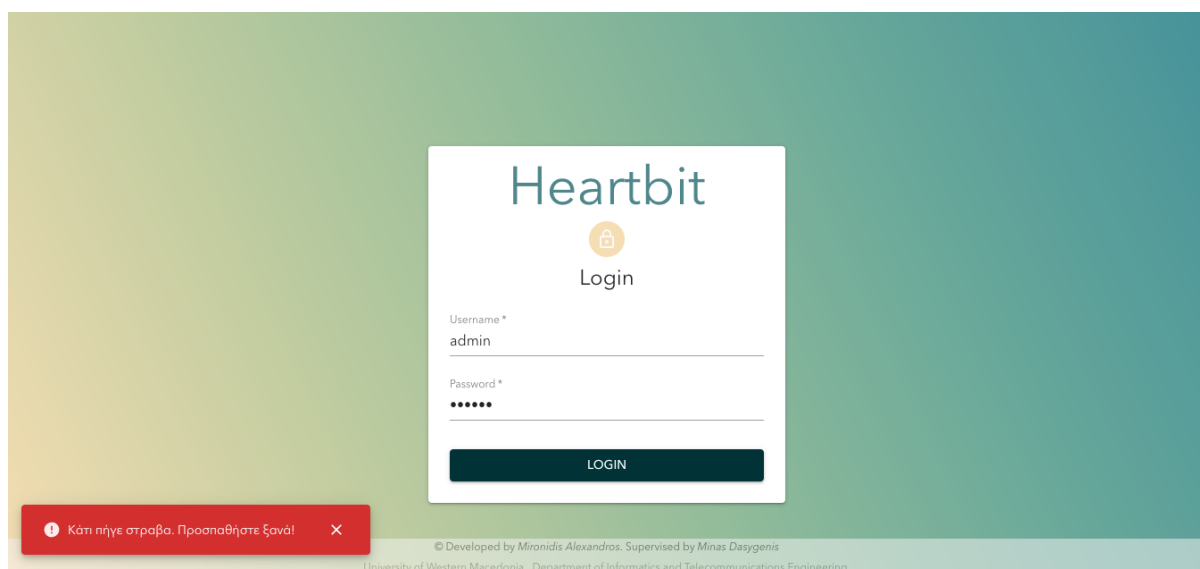
εισάγονται στις φόρμες, επικυρώνονται δύο φορές, μία στο frontend κομμάτι (React) και μία στο backend (server) μέσω της πιο γνωστής βιβλιοθήκης PHP για validation που ονομάζεται Respect Validator. Έτσι δεν υπάρχει τρόπος να προσπελαστούν και οι δύο οι έλεγχοι και να περαστούν δεδομένα που δεν ελέγχθηκαν. Επίσης, οι έλεγχοι που πραγματοποιούνται στο κομμάτι του server, είναι πολύ πιο αυστηροί και σε περίπτωση εύρεσης σφάλματος, ο server αποκρίνεται με το κατάλληλο μήνυμα στον χρήστη. Αυτή η υλοποίηση σχεδιάστηκε στοχευμένα έτσι διότι, υπάρχουν λογισμικά δημιουργίας http requests σε εφαρμογές και servers (για παράδειγμα το postman), παρακάμπτοντας το frontend κομμάτι της εφαρμογής. Τέλος, το Redux, αποτελεί το μέρος όπου αποθηκεύονται τα πιο σημαντικά δεδομένα κατά την σύνδεση του χρήστη με την εφαρμογή, όπως επίσης αποτελεί και το κέντρο πληροφοριών κάθε κατάστασης και διεπαφής. Αυτό που επιτυγχάνεται είναι, να παραμένει ο φόρτος εργασίας του server σε πολύ χαμηλό επίπεδο. Επίσης λόγω της συγκεκριμένης υλοποίησης, ο server στις περισσότερες αιτήσεις που δέχεται, επιστρέφει μία απλή πληροφορία και όχι πολύπλοκες μετατροπές δεδομένων από την βάση, ούτε αρχεία προβολής (view pages). Αυτό έχει σαν αποτέλεσμα, πολύ γρήγορη ανταπόκριση της εφαρμογής ακόμα και σε πιθανώς, μεγάλο εύρος ενεργών χρηστών.

#### 4.1.1 Αρχική σελίδα διαδικτυακής εφαρμογής

Για την επιτυχία μιας διαδικτυακής εφαρμογής, το πρώτο πράγμα που οφείλει λάβει υπόψιν του ο δημιουργός της είναι το κοινό στο οποίο απευθύνεται. Στην παρούσα διπλωματική εργασία η διαδικτυακή εφαρμογή που υλοποιήθηκε απευθύνεται σε εργαζόμενους του ΕΚΑΒ που πιθανόν δεν έχουν την απαραίτητη εμπειρία σε διαδικτυακές εφαρμογές, για αυτό ακριβώς το λόγο δόθηκε ιδιαίτερη έμφαση στην διεπαφή χρήστη (user interface) με σκοπό τέτοιο ώστε να γίνει η λειτουργία της εφαρμογής όσο το δυνατόν πιο εύκολη και να μπορεί να χειριστεί από τον καθένα. Έτσι λοιπόν ως αρχική σελίδα της εφαρμογής ορίστηκε η σελίδα σύνδεσης του διαχειριστή. Στην σελίδα σύνδεσης, συμπληρώνει το email καθώς και τον κωδικό του. Σε περίπτωση επιτυχημένης εξακρίβωσης στοιχείων, οδηγείται σε κεντρικό περιβάλλον της εφαρμογής για εξουσιοδοτημένους χρήστες. Σε περίπτωση σφάλματος δέχεται κατάλληλο μήνυμα.



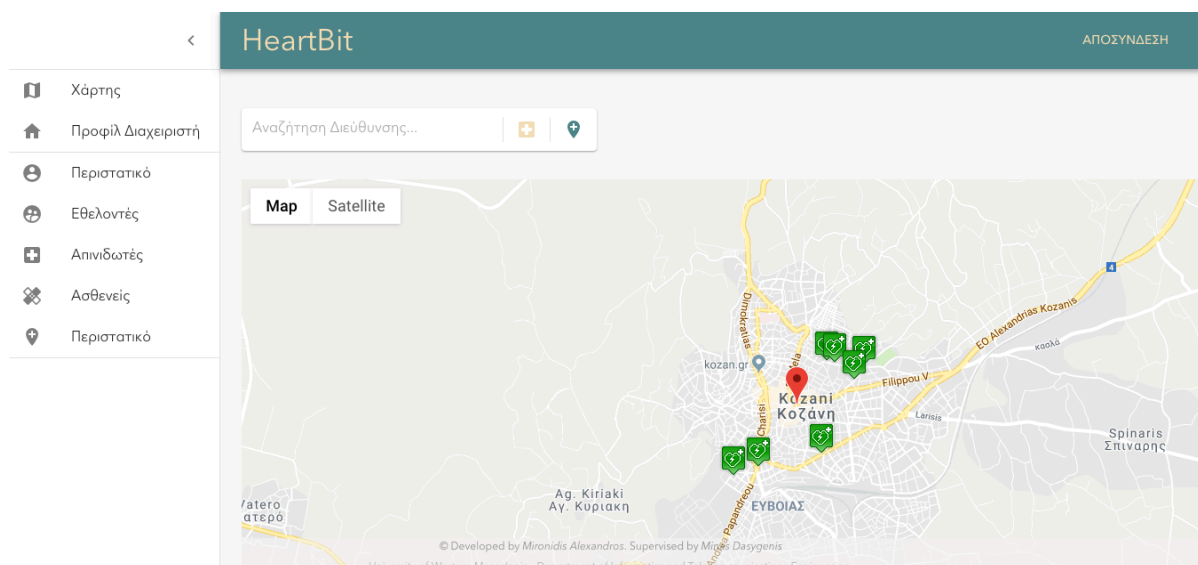
Εικόνα 18: Αρχική σελίδα – Σελίδα σύνδεσης



Εικόνα 19: Αρχική σελίδα – Μήνυμα εσφαλμένων στοιχείων σύνδεσης

#### 4.1.2 Κεντρικό περιβάλλον εξουσιοδοτημένου χρήστη (διαχειριστής)

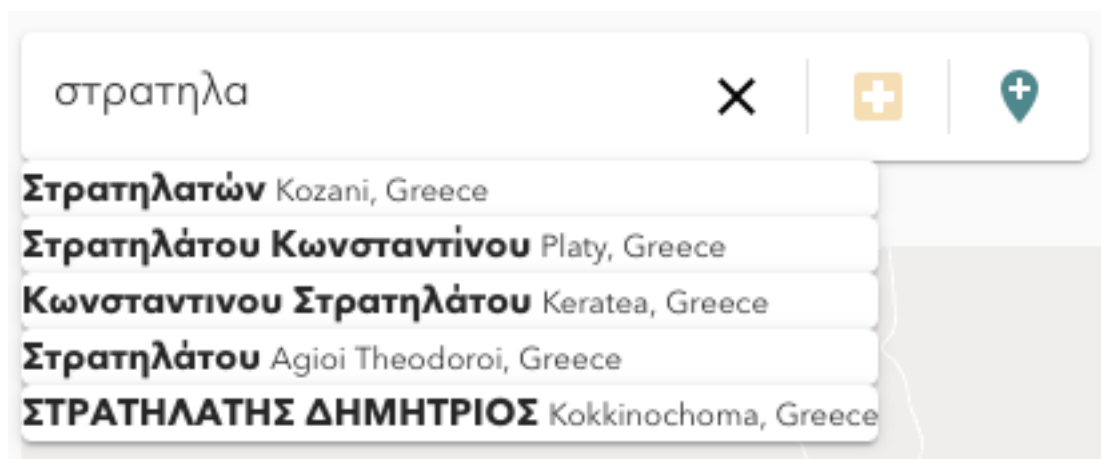
Ύστερα από επιτυχή σύνδεση ο χρήστης μεταφέρεται στην κεντρική σελίδα που έχει οριστεί η σελίδα χάρτη. Η κεντρική σελίδα όπως και κάθε άλλη σελίδα του εξουσιοδοτημένου χρήστη-διαχειριστή παρέχει τη δυνατότητα να πλοηγηθεί σε όλες τις σελίδες της εφαρμογής μέσω ενός αναδιπλούμενου drawer. Τέλος για την αποσύνδεση του χρήστη από την εφαρμογή, αρκεί το πάτημα του συνδέσμου αποσύνδεσης στην πάνω δεξιά γωνία της εφαρμογής. Έπειτα ο χρήστης οδηγείται στην αρχική σελίδα σύνδεσης.



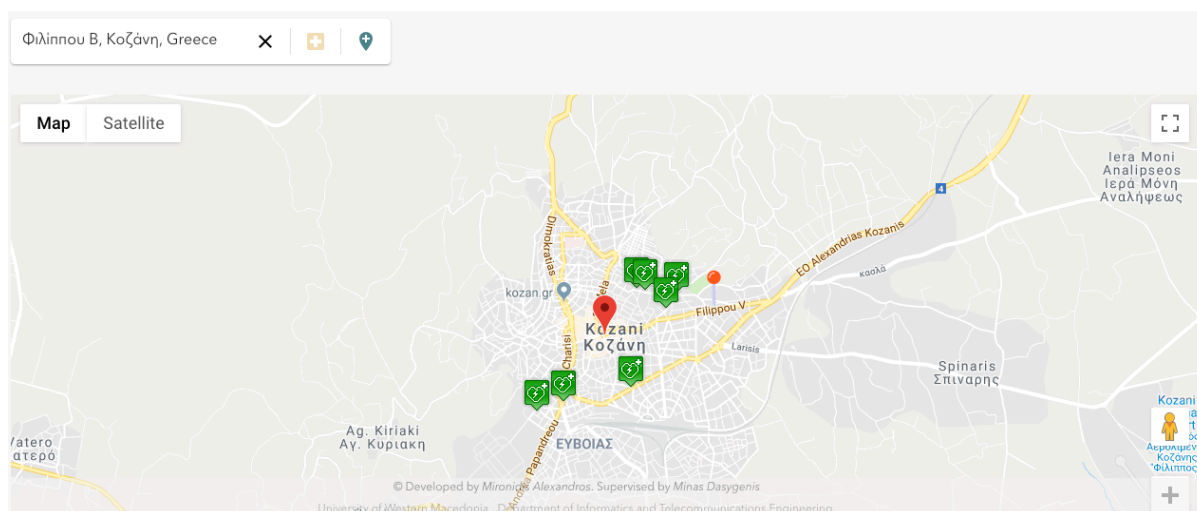
Εικόνα 20: Κεντρική σελίδα διαχειριστή – Drawer Πλοήγησης

### 4.1.3 Κεντρική Σελίδα Χάρτη

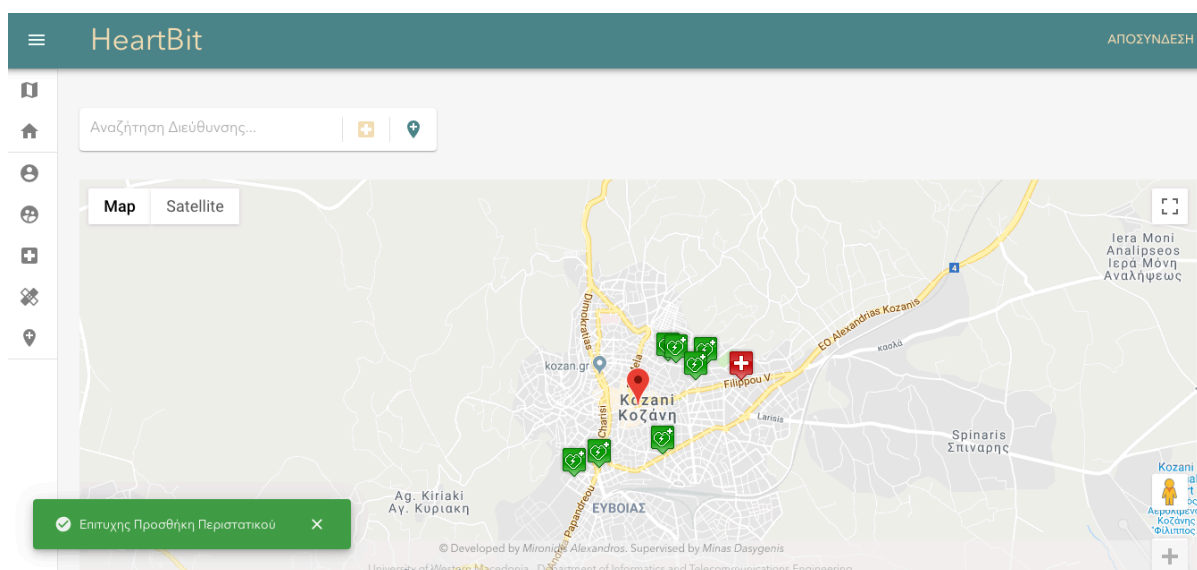
Η σελίδα του χάρτη η οποία και ορίστηκε ως κεντρική σελίδα της διαδικτυακής εφαρμογής είναι η πιο σημαντική καθώς η κύρια λειτουργικότητα της εφαρμογής προέρχεται από αυτήν. Σκοπός της συγκεκριμένης σελίδας είναι η γρήγορη και εύκολη προσθήκη περιστατικού για την ενημέρωση των εθελοντών. Η σελίδα του χάρτη διαθέτει ένα input αναζήτησης διεύθυνσης. Καθώς ο χρήστης πληκτρολογεί, κάτω από το input επεκτείνεται μια λίστα διευθύνσεων που αντιστοιχούν στην εισαγωγή του (autocomplete) με σκοπό την επιλογή της ζητούμενης διεύθυνσης. Σε περίπτωση λάθος διεύθυνσης, στην σελίδα αναγράφεται το αντίστοιχο μήνυμα. Ύστερα από επιτυχημένη επιλογή διεύθυνσης προστίθεται στον χάρτη ένα στίγμα (pin) στη συγκεκριμένη διεύθυνση. Ο χρήστης έπειτα έχει την δυνατότητα επιλογής εισαγωγής περιστατικού ή απινιδωτή στην διεύθυνση του pin πατώντας τα κουμπιά που περιέχει το input. Αν η προσθήκη οριστεί ως επιτυχημένη εμφανίζεται αντίστοιχο μήνυμα. Παρουσιάζεται αναλυτικά σε εικόνες η όλη διαδικασία.



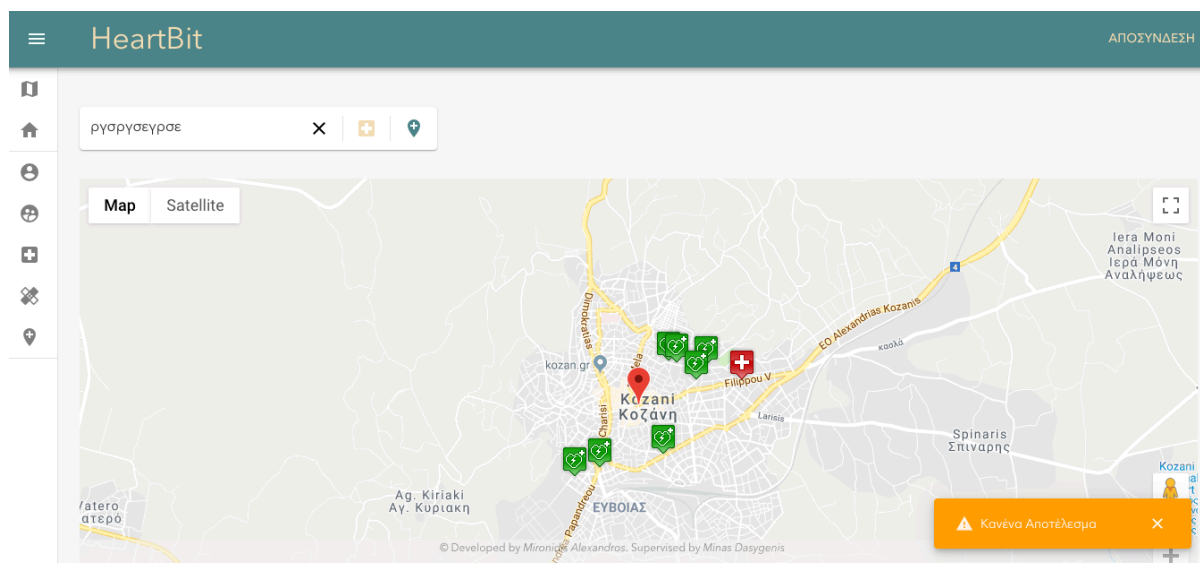
Εικόνα 21: Σελίδα Χάρτη – AutoComplete Αναζήτησης διεύθυνσεων



Εικόνα 22: Σελίδα Χάρτη – Προσθήκη pin στην απαιτούμενη διεύθυνση



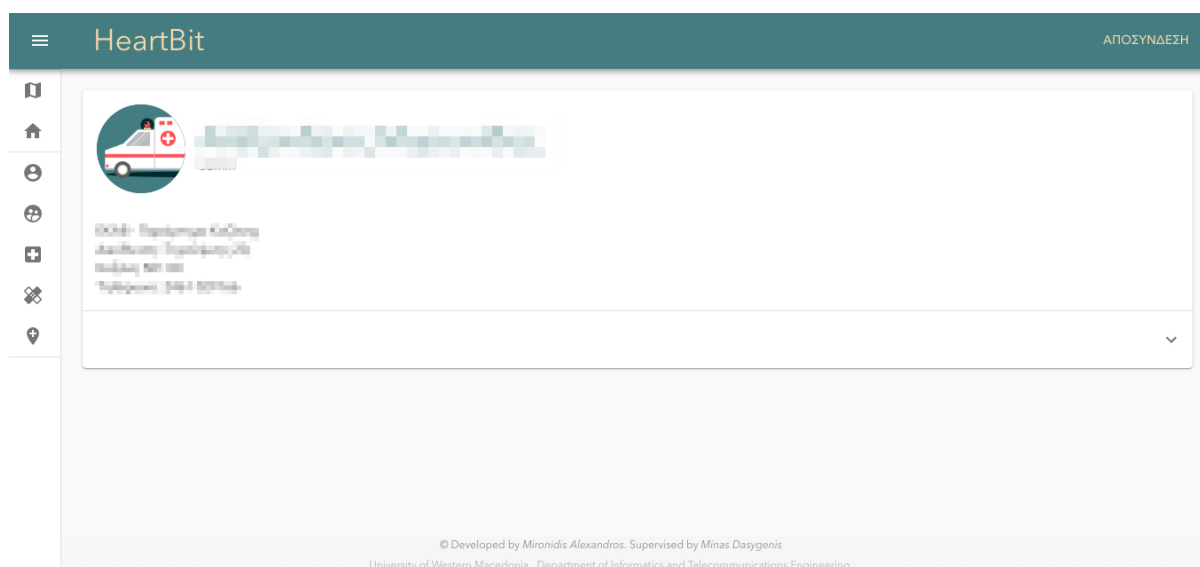
Εικόνα 23: Σελίδα Χάρτη – Προσθήκη περιστατικού με το αντίστοιχο μήνυμα



Εικόνα 24: Σελίδα Χάρτη – Παράδειγμα συμπλήρωσης λάθος διεύθυνσης και αντίστοιχο μήνυμα

#### 4.1.4 Σελίδα Προφίλ Διαχειριστή

Η σελίδα του Προφίλ διαχειριστή περιέχει πληροφορίες σχετικά με το παράρτημα ΕΚΑΒ της περιοχής του, καθώς και τις πληροφορίες του εκάστοτε συνδεδεμένου διαχειριστή όπως τη διεύθυνση, το email ή το τηλέφωνο του. Η μοναδική λειτουργικότητα αυτής της σελίδας είναι η αποστολή μηνύματος από την διαχειριστή στους εθελοντές σε πραγματικό χρόνο μέσω ενός input. Ακολουθούν παραδείγματα σε εικόνες.



Εικόνα 25: Σελίδα Προφίλ Διαχειριστή

Στοιχεία Διαχειριστή:

Τηλέφωνο επικοινωνίας

email

Διεύθυνση

Πληκτρολογήστε Μήνυμα  
 ➤

Εικόνα 26: Σελίδα Προφίλ Διαχειριστή – Φόρμα Διαχειριστή

#### 4.1.5 Σελίδα Διαχειριστών

Στη συγκεκριμένη σελίδα υπάρχει ο πίνακας με το πλήθος των διαχειριστών της διαδικτυακής εφαρμογής. Η συγκεκριμένη σελίδα περιλαμβάνει λειτουργίες οι οποίες αναλύονται στη συνέχεια.

Τύπος	Όνομα	Επώνυμο	E-mail	Διεύθυνση	Όνομα Χρήστη	Κωδικός	Επεξεργασία Διαχειριστή
admin	Alexis	mrds	alexmr1@hotmail.com	Κοζανή	alexmr1	123456789	
admin	Alexis	mrds	alexmr1@gmail.com	stratlatwn 10	alexmr1	123456789	
admin	Gimli	Dwarf	GimliDwarg@lotr.com	Erebor	gimli	123456789	
admin	Legolas	Elfe	LegolasElf@lotr.com	Minas Tirith	legolas	123456789	
admin	harry	potter	HarryPotter@hogwarts.com	Privet	harrypotter	123456789	

Εικόνα 27: Σελίδα Διαχειριστών

Ο πίνακας των διαχειριστών έχει τη δυνατότητα προβολής 5, 10 ή 15 καταχωρήσεων την οποία την επιλέγει ο χρήστης (pagination) όπως φαίνεται στην εικόνα 26.



Εικόνα 28: Σελίδα Πίνακα Διαχειριστών – Σελιδοποίηση

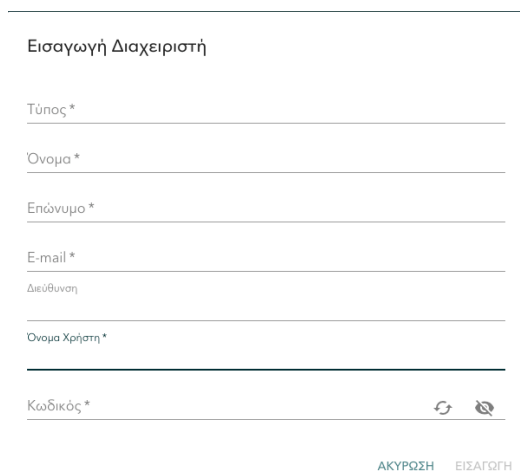
Για την προσθήκη νέου διαχειριστή, υπάρχει κουμπί προσθήκης στη δεξιά γωνία της οθόνης. Το κουμπί προσθήκης εμφανίζει επίσης παράλληλο μήνυμα ειδοποίησης ειδικού χρωματισμού (tooltip) όταν ο κέρσορας του ποντικιού βρίσκεται πάνω από το κουμπί. Ακολουθεί παράδειγμα στην εικόνα 27.



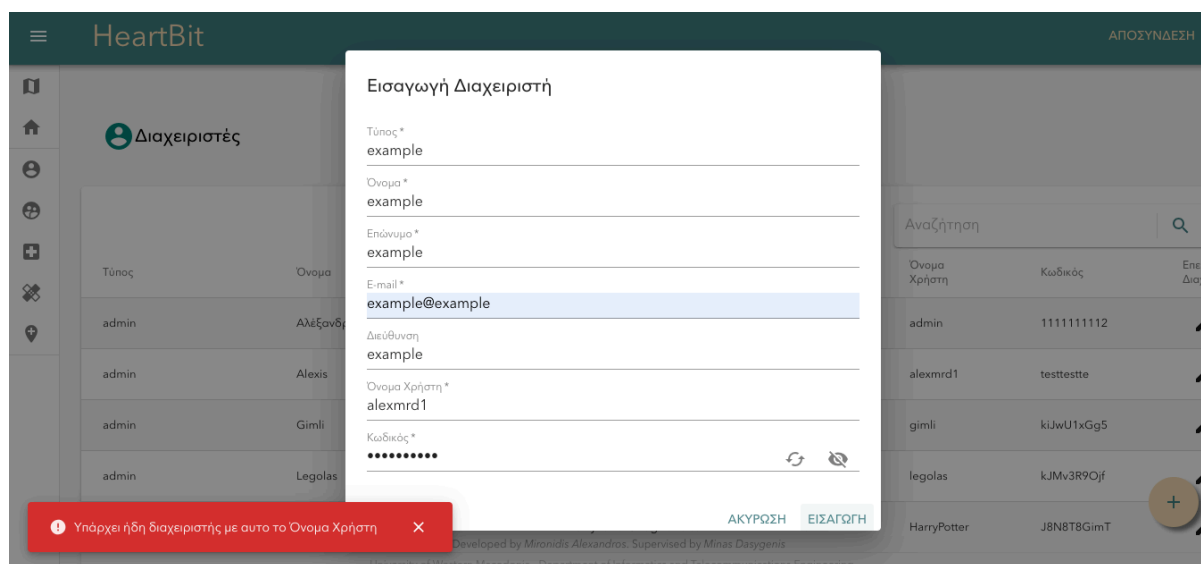
Εικόνα 29: Σελίδα Πίνακα Διαχειριστών – Κουμπί προσθήκης διαχειριστή & tooltip

Ύστερα από την πληκτρολόγηση του κουμπιού προσθήκης εμφανίζεται μια φόρμα εισαγωγής διαχειριστή με πεδία, τα στοιχεία που διαθέτει ένας διαχειριστής. Το κουμπί “ΕΙΣΑΓΩΓΗ”, αρχικά είναι ανενεργό μέχρι να αρχίσει ο χρήστης να πληκτρολογεί σε ένα από τα πεδία. Τα πεδία τα οποία έχουν αστερίσκο, υποδηλώνουν ότι η συμπλήρωσή τους είναι απαραίτητη. Σε περίπτωση μη εισαγωγής ενός από αυτά τα πεδία υπάρχει client side validation το οποίο δεν επιτρέπει την υποβολή της φόρμας και εμφανίζει αντίστοιχο μήνυμα. Υπάρχει επίσης server side validation το οποίο ελέγχει αν τα στοιχεία όπως το “Όνομα Χρήστη” και το “E-mail” είναι μοναδικά και έχουν την σωστή μορφή. Σε περίπτωση που δεν είναι εμφανίζεται αντίστοιχο μήνυμα.






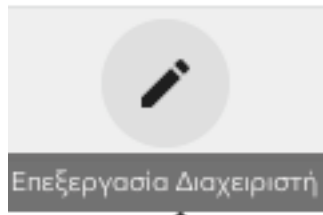


Εικόνα 30: Σελίδα Πίνακα Διαχειριστών – Καρτέλα Εισαγωγής Διαχειριστή



Εικόνα 31: Σελίδα Πίνακα Διαχειριστών – Μήνυμα σφάλματος στην εισαγωγή διαχειριστή (validation)

Στο πεδίο εισαγωγής κωδικού, υπάρχουν δυο κουμπιά τα οποία έχουν την εξής λειτουργικότητα. Το πρώτο κουμπί  αναπαράγει τυχαίους κωδικούς δέκα χαρακτήρων τόσο γραμμάτων όσο και αριθμών. Η λειτουργικότητα του δεύτερου κουμπιού  δίνει τη δυνατότητα στο χρήστη να αλλάζει την καταχώρηση κωδικού από μη ορατή σε ορατή. Σε περίπτωση εισαγωγής κωδικού με την καταχώρηση ορατή το εικονίδιο εμφανίζεται έτσι . Ο πίνακας των διαχειριστών παρέχει επίσης την δυνατότητα επεξεργασίας ενός διαχειριστή. Η τελευταία στήλη του πίνακα περιέχει το εικονίδιο ενός μολυβιού στην εικόνα 30.



Εικόνα 32: Σελίδα Πίνακα Διαχειριστών – Εικονίδιο Επεξεργασίας Διαχειριστή & Tooltip

Με το click στο συγκεκριμένο εικονίδιο εμφανίζεται μια αντίστοιχη φόρμα όπως την φόρμα εισαγωγής, με την διαφορά ότι η συγκεκριμένη φόρμα περιλαμβάνει όλα τα στοιχεία του εκάστοτε διαχειριστή που υπάρχει στη συγκεκριμένη σειρά του πίνακα όπου πατήθηκε το κουμπί. Το κουμπί “ΕΝΗΜΕΡΩΣΗ” το οποίο υποβάλλει τα νέα στοιχεία στη βάση είναι ανενεργό έως ότου ο διαχειριστής να αλλάξει ένα την καταχώρηση σε ένα πεδίο. Όπως στην φόρμα εισαγωγής έτσι και η φόρμα ενημέρωσης διαθέτει validation τόσο σε client side όσο και σε server side και προβάλλει αντίστοιχα μηνύματα με τη μορφή ειδοποίησης. Τόσο η εισαγωγή όσο και η καταχώρηση χρήστη γίνονται ασύγχρονα και γίνεται εμφανής η αλλαγή ή η προσθήκη αντίστοιχα καθώς όλες οι διαδικασίες γίνονται ασύγχρονα.

**Επεξεργασία Διαχειριστή**

Τύπος \*  
admin

---

Όνομα \*  
Απόστολος

---

Επώνυμο \*  
Μπακογιάννης

---

E-mail  
apostol@barbaros.com

---

Διεύθυνση  
Παύση

---

Όνομα Χρήστη  
admin

---

Κωδικός  
●●●●●●●●

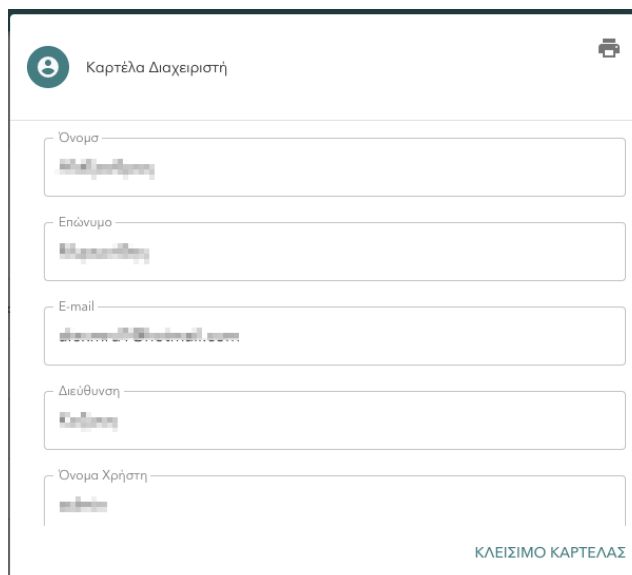
↻ ✖

ΑΚΥΡΩΣΗ ΕΝΗΜΕΡΩΣΗ

Εικόνα 33: Σελίδα Πίνακα Διαχειριστών – Παράδειγμα Επεξεργασίας Διαχειριστή

Μετακινώντας τον κέρσορα του ποντικιού στον πίνακα των διαχειριστών θα παρατηρήσει κανείς ότι η σειρά του πίνακα που υπάρχει κάτω από τον κέρσορα μεταβάλλει το χρώμα της σε ελαφρώς πιο σκούρες αποχρώσεις του γκρι. Αυτή η μεταβολή του χρώματος σκοπό έχει να κάνει κατανοητό στον χρήστη ότι κάθε σειρά είναι clickable, έχει δηλαδή τη

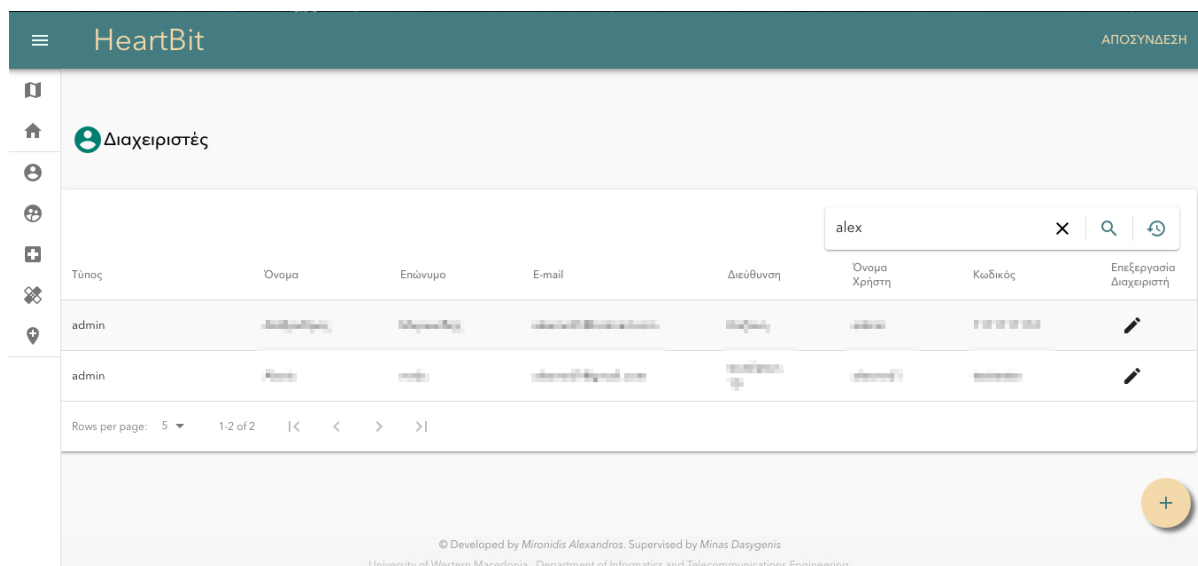
δυνατότητα να γίνει κλικ. Με κλικ σε οποιαδήποτε σειρά εμφανίζεται μια καρτέλα η οποία περιλαμβάνει το σύνολο των στοιχείων του διαχειριστή της συγκεκριμένης σειράς του πίνακα. Η συγκεκριμένη καρτέλα περιλαμβάνει στη πάνω δεξιά γωνία της ένα κουμπί με το εικονίδιο ενός εκτυπωτή. Με το κλικ σε αυτό το εικονίδιο ο χρήστης είναι ικανός να εκτυπώσει την συγκεκριμένη καρτέλα. Επίσης η καρτέλα δεν παρέχει την δυνατότητα επεξεργασίας.



The image shows a user management card titled "Καρτέλα Διαχειριστή". It features a circular icon with a person silhouette on the top left and a print icon on the top right. Below the title, there are five input fields with labels: "Όνομα" (Name) with value "Μιχάλης", "Επώνυμο" (Surname) with value "Μιχαηλίδης", "E-mail" with value "michail@bluemail.com", "Διεύθυνση" (Address) with value "Καθήκη", and "Όνομα Χρήστη" (Username) with value "michail". At the bottom right of the card, there is a button labeled "ΚΛΕΙΣΙΜΟ ΚΑΡΤΕΛΑΣ".

Εικόνα 34: Σελίδα Πίνακα Διαχειριστών – Καρτέλα Διαχειριστή

Τέλος, ο πίνακας των διαχειριστών περιλαμβάνει στην πρώτη του σειρά ένα input αναζήτησης στο πλήθος όλων των καταχωρήσεων των διαχειριστών. Για την αναζήτηση μιας καταχώρησης αρκεί το κλικ στο κουμπί 🔍, ενώ για την επαναφορά του πίνακα στην αρχική του κατάσταση το κλικ στο κουμπί ↺. Ακολουθεί παράδειγμα αναζήτησης στην εικόνα 31.



Εικόνα 35: σελίδα Πίνακα Διαχειριστών – Παράδειγμα Αναζήτησης Διαχειριστή

#### 4.1.6 Σελίδα Εθελοντών

Στη συγκεκριμένη σελίδα υπάρχει ο πίνακας ο οποίος περιλαμβάνει το πλήθος των εθελοντών που έχουν λογαριασμό στην εφαρμογή κινητού. Η δημιουργία ενός νέου εθελοντή γίνεται από τον διαχειριστή της διαδικτυακής εφαρμογής. Επίσης ο διαχειριστής έχει την δυνατότητα να επεξεργαστεί τα στοιχεία των εθελοντών. Όπως στην προηγούμενο πίνακα των διαχειριστών, έτσι και σε αυτόν υπάρχει αναζήτηση στο πλήθος των εθελοντών καθώς και εκτυπώσιμη καρτέλα στοιχείων του καθένα ξεχωριστά. Λόγου το ότι ο πίνακας των εθελοντών έχει ως προς το πλείστον κοινή λειτουργικότητα με των πίνακα διαχειριστών θα γίνει εκτενής αναφορά με παραδείγματα μόνο στις ιδιαιτερότητές του. Η ιδιαιτερότητα που περιέχει ο συγκεκριμένος πίνακας σε σχέση με τον πίνακα των διαχειριστών είναι η στήλη “Κατάσταση Δραστηριότητας”, η οποία δίνει την δυνατότητα στον διαχειριστή να απενεργοποιήσει ή να ενεργοποιήσει τα δικαιώματα ενός χρήστη. Στις επόμενες εικόνες ακολουθούν παραδείγματα.

Όνομα	Επώνυμο	Email	Ημερομηνία Γέννησης	Περιφέρεια	Κατάσταση Δραστηριότητας	Επεξεργασία
Monica	Geller	monicageller@monicageller.com	1963-12-12	Θεσσαλία	Ενεργός	✎
ross	gellerr	rossgeller@rossgeller.com	1950-03-23	Αττική	Ενεργός	✎
phoebe	bouffee	phoebouffee@phoebouffee.com	1960-12-12	Θεσσαλία	Ενεργός	✎
joey	tribbiani	joeytribbiani@tribbiani.com	1950-12-12	Ηπειρος	Ενεργός	✎

© Developed by Mironidis Alexandros. Supervised by Minas Dasygenis  
University of Western Macedonia, Department of Informatics and Telecommunications Engineering.

Εικόνα 36: Σελίδα Εθελοντών

Όνομα	Επώνυμο	Email	Ημερομηνία Γέννησης	Περιφέρεια	Κατάσταση Δραστηριότητας	Επεξεργασία
Monica	Geller	monicageller@monicageller.com	1963-12-12	Θεσσαλία	Ενεργός	✎
ross	gellerr	rossgeller@rossgeller.com	1950-03-23	Αττική	Ενεργός	✎
phoebe	bouffee	phoebouffee@phoebouffee.com	1960-12-12	Θεσσαλία	Ενεργός	✎
joey	tribbiani	joeytribbiani@tribbiani.com	1950-12-12	Ηπειρος	Εθελοντής Απενεργοποιημένος	✎

© Developed by Mironidis Alexandros. Supervised by Minas Dasygenis  
University of Western Macedonia, Department of Informatics and Telecommunications Engineering.

Εικόνα 37: Σελίδα Πίνακα Εθελοντών - Παράδειγμα Ενεργοποίησης Δικαιωμάτων Εθελοντή

Όνομα	Επώνυμο	Email	Ημερομηνία Γέννησης	Περιφέρεια	Κατάσταση Δραστηριότητας	Επεξεργασία
joey	tribbiani		1950-12-12	Ήπειρος		
phoebe	boufee	phoeboufee@phoeboufee.com	1960-12-12	Θεσσαλία		
ross	geller	rossgeller@rossgeller.com	1950-03-23	Αττική		

© Developed by Mironidis Alexandros. Supervised by Minas Dasygenis  
 mironidageller@mironidageller.com 1963-12-12  
 University of Western Macedonia, Department of Informatics and Telecommunications Engineering.

Εικόνα 38: Σελίδα Πίνακα Εθελοντών - Παράδειγμα Απενεργοποίησης Δικαιωμάτων Εθελοντή

#### 4.1.7 Σελίδα Απινιδωτών





Ακολουθεί η σελίδα των απινιδωτών, η οποία περιέχει επίσης έναν πίνακα με το σύνολο των εγκατεστημένων απινιδωτών καθώς και τα στοιχεία αυτών. Στοιχεία όπως η το μοντέλο του απινιδωτή, η διεύθυνση του, η ημερομηνία εγκατάστασης και η ημερομηνία τελευταίας αναβάθμισης του καθώς και αν είναι ξεκλειδωτος ή όχι ή αν λείπει από τη θέση του. Η ιδιαιτερότητα του συγκεκριμένου πίνακα είναι ότι η προσθήκη νέου απινιδωτή είναι εφικτή μόνο από τη σελίδα του χάρτη, καθώς είναι απαραίτητη η ακριβής διεύθυνση στην οποία έχει εγκατασταθεί νέος απινιδωτής. Η επεξεργασία των στοιχείων του εκάστοτε απινιδωτή παρόλα αυτά γίνεται όπως και στις προηγούμενες δύο σελίδες που αναφέρθηκαν.

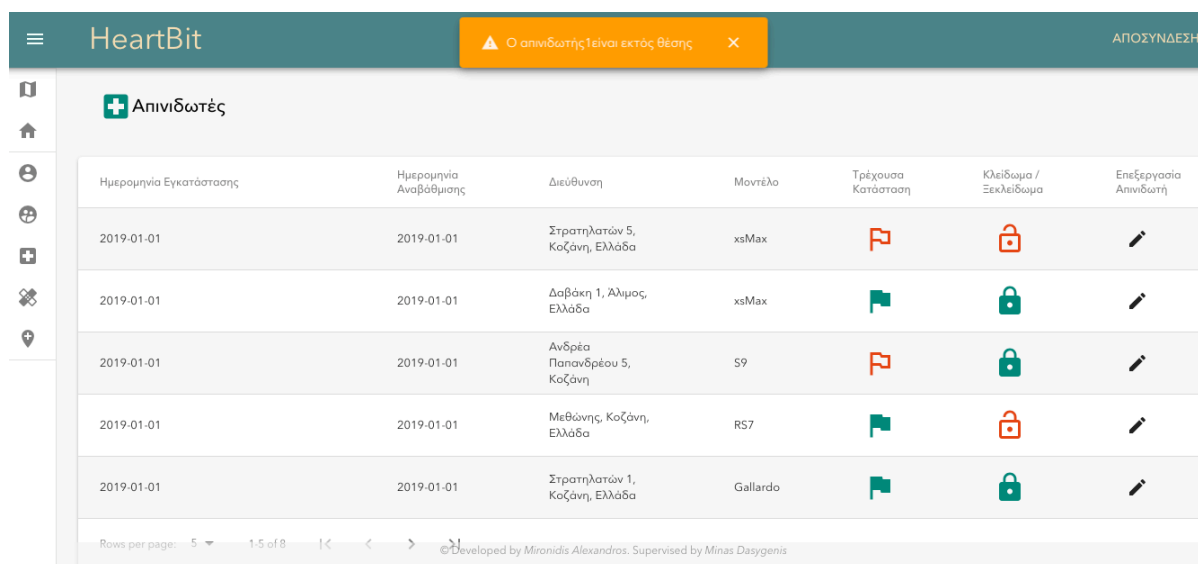
Ημερομηνία Εγκατάστασης	Ημερομηνία Αναβάθμισης	Διεύθυνση	Μοντέλο	Τρέχουσα Κατάσταση	Κλειδωμο / Ξεκλειδωμο	Επεξεργασία Απινιδωτή
2019-01-01	2019-01-01	Στρατηλατών 5, Κοζάνη, Ελλάδα	xsMax			
2019-01-01	2019-01-01	Δαβάκη 1, Αλιμος, Ελλάδα	xsMax			
2019-01-01	2019-01-01	Ανδρέα Παπανδρέου 5, Κοζάνη	S9			
2019-01-01	2019-01-01	Μεθώνης, Κοζάνη, Ελλάδα	RS7			
2019-01-01	2019-01-01	Στρατηλατών 1, Κοζάνη, Ελλάδα	Gallardo			
















© Developed by Mironidis Alexandros. Supervised by Minas Dasygenis  
 University of Western Macedonia, Department of Informatics and Telecommunications Engineering.

Εικόνα 39: Σελίδα Απινιδωτών

Εικόνα 36: Σελίδα Απινιδωτών

Η ιδιαιτερότητα του πίνακα της συγκεκριμένης σελίδας είναι ότι κάποια στοιχεία της είναι πιθανόν να αλλάξουν αυτόματα από αντίστοιχο αίτημα από Arduino που έχει τοποθετηθεί σε κάθε απινιδωτή ξεχωριστά. Τα στοιχεία που είναι ικανά να αλλάξουν είναι η στήλη “Τρέχουσα Κατάσταση” και η στήλη “Κλειδωμα/Ξεκλείδωμα”. Η στήλη “Τρέχουσα Κατάσταση” υποδηλώνει το αν ο απινιδωτής βρίσκεται στη θέση του με εικονίδιο την πράσινη σημαία  ή όχι με εικονίδιο την κόκκινη σημαία . Αντίστοιχα η στήλη “Κλειδωμα/Ξεκλείδωμα” υποδηλώνει αν το κελί το απινιδωτή είναι κλειδωμένο με εικονίδιο την πράσινη κλειδαριά  ή αν είναι ξεκλειδωτο με εικονίδιο την ξεκλειδωτη κόκκινη κλειδαριά . Σε περίπτωση που κάποια από αυτές τις καταχωρήσεις αλλάξει αυτόματα από αίτημα του Arduino στον Server, τότε εμφανίζεται στον διαχειριστή αντίστοιχο μήνυμα σε όποια καρτέλα και αν βρίσκεται, όπως φαίνεται στα παρακάτω παραδείγματα.



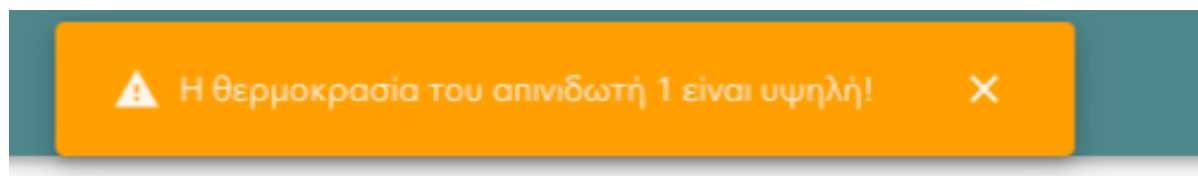
Ημερομηνία Εγκατάστασης	Ημερομηνία Αναβάθμισης	Διεύθυνση	Μοντέλο	Τρέχουσα Κατάσταση	Κλειδωμα / Ξεκλείδωμα	Επεξεργασία Απινιδωτή
2019-01-01	2019-01-01	Στρατηλατών 5, Κοζάνη, Ελλάδα	xsMax			
2019-01-01	2019-01-01	Δαβάκη 1, Άλιμος, Ελλάδα	xsMax			
2019-01-01	2019-01-01	Ανδρέα Παπανδρέου 5, Κοζάνη	S9			
2019-01-01	2019-01-01	Μεθώνης, Κοζάνη, Ελλάδα	RS7			
2019-01-01	2019-01-01	Στρατηλατών 1, Κοζάνη, Ελλάδα	Gallardo			

Εικόνα 40 :Παράδειγμα απώλειας ενός απινιδωτή από το και αλλαγή καταχώρησης από Arduino

Ημερομηνία Εγκατάστασης	Ημερομηνία Αναβάθμισης	Διεύθυνση	Μοντέλο	Τρέχουσα Κατάσταση	Κλειδωμο / Ξεκλειδωμο	Επεξεργασία Απινιδωτή
2019-01-01	2019-01-01	Στρατηλατών 5, Κοζάνη, Ελλάδα	xsMax	🚩	🔒	✎
2019-01-01	2019-01-01	Δαβάκη 1, Αλιμος, Ελλάδα	xsMax	🚩	🔒	✎
2019-01-01	2019-01-01	Ανδρέα Παπανδρέου 5, Κοζάνη	S9	🚩	🔒	✎
2019-01-01	2019-01-01	Μεθώνης, Κοζάνη, Ελλάδα	RS7	🚩	🔒	✎
2019-01-01	2019-01-01	Στρατηλατών 1, Κοζάνη, Ελλάδα	Gallardo	🚩	🔒	✎

Εικόνα 41: Παράδειγμα κλειδώματος ενός απινιδωτή από το και αλλαγή καταχώρησης από Arduino

Το Arduino επίσης έχει την δυνατότητα να ελέγχει και άλλες μετρήσεις και να ενημερώνει κατάλληλα τον χειριστή σε πραγματικό χρόνο. Μια από αυτές είναι και η θερμοκρασία των απινιδωτών. Σε περίπτωση που θα παρουσιαστεί μεγάλη μεταβολή της θερμοκρασίας κάποιου απινιδωτή, τότε θα εμφανιστεί αντίστοιχο μήνυμα που θα ενημερώνει τον χρήστη. Ακολουθεί παράδειγμα.



Εικόνα 42: Παράδειγμα μηνύματος υψηλής θερμοκρασίας ενός απινιδωτή



Εικόνα 43: Παράδειγμα μηνύματος χαμηλής θερμοκρασίας ενός απινιδωτή



### 4.1.7 Σελίδα Ασθενών

Ακολουθεί η σελίδα των ασθενών που έχουν καταγραφεί στο σύστημα. Στη συγκεκριμένη σελίδα το σύνολο των ασθενών καταρτίζει έναν πίνακα και ως στήλες έχει σημαντικά στοιχεία που πρέπει το ΕΚΑΒ να είναι ενήμερο. Τα στοιχεία αυτά είναι το Όνομα, το Επώνυμο και το Γένος. Επιπρόσθετα μη απαιτούμενα πεδία είναι το Έτος Γέννησης, η Διεύθυνση, το Ιστορικό και η Περιγραφή του ασθενή. Τα στοιχεία αυτά είναι κυρίως σημαντικά για στατιστικές μελέτες από το διαχειριστικό κομμάτι του ΕΚΑΒ. Η προσθήκη και η επεξεργασία ενός ασθενή γίνονται με τον όποιο και στις προηγούμενες σελίδες της διαδικτυακής εφαρμογής για αυτό τον λόγο δεν θα αναλυθούν εκτενέστερα με παραδείγματα.

Όνομα	Επώνυμο	Διεύθυνση	Έτος Γέννησης	Ιστορικό	Περιγραφή	Φύλο	Επεξεργασία Ασθενή
...	...	...	1960			...	✎
...	...	...	1997		1993	...	✎
...	...	...	1995		1967	...	✎
...	...	...	1999			...	✎
...	...	...	1999			...	✎



Rows per page: 5 | 1-5 of 14 | < > >> <<

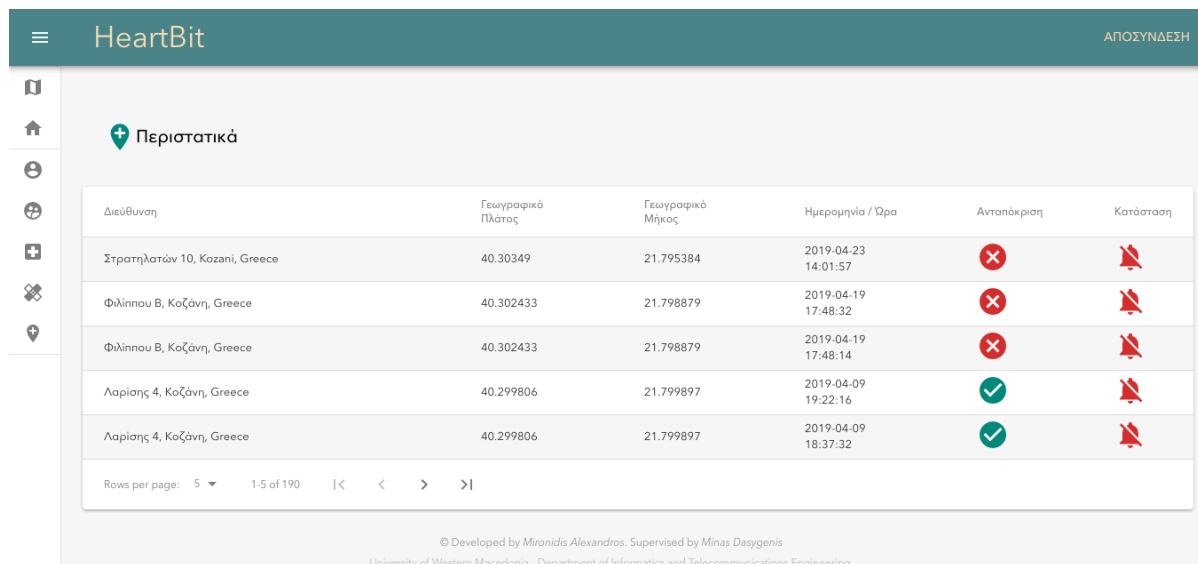
© Developed by Mironidis Alexandros. Supervised by Minas Dasygenis  
University of Western Macedonia, Department of Informatics and Telecommunications Engineering.











Εικόνα 44: Σελίδα Ασθενών

### 4.1.8 Σελίδα Περιστατικών

Τέλος, υπάρχει και η σελίδα των Περιστατικών. Στη συγκεκριμένη σελίδα υπάρχει ένας πίνακας με το πλήθος των περιστατικών με σημαντικά στοιχεία. Αυτά τα στοιχεία είναι η ακριβής Διεύθυνση του περιστατικού μαζί με το Γεωγραφικό Μήκος και Πλάτος της, η Ακριβής Ημερομηνία και Ώρα που συντελέστηκε το περιστατικό, αν υπήρξε Ανταπόκριση και ποια η ακριβής της ώρα, καθώς και η κατάσταση δραστηριότητας του περιστατικού. Πιο αναλυτικά, μετά την επιτυχή προσθήκη ενός περιστατικού στη σελίδα του Χάρτη (με την διαδικασία που αναλύθηκε σε προηγούμενη ενότητα), το περιστατικό προστίθεται αυτόματα

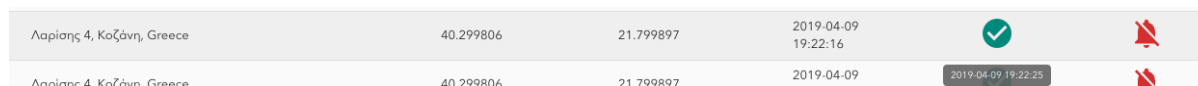
στον πίνακα αυτής της σελίδας. Η ανταπόκριση αρχικά ορίζεται ως απορριπτέα και εμφανίζει το εξής εικονίδιο , έως ότου ανταποκριθεί ένας ασθενής από την εφαρμογή του κινητού του. Ύστερα από επιτυχή ανταπόκριση, ενημερώνεται αυτόματα σε πραγματικό χρόνο η διαδικτυακή εφαρμογή και το εικονίδιο αλλάζει ως εξής  .







Διεύθυνση	Γεωγραφικό Πλάτος	Γεωγραφικό Μήκος	Ημερομηνία / Ώρα	Ανταπόκριση	Κατάσταση
Στρατηλατών 10, Κοζάνη, Greece	40.30349	21.795384	2019-04-23 14:01:57		
Φιλίππου Β, Κοζάνη, Greece	40.302433	21.798879	2019-04-19 17:48:32		
Φιλίππου Β, Κοζάνη, Greece	40.302433	21.798879	2019-04-19 17:48:14		
Λαρίσης 4, Κοζάνη, Greece	40.299806	21.799897	2019-04-09 19:22:16		
Λαρίσης 4, Κοζάνη, Greece	40.299806	21.799897	2019-04-09 18:37:32		



Εικόνα 45: Σελίδα Περιστατικών

Την ακριβή ώρα ανταπόκρισης μπορεί να την δει κανείς μετακινώντας τον κέρσορα του ποντικιού στο εικονίδιο ανταπόκρισης του απαιτούμενου περιστατικού καθώς εμφανίζεται με tooltip, όπως φαίνεται στην εικόνα 39.



Λαρίσης 4, Κοζάνη, Greece	40.299806	21.799897	2019-04-09 19:22:16		
Λαρίσης 4, Κοζάνη, Greece	40.299806	21.799897	2019-04-09 18:37:32		

Εικόνα 46: Σελίδα Περιστατικών – Παράδειγμα εμφάνισης ώρας ανταπόκρισης

Η στήλη της κατάστασης δραστηριότητας του περιστατικού ορίζεται αρχικά, μετά την επιτυχή προσθήκη του, ως ενεργή με το εξής εικονίδιο , και με το πέρας εικοσιτεσσάρων ωρών τίθεται αυτόματα ως ανενεργό και το εικονίδιο μεταβάλλεται ως εξής  .

Τέλος για την αποσύνδεση του χρήστη από την εφαρμογή, αρκεί το πάτημα του συνδέσμου αποσύνδεσης στην πάνω δεξιά γωνία της εφαρμογής σε οποιαδήποτε σελίδα και αν βρίσκεται. Έπειτα ο χρήστης οδηγείται στην αρχική κεντρική σελίδα, που έχει οριστεί η σελίδα σύνδεσης.

Εικόνα 47: Κουμπί αποσύνδεσης χρήστη

## 4.2 Ανάλυση λειτουργιών εφαρμογής κινητού και διεπαφές χρήστη

Όπως στο πλαίσιο της διαδικτυακής εφαρμογής, έτσι και στην εφαρμογή που αναπτύχθηκε για τα κινητά τηλέφωνα, δόθηκε μεγάλη σημασία στον σχεδιασμό διεπαφής χρήστη. Στόχος της είναι να προσελκύσει νέους χρήστες, και να διευκολύνει τις λειτουργίες στους ήδη εγγεγραμμένους χρήστες της εφαρμογής. Σε αντίθεση με τη διαδικτυακή εφαρμογή που σχεδιάστηκε να χρησιμοποιηθεί από τον κύριο διαχειριστή του συστήματος (EKAB), η εφαρμογή κινητών σχεδιάστηκε για να χρησιμοποιηθεί από τους εθελοντές. Για αυτό το σκοπό οι λειτουργίες σε σύγκριση με την διαδικτυακή εφαρμογή είναι λιγότερες διότι δόθηκε μεγαλύτερη σημασία στην επίτευξη του αρχικού σκοπού, ο οποίος είναι η γρήγορη ανταπόκριση σε ένα καρδιακό περιστατικό. Στις υποενότητες που ακολουθούν περιγράφονται αναλυτικά και με παραδείγματα οι οθόνες (screens) που σχεδιάστηκαν στην εφαρμογή καθώς και οι λειτουργίες τους. Το εικονίδιο της εφαρμογής δίνεται στην εικόνα 41.



Εικόνα 48: Εικονίδιο Εφαρμογής Κινητού

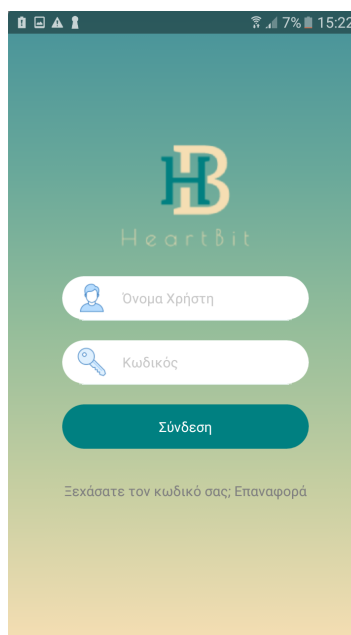
#### 4.2.1 Σελίδα Σύνδεσης Εθελοντή

Όπως στη διαδικτυακή εφαρμογή, έτσι και στην εφαρμογή κινητού, δόθηκε ιδιαίτερη σημασία στη διεπαφή χρήστη καθώς το κοινό που απευθύνεται πολύ πιθανόν να μην είναι εξοικειωμένο με τέτοιου είδους εφαρμογές. Ως κεντρική οθόνη της εφαρμογής ορίστηκε η οθόνη σύνδεσης. Στην οθόνη σύνδεσης, ο εθελοντής συμπληρώνει το email καθώς και τον κωδικό του. Σε περίπτωση επιτυχημένης εξακρίβωσης στοιχείων, οδηγείται σε κεντρικό περιβάλλον της εφαρμογής για εξουσιοδοτημένους χρήστες. Σε περίπτωση σφάλματος δέχεται κατάλληλο μήνυμα. Ο χρήστης παραμένει συνδεδεμένος έως ότου αποσυνδεθεί από την εφαρμογή πατώντας το κουμπί αποσύνδεσης στη πάνω δεξιά γωνία της οθόνης του, όπως φαίνεται στην εικόνα 42.

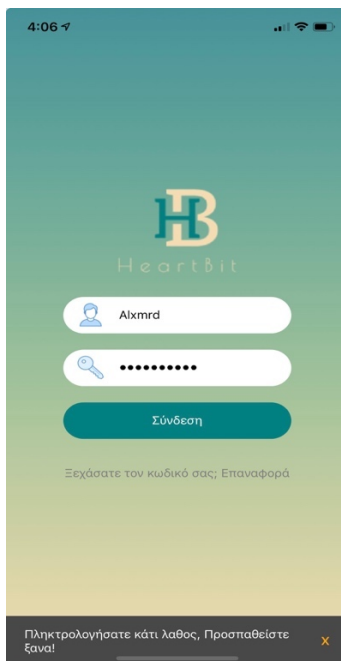


Εικόνα 49: Κουμπί Αποσύνδεσης Εφαρμογής

Ακολουθούν παραδείγματα σε εικόνες της οθόνης σύνδεσης καθώς και των μηνυμάτων εσφαλμένης απόπειρας εισόδου.

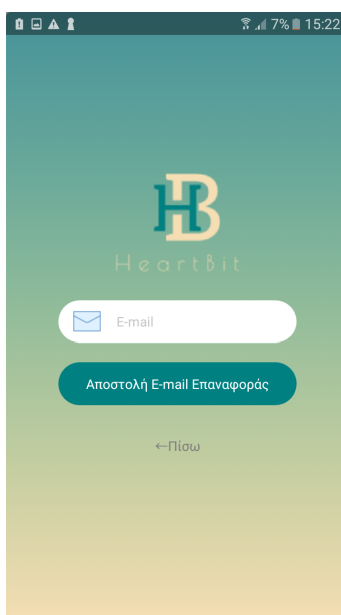


Εικόνα 50: Οθόνη σύνδεσης εφαρμογής



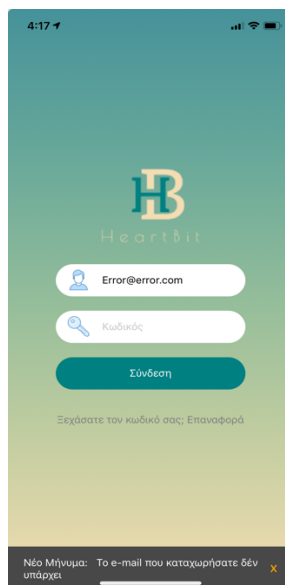
Εικόνα 51: Μήνυμα αποτυχημένης προσπάθειας εισόδου

Στην εφαρμογή του κινητού ο χρήστης έχει τη δυνατότητα ανάκτησης του κωδικού του πρόσβασης σε περίπτωση απώλειας του. Στην σελίδα σύνδεσης, υπάρχει σύνδεσμος, που οδηγεί στην οθόνη ανάκτησης κωδικού. Εκεί ο χρήστης συμπληρώνει σε φόρμα την διεύθυνση email, που έχει δηλώσει κατά την εγγραφή του και του αποστέλλεται email με σύνδεσμο επαναφοράς κωδικού.



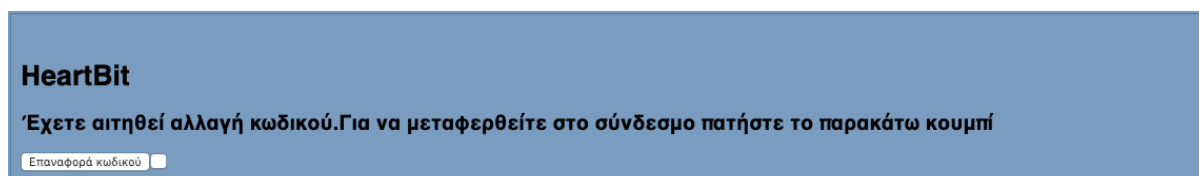
Εικόνα 52: Οθόνη Ανάκτησης κωδικού

Σε περίπτωση που ο χρήστης εισάγει E-mail το οποίο δεν υπάρχει στη βάση δεδομένων της εφαρμογής ο χρήστης ανακατευθύνεται στην οθόνη σύνδεσης και του εμφανίζεται κατάλληλο μήνυμα όπως φαίνεται στην εικόνα 46.



Εικόνα 53: Μήνυμα λάθος εισαγωγής E-mail για ανάκτηση κωδικού

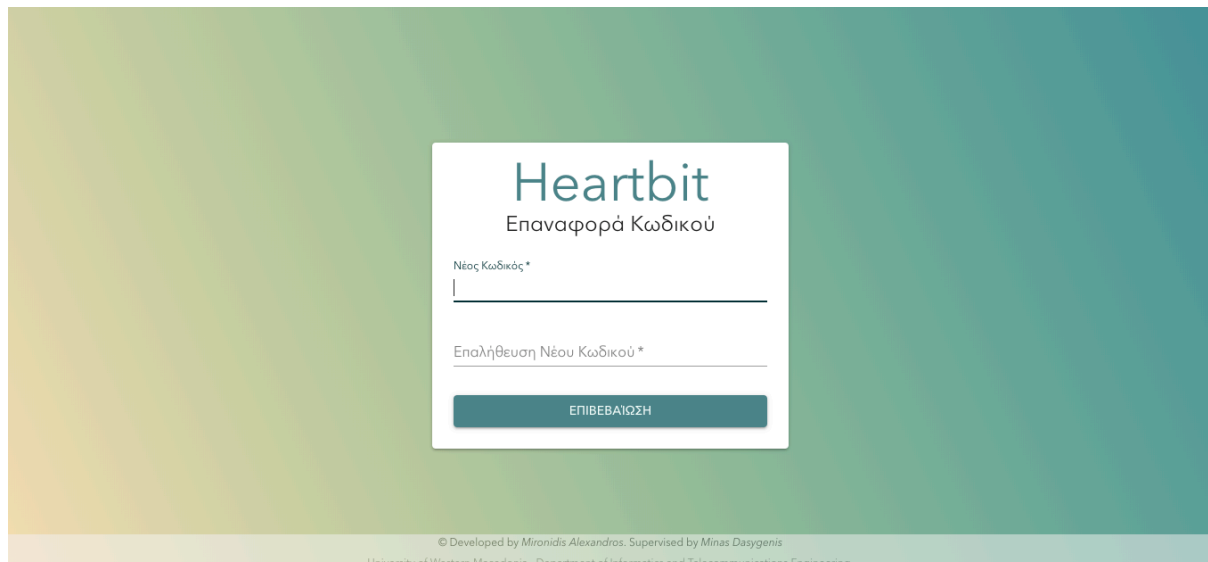
Ύστερα από επιτυχή συμπλήρωση του E-mail του ο χρήστης λαμβάνει σε αυτό έναν σύνδεσμο επαναφοράς κωδικού όπως φαίνεται στην εικόνα 47. Ο σύνδεσμος αυτός είναι κάθε φορά μοναδικός.



Εικόνα 54: E-mail με σύνδεσμο επαναφοράς κωδικού

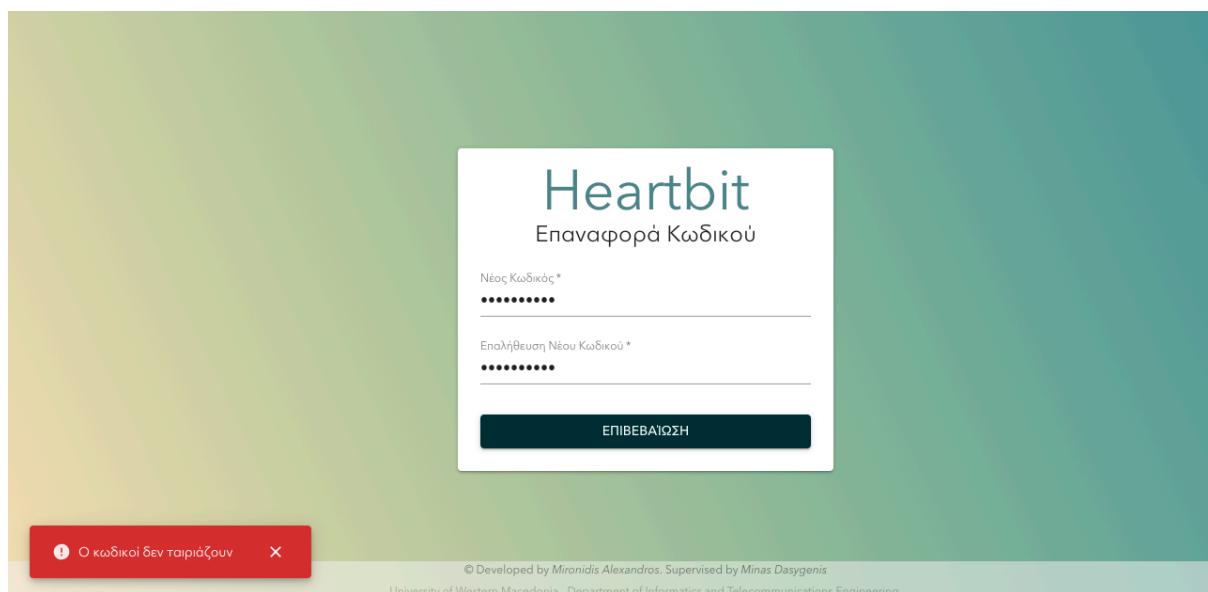
Με την επιλογή αυτού του συνδέσμου, ο χρήστης οδηγείται σε περιβάλλον της εφαρμογής, όπου του ζητείται να εισάγει τον καινούργιο κωδικό πρόσβασης καθώς και την

επιβεβαίωση αυτού. Με τον επιτυχή έλεγχο των πεδίων, ο κωδικός πρόσβασης ανανεώνεται με παράλληλη ενημέρωση του χρήστη, αλλιώς σε περίπτωση σφάλματος εμφανίζεται κατάλληλο μήνυμα.



The image shows a web form titled "Heartbit Επαναφορά Κωδικού". It contains two input fields: "Νέος Κωδικός \*" and "Επαλήθευση Νέου Κωδικού \*". Below the fields is a dark teal button labeled "ΕΠΙΒΕΒΑΙΩΣΗ". At the bottom of the page, there is a small copyright notice: "© Developed by Mironidis Alexandros. Supervised by Minas Dasygenis University of Western Macedonia, Department of Informatics and Telecommunications Engineering."

Εικόνα 55: Σελίδα επαναφοράς κωδικού εθελοντή

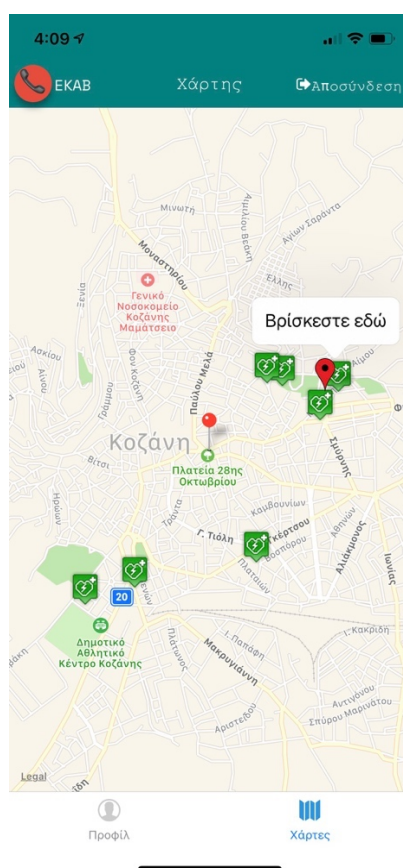


The image shows the same "Heartbit Επαναφορά Κωδικού" form as in Figure 55, but with the input fields filled with dots. A red error message box is visible at the bottom left, containing the text "Ο κωδικός δεν ταιριάζουν" and a close button (X). The same copyright notice is present at the bottom of the page.

Εικόνα 56: Μήνυμα λάθος συμπλήρωσης νέου κωδικού

## 4.2.2 Κεντρικό περιβάλλον εξουσιοδοτημένου χρήστη – Οθόνη Χάρτη

Ύστερα από επιτυχή συμπλήρωση των στοιχείων του, ο χρήστης μεταφέρεται στη κεντρική οθόνη της εφαρμογής, από την οποία μπορεί να περιηγηθεί στην εφαρμογή. Αυτή η σελίδα έχει οριστεί ως η οθόνη του χάρτη. Στη συγκεκριμένη οθόνη, ο εθελοντής μπορεί να δει όλους τους απινιδωτές που έχουν εγκατασταθεί στα πλαίσια της εφαρμογής, καθώς και που βρίσκεται ο ίδιος σε σχέση με αυτούς.

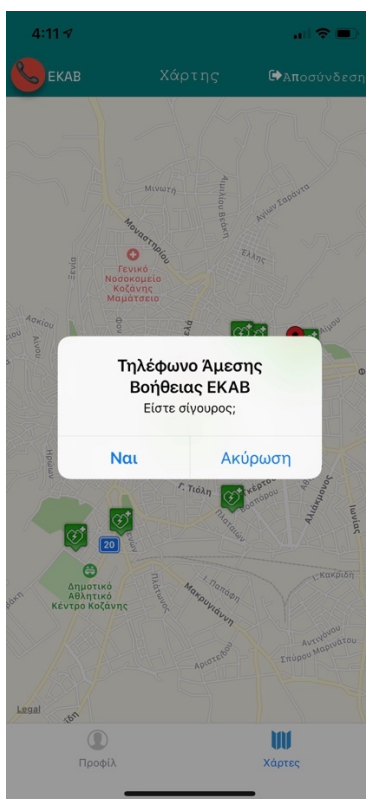


Εικόνα 57: Οθόνη χάρτη εφαρμογής κινητού



Όπως θα παρατηρήσει κανείς στην εικόνα που δόθηκε ως παράδειγμα, ότι η εφαρμογή κινητού απαρτίζεται από τρία ξεχωριστά πλαίσια. Το πρώτο είναι η μπάρα του τίτλου ή κεφαλίδα (HeaderBar), στην οποία υπάρχει στο κέντρο ο τίτλος της εκάστοτε οθόνης που βρίσκεται ο χρήστης. Στα δεξιά της μπάρας τίτλου έχει τοποθετηθεί το κουμπί αποσύνδεσης



όπως αναφέρθηκε προηγουμένως. Ενώ στα αριστερά έχει τοποθετηθεί ένα κουμπί άμεσης ανάγκης κλήσης στο ΕΚΑΒ όπως φαίνεται στην παρακάτω εικόνα.

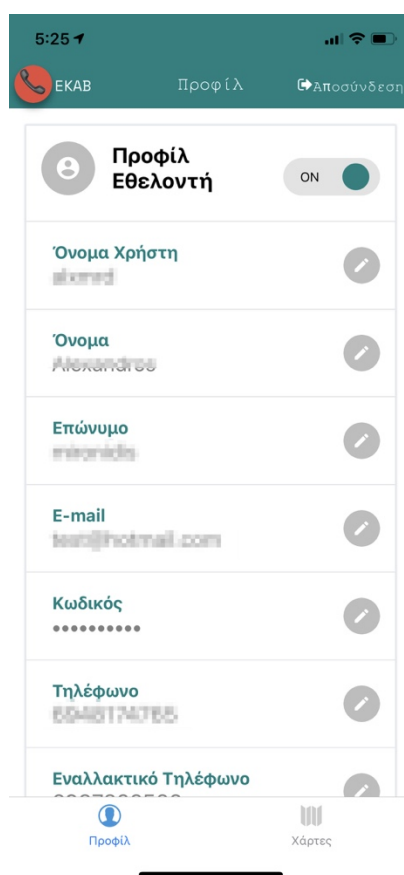


Εικόνα 58: Ειδοποίηση κλήσης άμεσης βοήθειας ΕΚΑΒ

Στο δεύτερο υπάρχει η οθόνη της εφαρμογής. Στη προκειμένη περίπτωση αυτή η οθόνη είναι η οθόνη του χάρτη. Τέλος, το τρίτο πλαίσιο που βρίσκεται στο κάτω μέρος της οθόνης είναι η μπάρα πλοήγησης (NavBar). Στη μπάρα πλοήγησης έχουν τοποθετηθεί κουμπιά με σκοπό τη πλοήγηση του χρήστη στην εφαρμογή. Η μπάρα πλοήγησης της εφαρμογής που αναπτύχθηκε σε αυτή τη διπλωματική έχει το κουμπί “Χάρτες” στα δεξιά με το εξής εικονίδιο  , ενώ στα αριστερά το κουμπί “Προφίλ” με το εξής εικονίδιο  . Ανάλογα την οθόνη που βρίσκεται ο χρήστης το εικονίδιο μεταβάλλει το χρώμα του όπως φαίνεται στο παράδειγμα της εικόνας 50. Η μπάρα κεφαλίδας, καθώς και η μπάρα πλοήγησης είναι στατικές κατά την πλοήγηση του χρήστη στην εφαρμογή. Το μόνο που αλλάζει είναι η κεντρική οθόνη.

### 4.2.3 Οθόνη Προφίλ Εθελοντή

Η δεύτερη οθόνη του εθελοντή, είναι η οθόνη του προφίλ του. Σε αυτή την οθόνη, ο εθελοντής έχει τη δυνατότητα να δει τα στοιχεία που έχει καταχωρήσει στην εφαρμογή καθώς και να τα επεξεργαστεί όλα εκτός τις ημερομηνία γέννησης και ημερομηνία τελευταίας εκπαίδευσης, η οποία επεξεργάζεται μόνο από τον χρήστη.




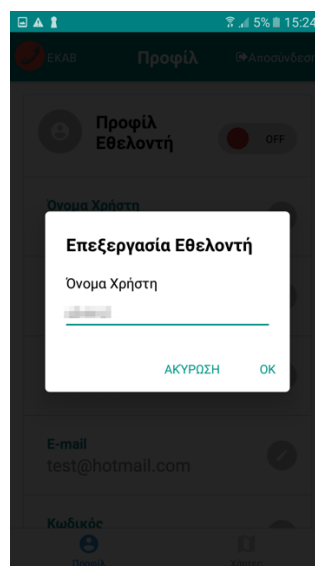
Εικόνα 59: Οθόνη Προφίλ Εθελοντή

Επίσης ο εθελοντής στην συγκεκριμένη οθόνη έχει την δυνατότητα να θέσει τον λογαριασμό του ως ανενεργό έτσι ώστε να μη του έρχονται ειδοποιήσεις μετακινώντας την μπάρα που βρίσκεται στα δεξιά του Avatar από ON σε OFF όπως φαίνεται στο παρακάτω παράδειγμα.



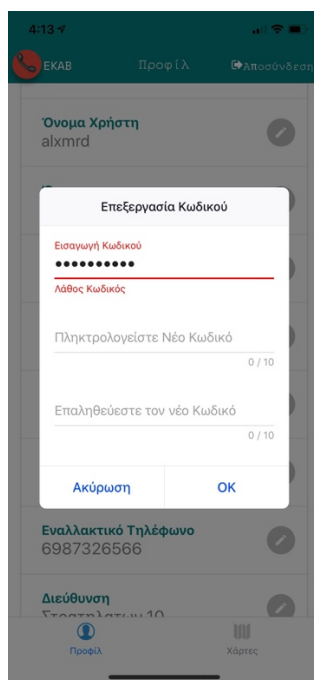
Εικόνα 60: Μπάρα απενεργοποίησης χρήστη

Στην οθόνη του προφίλ, ο εθελοντής έχει τη δυνατότητα επεξεργασία των στοιχείων του. Για να το πετύχει αυτό, αρκεί να πατήσει το κουμπί επεξεργασίας που βρίσκεται στα δεξιά από κάθε πεδίο. Το κουμπί επεξεργασίας έχει το εξής εικονίδιο . Ύστερα από κλικ στο εικονίδιο, εμφανίζεται στο χρήστη μια καρτέλα επεξεργασίας του αντίστοιχου πεδίου όπως φαίνεται στην εικόνα που ακολουθεί.



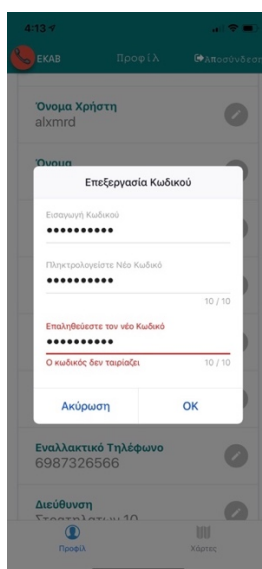
Εικόνα 61: Καρτέλα επεξεργασίας πεδίου

Τα στοιχεία που υποβάλλονται στη φόρμα επεξεργασίας, περνούν από το αντίστοιχο validation όπως στη διαδικτυακή εφαρμογή. Αν τα στοιχεία δεν είναι αποδεκτά, εμφανίζεται στο χρήστη αντίστοιχο μήνυμα. Σε αντίθετη περίπτωση τα στοιχεία του πεδίου αλλάζουν επιτυχώς και ενημερώνονται αυτόματα. Στο πεδίο του κωδικού υπάρχει και server-side validation και client-side validation, με προτροπή του χρήστη να υποβάλλει τον παλιό του κωδικό πριν τον αλλάξει. Σε περίπτωση που εισάγει λάθος κωδικό, εμφανίζεται αντίστοιχο μήνυμα όπως φαίνεται στην εικόνα 54.



Εικόνα 62: Μήνυμα λάθος εισαγωγής κωδικού στην καρτέλα επεξεργασίας

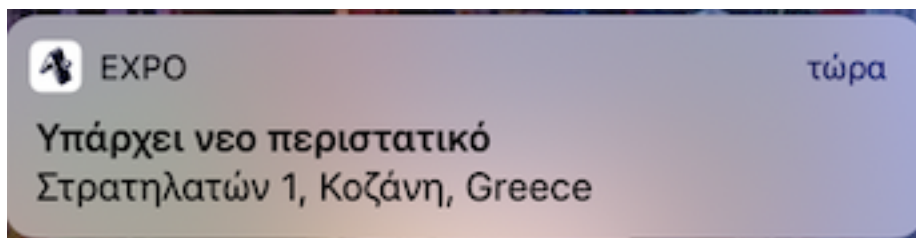
Επίσης, για την αλλαγή του κωδικού του, πρέπει τον νέο κωδικό να τον υποβάλλει δύο φορές για την αποφυγή ανεπιθύμητων λαθών. Σε περίπτωση που τα πεδία δεν ταιριάζουν το σύστημα ενημερώνει τον χρήστη.



Εικόνα 63: Μήνυμα λάθος εισαγωγής νέου κωδικού στην καρτέλα επεξεργασίας

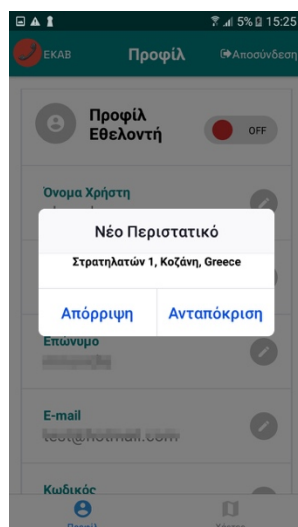
#### 4.2.4 Κεντρική λειτουργία Εφαρμογής

Η εφαρμογή για κινητό τηλέφωνο δημιουργήθηκε με πρώτο και κύριο σκοπό, την άμεση βοήθεια σε περίπτωση περιστατικού. Η διαδικασία αυτή έχει αναλυθεί με λεπτομέρεια σε προηγούμενο κεφάλαιο της διπλωματικής εργασίας. Στη παρούσα ενότητα θα περιγραφεί η συμμετοχή του εθελοντή στην διαδικασία αυτή. Σε μια τέτοια δύσκολη κατάσταση, αντίπαλος είναι χρόνος. Για αυτό τον λόγο δόθηκε ιδιαίτερη έμφαση στις γρήγορες ενέργειες από πλευράς εθελοντή. Όσο ο εθελοντής είναι συνδεδεμένος στην εφαρμογή, ακόμα και αν η εφαρμογή βρίσκεται στο παρασκήνιο, του έρχεται ειδοποίηση στο κινητό με τη μορφή push notification για να τον ενημερώσει για την ύπαρξη νέου περιστατικού.



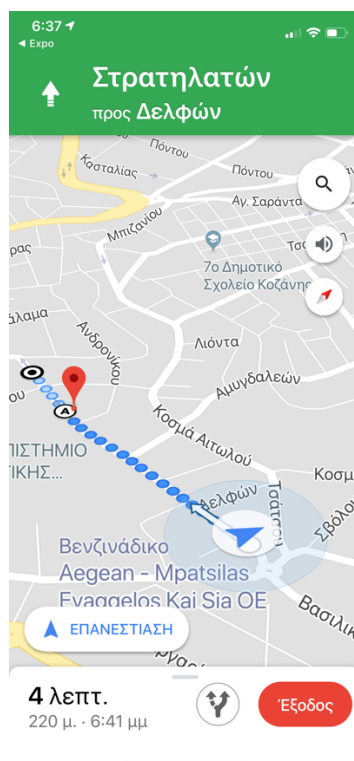
Εικόνα 64: Ειδοποίηση νέου περιστατικού σε κατάσταση εφαρμογής στο παρασκήνιο

Ύστερα από κλικ στην ειδοποίηση, ο εθελοντής ανακατευθύνεται στο περιβάλλον της εφαρμογής και του εμφανίζεται μια καρτέλα. Η καρτέλα έχει την δυνατότητα απόρριψης του περιστατικού ή ανταπόκρισης. Σε περίπτωση ανταπόκρισης ενημερώνεται ο διαχειριστής του συστήματος δηλαδή το ΕΚΑΒ.

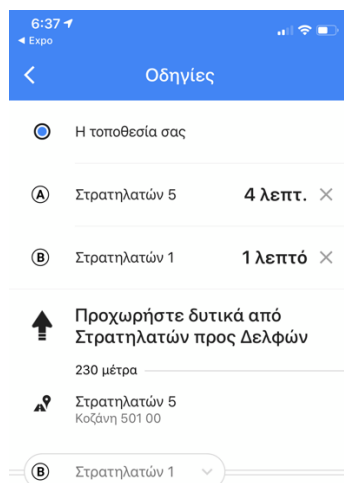


Εικόνα 65: Καρτέλα νέου περιστατικού με μορφή ειδοποίησης

Όπως παρατηρείται στην εικόνα 58, ο τίτλος του κουμπιού “Ανταπόκριση”, είναι ελαφρώς πιο Bold, προτρέποντας έτσι τον εθελοντή να ανταποκριθεί στο περιστατικό. Αν ο εθελοντής αποφασίσει να ανταποκριθεί στο περιστατικό και πατήσει το κουμπί, τότε αυτόματα ανακατευθύνεται στους χάρτες του κινητού του. Οι χάρτες μπορεί να είναι Google Maps, Apple Maps, ή αν δεν υπάρχει εφαρμογή τότε ανοίγουν οι χάρτες από τον προεπιλεγμένο φυλλομετρητή. Στους χάρτες θα υπάρχει διαδρομή από την τωρινή τοποθεσία του εθελοντή με πρώτο προορισμό τον πλησιέστερο απινιδωτή σε σχέση με τη τοποθεσία του και το περιστατικό. Ως δεύτερος προορισμός έχει οριστεί προφανώς από τον απινιδωτή στο περιστατικό. Οι οδηγίες έχουν οριστεί ως πλοήγηση με τα πόδια. Ακολουθεί παράδειγμα στην εικόνα 59 και 60.



Εικόνα 66: Πλοήγηση προς τον απινιδωτή και το περιστατικό



Εικόνα 67: Οδηγίες προς το πρώτο σημείο που βρίσκεται ο απινιδωτής και έπειτα στο περιστατικό καθώς και η απόσταση μετρημένη στην κλίμακα του χρόνου.

## 4.3 Ανάλυση σημαντικών σημείων κώδικα

Στο παρόν κεφάλαιο θα αναλυθούν τα σημαντικότερα κομμάτια κώδικα και βιβλιοθήκες που αναπτύχθηκαν για τη διπλωματική εργασία. Ο κώδικας αυτός αποτελείται από τρία μέρη τα οποία συνδέονται μεταξύ τους. Αυτά είναι η διαδικτυακή εφαρμογή, η εφαρμογή κινητών και τέλος το api. Θα ξεκινήσουμε αναλύοντας τον κώδικα της διαδικτυακής εφαρμογής στην υποενότητα που ακολουθεί.

### 4.3.1 Ανάλυση σημαντικών σημείων κώδικα διαδικτυακής εφαρμογής

#### react-router-dom

Στο πλαίσιο της διαδικτυακής εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη “react-router-dom”. Η βιβλιοθήκη αυτή είναι υπεύθυνη για τη πλοήγηση του χρήστη μεταξύ των σελίδων της εφαρμογής με ασύγχρονο τρόπο. Η κύρια λειτουργία της αποδίδεται στο παράδειγμα της εικόνας 61.

```
export default function route() {  
  return (  
    <Switch>  
      <Route path="/home" exact component={Home} />  
      <Route path="/Volunteers" exact component={Volunteers} />  
      <Route path="/defibrillators" exact component={defibrillators} />  
      <Route path="/patients" exact component={patients} />  
      <Route path="/admin" exact component={admin} />  
      <Route path="/event" exact component={event} />  
      <Route path="/" exact component={Login} />  
      <Route path="/map" exact component={map} />  
      <Route  
        path="/restorepassword/:email/:restoretoken"  
        exact  
        component={RestorePassword}  
      />  
      <Route component={NotFound} />  
    </Switch>  
  );  
}
```

Εικόνα 68: Κώδικας με τη χρήση της βιβλιοθήκης “react-router-dom”



## Redux

Στην υλοποίηση της διαδικτυακής εφαρμογής σημαντικό ρόλο έπαιξε η “Redux” η οποία είναι ένας state manager για εφαρμογές με κώδικα Javascript. Στη παρακάτω εικόνα αναγράφεται το store της Redux.

```
const initialState = {};  
const middleware = [thunk];  
  
const store = createStore(  
  rootReducer,  
  initialState,  
  composeWithDevTools(  
    applyMiddleware(...middleware)  
  )  
);  
  
export default store;
```

Εικόνα 69: Κώδικας αρχείου store της Redux

## geocodeByAddress

Κατά την ανάπτυξη της διαδικτυακής εφαρμογής δημιουργήθηκε η ανάγκη να βρεθεί το γεωγραφικό μήκος και πλάτος μιας συγκεκριμένης διεύθυνσης. Ο τρόπος εκπλήρωσης αυτής της ανάγκης ήρθε μέσω της βιβλιοθήκης “react-places-autocomplete” και της συνάρτησης “geocodeByAddress”. Ο τρόπος σύνταξης της φαίνεται στην εικόνα 63.

```

geocodeByAddress(suggestion)
  .then(results => getLatLng(results[0]))
  .then(latLng =>
    this.setState({
      latitude: latLng.lat,
      longitude: latLng.lng,
      location: suggestion
    })
  );

```

Εικόνα 70: Κώδικας συνάρτησης geocodeByAddress

## Pusher

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, η διαδικτυακή εφαρμογή δέχεται ή από Arduino ή από την εφαρμογή κινητού κάποια response σε πραγματικό χρόνο και αυτό επιτεύχθηκε με τη βιβλιοθήκη pusher. Για να είναι ικανή η διαδικτυακή εφαρμογή να “ακούει” αυτά τα response χρειάστηκε να αναπτυχθεί ο ακόλουθος κώδικας.

```

var pusher = new Pusher('...') {
  cluster: "eu",
  forceTLS: true
});

var channel = pusher.subscribe("channel");
channel.bind("arduino", data => {
  this.props.onStoreArduinoData(data);
});

```

Εικόνα 71: Κώδικας βιβλιοθήκης pusher

## generate-password

Τέλος, αξίζει να αναφερθεί και η συνάρτηση που χρησιμοποιήθηκε για την παραγωγή τυχαίων κωδικών και RFID. Η συνάρτηση αυτή ονομάζεται “generate-password” και δίνει τη δυνατότητα για παραγωγή τυχαίων χαρακτήρων, αριθμών, γραμμάτων και συμβόλων, των οποίων το μήκος το καθορίζει η συνάρτηση. Στο παράδειγμα που ακολουθεί στην εικόνα 66 το μήκος που έχει επιλεγεί είναι δέκα χαρακτήρες και οι αριθμοί είναι αποδεκτοί.

```

onGenerate = event => {
  event.preventDefault();

  var generator = require("generate-password");

  var password = generator.generate({
    length: 10,
    numbers: true
  });

  // 'uEyMTw32v9'
  this.setState({
    password: password,
    hasChanged: true
  });
};

```

Εικόνα 72: Κώδικας συνάρτησης generate-password

### 4.3.2 Ανάλυση σημαντικών σημείων κώδικα εφαρμογής κινητού

Στο σημείο αυτό, αναλύονται τα σημαντικότερα κομμάτια κώδικα που έπαιξαν ρόλο στην ανάπτυξη της κινητής Εφαρμογής. Σαφώς το pusher και η redux ήταν εξίσου σημαντικά και στην υλοποίηση της εφαρμογής κινητού όπως αναφέρθηκαν στο προηγούμενο κεφάλαιο.

#### react-native-awesome-alerts

Ιδιαίτερη έμφαση στην εφαρμογή δόθηκε στη δημιουργία ειδοποιήσεων που είτε θα προτρέπουν τον χρήστη να ανταποκριθεί στο περιστατικό. Για την υλοποίηση αυτή χρειάστηκε η βιβλιοθήκη με όνομα “react-native-awesome-alerts”.

```

_showAlert = () => {
  Alert.alert(
    "Τηλέφωνο Άμεσης Βοήθειας EKAB",
    "Είστε σίγουρος;",
    [
      {
        text: "Ναι",
        onPress: () => Linking.openURL("tel:+302461029166"),
        style: "cancel"
      },
      {
        text: "Ακύρωση",
        onPress: () => console.log("Cancel Pressed")
      }
    ],
    { cancelable: false }
  );
};

```

Εικόνα 73: Κώδικας βιβλιοθήκης generate-password

## handleGetDirections

Η κύρια λειτουργία της κινητής εφαρμογής είναι οι οδηγίες στον χάρτη προς τον απινιδωτή και το περιστατικό. Για την υλοποίηση αυτής της λειτουργίας χρειάστηκε μια συνάρτηση που θα μεταφέρει τον χρήστη σε περιβάλλον του χάρτη του κινητού, με οδηγίες για περπάτημα προς τους απαιτούμενους προορισμούς. Αυτή η συνάρτηση είναι η “handleGetDirections”

```
_showAlert = () => {
  Alert.alert(
    "Τηλέφωνο Άμεσης Βοήθειας ΕΚΑΒ",
    "Είστε σίγουρος;",
    [
      {
        text: "Ναι",
        onPress: () => Linking.openURL("tel:+302461029166"),
        style: "cancel"
      },
      {
        text: "Ακύρωση",
        onPress: () => console.log("Cancel Pressed")
      }
    ],
    { cancelable: false }
  );
};
```

Εικόνα 74: Κώδικας συνάρτησης handleGetDirections

## getCurrentPositionAsync

Η εφαρμογή κινητού διαθέτει pin που εμφανίζει τη τωρινή τοποθεσία του χρήστη. Τόσο για λόγους διεπαφής χρήστη, όσο και γιατί απαιτείται για τις οδηγίες προς το

περιστατικό. Η react-native διαθέτει έτοιμη συνάρτηση που κάνει τη δουλειά του προγραμματιστή πιο εύκολη. Αυτή η συνάρτηση είναι η `getCurrentPositionAsync` και η λειτουργία της φαίνεται παρακάτω στην εικόνα 68.

```
let { status } = await Permissions.askAsync(Permissions.LOCATION);
if (status !== "granted") {
  alert = "Η Τοποθεσία είναι απενεργοποιημένη, μεταβείτε στις ρυθμίσεις!"
}

let location = await Location.getCurrentPositionAsync({});

let currentLocation = location.coords;
```

Εικόνα 75: Κώδικας συνάρτησης `getCurrentPositionAsync`

## GetPathLength

Τέλος, στη κινητή εφαρμογή αναπτύχθηκε αλγόριθμος που υπολογίζει τον πλησιέστερο απινιδωτή μεταξύ της τοποθεσίας του χρήστη και του περιστατικού. Για την υλοποίηση αυτού του αλγορίθμου χρησιμοποιήθηκαν συνάρτηση `getPathLength` της βιβλιοθήκης `geolib`. Επίσης χρειάστηκαν και οι συναρτήσεις `indexOf` και `Math.min` της Javascript. Ο κώδικας βρίσκεται στην εικόνα 69.

```
var distances = state.defibrillators.map(item =>
  geolib.getPathLength([
    {
      latitude: state.currentLocation.latitude,
      longitude: state.currentLocation.longitude
    }, // που είσαι
    { latitude: item.latitude, longitude: item.longitude }, // 8 απινιδω
    { latitude: state.event.latitude, longitude: state.event.longitude }
  ])
);
let nearestDefIndex = distances.indexOf(Math.min(...distances));

let nearestDefibrillator = state.defibrillators[nearestDefIndex];
```

Εικόνα 76: Κώδικας συνάρτησης `getPathLength`

### 4.3.3 Ανάλυση σημαντικών σημείων RESTful API

Στο σημείο αυτό, αναλύονται τα σημαντικότερα κομμάτια κώδικα που έπαιξαν ρόλο στην ανάπτυξη της του backend. Το backend είναι υπεύθυνο για τις αιτήσεις που γίνονται τόσο από τη διαδικτυακή εφαρμογή όσο και από την εφαρμογή κινητού. Επίσης η ασφάλεια του συστήματος υπάρχει στο κομμάτι του backend.

#### JWTauthentication

Το “JWTauthentication” (Jason Web Token) είναι ένας από τους πιο γνωστούς και ασφαλείς τρόπους για τη σύνδεση εξουσιοδοτημένων χρηστών. Με το JWT ο χρήστης ύστερα από επιτυχή είσοδο δέχεται ένα token σε μορφή Jason, το οποίο είναι απαραίτητο για την είσοδο σε σελίδες εξουσιοδοτημένων χρηστών. Το token είναι κάθε φορά μοναδικό. Αν ο χρήστης δεν έχει το συγκεκριμένο token τότε η εφαρμογή τον ανακατευθύνει στην αρχική σελίδα σύνδεσης. Παράδειγμα ανάπτυξης κώδικα που υλοποιεί το JWTauthentication ακολουθεί στην εικόνα 69.

```

$JwtAuthentication=new Tuupola\Middleware\JwtAuthentication([
    "attribute" => "decoded_token_data",

    "path" => ["/api"],
    "ignore" => ["/api/login","/api/mobilelogin","/volunteers/push-token","/volunteer/restorepassword"],
    "secure" => true,
    "relaxed" => ["localhost", "dev.example.com"], // when in deployment must set from localhost to zafora
    "secret" => getenv("SECRET_KEY"),
    "algorithm" => ["HS256"],
    "before" => function ($request, $arguments) use ($container) {

        $container["jwt"] = $arguments["decoded"];
    },
    "error" => function ($response, $arguments) {
        $data["status"] = "error";
        $data["message"] = $arguments["message"];

        return $response
            ->withHeader("Content-Type", "application/json")
            ->write(json_encode($data, JSON_UNESCAPED_SLASHES | JSON_PRETTY_PRINT));
    }
]);

```

Εικόνα 77: Κώδικας JWTauthentication

## RespectValidator

Άλλο ένα κομμάτι που εξυπηρετεί την ασφάλεια, είναι ο έλεγχος των δεδομένων που λαμβάνονται από τον client (validation). Σε αυτό το κομμάτι, σημαντικό ρόλο διατέλεσε η χρήση της πιο γνωστής βιβλιοθήκης της PHP για validation δεδομένων. Το όνομα της βιβλιοθήκης είναι “RespectValidator” και οι κανόνες που ορίστηκαν είναι οι εξείς.

```
use Respect\Validation\Validator as v;
$greek = 'ς ε ρ τ υ θ ι ο π α σ δ φ γ η ξ κ λ ζ χ ψ ω β ν μ Ε Ρ Τ Υ Θ Ι Ο Π Α Σ Δ Φ Γ Η Ξ Κ Λ Ζ Χ Ψ Ω Β Ν Μ έ ύ ί ό ά ή ώ ';
$usernameValidator = v::alnum($greek)->noWhitespace()->length(1, 32)->notBlank();//Validates alphanumeric characters from a-
$nameValidator = v::stringType()->alnum($greek)->noWhitespace()->length(1, 32)->notBlank();//Validates string characters fro
$surnameValidator = v::stringType()->alnum($greek)->noWhitespace()->length(1, 32)->notBlank();//Validates string characters
$dateValidator = v::date('Y-m-d')->between('1950-01-01', '2000-01-01')->notBlank();//Date must be between 1/1/1950 to 1/1/2
$trainingDateValidator = v::optional(v::date('Y-m-d')->between('2010-01-01', 'now')); //Training Date must be between 1/1/20
$emailValidator = v::email()->notBlank()->alnum('@.');//Validates email format,required
$mobileNumberValidator = v::positive()->notBlank()->length(10,10);//Validates mobile number format ,required
$mobile2NumberValidator = v::optional(v::positive()->length(10,10));//Validates mobile number format
$addressValidator=v::optional(v::alnum($greek));//Validates alphanumeric characters from a-Z and 0-9
$locationValidator = v::stringType()->alnum($greek)->notBlank();// validates location to be string and not blank,required
$passwordValidator = v::stringType()->alnum($greek)->notBlank()->length(10,10);
$defibrillatordateValidator=v::optional(v::date('Y-m-d')->between('2019-01-01', 'now'))->notBlank();
$patientDateOfBirth=v::intType()->min(1950)->max(2010);
$adminTypeValidator=v::alnum($greek)->stringType()->noWhitespace();
$RFIDValidator = v::intVal()->notBlank();
```

Εικόνα 78: Κώδικας με το σύνολο validation κανόνων

## ExpoNotify

Για την αποστολή ειδοποιήσεων χρησιμοποιήθηκε η συνάρτηση της notify της PHP που δίνεται από το documentation του Expo. Παράδειγμα υλοποίησης της ακολουθεί στην εικόνα 71.

```
$interestDetails = ['heartbit-e68af', 'ExponentPushToken
// You can quickly bootup an expo instance
$expo = \ExponentPhpSDK\Expo::normalSetup();
// Subscribe the recipient to the server
$expo->subscribe($interestDetails[0], $interestDetails[1]);
// Build the notification data
$notification = ['title' => 'Έχετε νέο μήνυμα', 'body'=>$message];
// Notify an interest with a notification
$expo->notify($interestDetails[0], $notification);
return json_encode($myObj);
```

Εικόνα 79: Κώδικας με χρήση της συνάρτησης Expo->notify

## PusherTrigger

Όσο σημαντικό ρόλο έπαιξε το pusher στις client-side εφαρμογές, δηλαδή την διαδικτυακή και την εφαρμογή του κινητού, άλλο τόσο ρόλο έπαιξε και στο backend. Ο λόγος είναι ότι για να “ακούσει” η εφαρμογή ένα event πρέπει κάποιος να το αποστείλει. Εκεί αναλαμβάνει το backend με τη συνάρτηση trigger του pusher.

```
$options = array(
    'cluster' => 'eu',
    'useTLS' => true
);
$pusher = new Pusher\Pusher(
    '700333',
    $options
);
$myObj = new stdClass();
$myObj->longitude = $longitude;
$myObj->latitude = $latitude;
$myObj->address = $address;

$data = $myObj;
//$data['address'] = $myObj;
$pusher->trigger('channel', 'peristatiko', $data);
```

Εικόνα 80: Κώδικας με χρήση της συνάρτησης pusher->trigger

## 4.4 Σύνοψη κεφαλαίου

Στο παρόν κεφάλαιο έγινε μία ανάλυση του τρόπου λειτουργίας των Εφαρμογών. Παρουσιάστηκε το γραφικό τους περιβάλλον και όλες οι ενέργειες που μπορούν να εκτελέσουν οι χρήστες. Τέλος περιγράφηκαν μερικά σημαντικά σημεία του κώδικα .



## Κεφάλαιο 5ο – Επίλογος

Στο κεφάλαιο αυτό γίνεται μια σύνοψη της Ιστοσελίδας και της Εφαρμογής που σχεδιάστηκε και υλοποιήθηκε σε αυτή τη διπλωματική εργασία. Παρουσιάζεται μία ανακεφαλαίωση των όσων παρουσιάστηκαν και αναλύθηκαν, καταγράφονται τα συμπεράσματα που εξάχθηκαν κατά την υλοποίηση και περιγράφονται οι μελλοντικές επεκτάσεις που μπορούν να υπάρξουν για τη βελτιστοποίηση του συστήματος. Όπως ήταν λογικό ωστόσο, προέκυψαν ορισμένα προβλήματα κατά τη φάση του σχεδιασμού και της υλοποίησης τα οποία όμως ξεπεράστηκαν.

### 5.1 Σύνοψη και συμπεράσματα

Συνοψίζοντας, καταλαβαίνει κανείς ότι το έργο αυτό αποτελεί μια καινοτόμα ιδέα, η οποία συνδυάζει διαφορετικές τεχνολογίες για την άμεση επέμβαση σε ένα καρδιακό επεισόδιο. Επιπροσθέτως, συνδυάζει την αμφίδρομη επικοινωνία μεταξύ των κινητών συσκευών με την εφαρμογή του συστήματος. Όσον αφορά την κινητή εφαρμογή, έχει δοθεί κυρίως έμφαση στην επιτέλεση του αρχικού στόχου που τέθηκε στη διπλωματική εργασία, αποφεύγοντας έτσι, ανούσιες λειτουργικότητες που θα αποτρέψουν τον εθελοντή στην χρήση αυτής. Από την άλλη, στη διαδικτυακή εφαρμογή δόθηκε ιδιαίτερη σημασία στην διεπαφή του χρήστη, με σκοπό τέτοιο ώστε να είναι προφανής και εύκολη η κάθε λειτουργία της.

Οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για τον ιστοχώρο είναι οι: HTML5, CSS, PHP, JavaScript (React Framework), AJAX και το framework του Material-ui. Η MySQL βάση δεδομένων περιλαμβάνει πέντε πίνακες, των διαχειριστών, των εθελοντών, των απινιδωτών, των περιστατικών, και των ασθενών. Για μεγαλύτερη ασφάλεια, η σύνδεση με το σύστημα και η διαχείριση των δεδομένων της βάσης γίνεται με τη χρήση προκατασκευασμένων εντολών PDO, καθώς και με μοναδικά tokens σύνδεσης. Για επιπλέον ασφάλεια, χρησιμοποιούνται κρυπτογραφημένοι κωδικοί πρόσβασης, γίνεται φιλτράρισμα και επικύρωση των τύπων των δεδομένων που εισάγονται από το χρήστη από τις διάφορες φόρμες επικοινωνίας, και ασφαλή σύνδεση HTTPS. Για την κινητή εφαρμογή χρησιμοποιήθηκε η γλώσσα Javascript και ποιο συγκεκριμένα το framework της React Native. Η αποστολή αρχείων μεταξύ εφαρμογής και ιστοσελίδας γίνεται μέσω JSON.

Η δημιουργία τέτοιων εφαρμογών αποτελούν μονόδρομο για το μέλλον, καθώς χρησιμοποιείται η τεχνολογία προς όφελος των ανθρώπων και ιδιαίτερα στον τομέα της υγείας. Επιπλέον ενθαρρύνει τον εθελοντισμό και εμπνέει το συλλογικό ενδιαφέρον για τον συνάνθρωπο.

## 5.2 Προβλήματα που προέκυψαν και η αντιμετώπιση τους

Το κύριο πρόβλημα υπήρξε η αρχική απαίτηση της παρούσας διπλωματικής, που ήταν η δημιουργία μίας διαδικτυακής εφαρμογής που αποτελείται από τεχνολογίες του σήμερα, οι οποίες χρησιμοποιούνται από διαδικτυακούς κολοσσούς. Ένα ακόμη πρόβλημα που προέκυψε κατά την υλοποίηση της κινητής εφαρμογής, ήταν η εμφάνιση ενός κοινού αποτελέσματος σε διαφορετικά μεγέθη οθονών. Η αντιμετώπιση και των δύο προβλημάτων ήρθε με την μελέτη και την αναζήτηση λύσεων στο διαδίκτυο, ωστόσο χρειάστηκαν αρκετές δοκιμές και τεράστια υπομονή ώστε να έρθει το επιθυμητό αποτέλεσμα. Με την χρησιμοποίηση τέτοιων τεχνολογιών στην παρούσα εργασία γίνεται σαφές ότι, έγινε μεγάλη προσπάθεια εναρμόνισης και συγχρονισμού της εφαρμογής με τα τεχνολογικά μέσα της σύγχρονης εποχής.

## 5.3 Μετρικά κώδικα συστήματος

Σε αυτή την ενότητα παραθέτονται τα στοιχεία σχετικά με τους κώδικες που αναπτύχθηκαν για τον προγραμματισμό της Ιστοσελίδας και της Εφαρμογής καθώς και του backend.

### 5.3.1 Μετρικά κώδικα διαδικτυακής εφαρμογής

Για την υλοποίηση της διαδικτυακής εφαρμογής δημιουργήθηκαν 58 αρχεία πηγαίου κώδικα Javascript, 2 αρχεία CSS και 8 εικονίδια. Πιο αναλυτικά φαίνονται στον πίνακα 7.

<b>Ομάδα αρχείων</b>	<b>Αριθμός γραμμών</b>
Controllers	3400
JS - React	2100
Other	400

Πίνακας 7: Μετρικές διαδικτυακής εφαρμογής

### 5.3.2 Μετρικά κώδικα κινητής εφαρμογής εφαρμογής

Για την υλοποίηση της κινητής εφαρμογής δημιουργήθηκαν 21 αρχεία πηγαίου κώδικα Javascript, και 7 εικονίδια. Πιο αναλυτικά φαίνονται στον πίνακα 8.

<b>Ομάδα αρχείων</b>	<b>Αριθμός γραμμών</b>
Controllers	800
JS - React	1600
Other	200

Πίνακας 8: Μετρικές κινητής εφαρμογής

### 5.3.3 Μετρικά κώδικα backend

Για την υλοποίηση του Backend δημιουργήθηκαν 5 αρχεία πηγαίου κώδικα PHP. Πιο αναλυτικά φαίνονται στον πίνακα 9.

<b>Ομάδα αρχείων</b>	<b>Αριθμός γραμμών</b>
Configuration	100
PHP (routes)	2300

Πίνακας 9: Μετρικές Backend

## 5.4 Ανάλυση SWOT

Στην ενότητα αυτή παρουσιάζεται η ανάλυση SWOT (Strengths, Weaknesses, Opportunities, Threats) του συστήματος. Η ανάλυση SWOT αποτελεί ένα εργαλείο στρατηγικού σχεδιασμού, το οποίο βοηθά να συλλέξουμε κάποιες πληροφορίες σχετικές με το σύστημά μας. Οι πληροφορίες αυτές σχετίζονται με τα δυνατά και αδύνατα σημεία και με τις ευκαιρίες και τις απειλές. Συλλέγοντας αυτές τις πληροφορίες καθίσταται ευκολότερος ο καθορισμός των απαιτήσεων και των μελλοντικών επεκτάσεων του συστήματος.

### Δυνατά Σημεία (Strengths)

- Σχεδιάστηκε με πρωταρχικό στόχο την άμεση βοήθεια σε έναν ασθενή καρδιακού επεισοδίου με σκοπό να του σώσει τη ζωή.
- Ενθαρρύνει τον εθελοντισμό.
- Προωθεί την συνεργασία ΕΚΑΒ-εθελοντών με κοινό σκοπό την βοήθεια στον συνάνθρωπο και την αλληλεγγύη.
- Το σύστημα έχει μηδενικό κόστος στη λειτουργία του.
- Παρέχει πρόσβαση από οποιαδήποτε υπολογιστική μηχανή ή κινητή συσκευή

### Αδυναμίες (Weaknesses)

- Το σύστημα λειτουργεί μόνο με την χρήση διαδικτύου.
- Δεν έχει δοκιμαστεί σε πραγματικές συνθήκες..

### Ευκαιρίες (Opportunities)

- Είναι το μοναδικό σύστημα που συνδέει ιατρούς, περιθάλποντες και εθελοντές πολίτες με σκοπό την σωστή χρήση των απινιδωτών για έγκαιρη επέμβαση σε περιπτώσεις καρδιακών περιστατικών.

### Απειλές (Threats)

- Η κύρια απειλή που εμφανίζεται είναι ο ολοένα και αυξανόμενος όγκος πληροφοριών που μπορεί να προκύψουν στο σύστημα κατά τη διάρκεια λειτουργίας του.
- Η πιθανότητα κλοπής των απινιδωτών. Για την απειλή αυτή οι απινιδωτές θα βρίσκονται μέσα σε ειδικά κελιά, καθώς και θα ενημερώνεται ο διαχειριστής του συστήματος όταν βρίσκονται εκτός θέσης.

## 5.5 Μελλοντικές επεκτάσεις

Ως το βασικό πλεονέκτημα του συστήματος μπορεί να θεωρηθεί η δυνατότητα βελτίωσης του. Αυτό επιτυγχάνεται με την προσθήκη, νέων χαρακτηριστικών στην εφαρμογή του συστήματος με τη προσθήκη νέων λειτουργιών. Παρακάτω παρατίθενται ενδεικτικά μερικές από τις επεκτάσεις που μπορούν να βελτιώσουν σημαντικά το σύστημα.

- **Βοηθητικό ηλεκτρονικό εγχειρίδιο πρώτων βοηθειών:** Μια σημαντική προσθήκη θα ήταν οδηγίες σχετικά με την αντιμετώπιση ενός καρδιακού επεισοδίου, με βήμα προς βήμα καθοδήγηση. Επιπρόσθετα θα μπορούσαν αυτές οι οδηγίες να είναι και σε μορφή βίντεο για να είναι πιο κατανοητές.
- **Πλοήγηση στους χάρτες της εφαρμογής:** Μια επέκταση στη κινητή εφαρμογή η οποία θα ήταν σημαντική στη διεπαφή χρήστη θα ήταν η χρήση των χαρτών όχι μόνο για τα σημεία τα οποία είναι εγκατεστημένοι οι απινιδωτές, αλλά και για πλοήγηση σε αυτούς στα πλαίσια της εφαρμογής χωρίς να χρειάζεται η ανακατεύθυνση στους χάρτες του κινητού.

- **Ζωντανή παρακολούθηση (Live Tracking) των Εθελοντών στον χάρτη από τον διαχειριστή:** Μια ακόμα λειτουργία η οποία θα μπορούσε να ήταν βοηθητική τόσο σε επίπεδο διεπαφής χρήστη, όσο και στην υλοποίηση του αρχικού στόχου της διπλωματικής εργασίας, θα ήταν η ζωντανή παρακολούθηση των εθελοντών που ανταποκρίθηκαν στο περιστατικό στα πλαίσια της διαδικτυακής εφαρμογής.

## 5.6 Συμπεράσματα

Τα συμπεράσματα που μπορούν να εξαχθούν είναι πολλά και ενδιαφέροντα και σχετίζονται άμεσα με τα οφέλη χρήσης του συστήματος. Τα οφέλη από τον επιτυχή σχεδιασμό και την υλοποίηση μιας εφαρμογής στον τομέα της υγείας αναμένονται πολλά και σημαντικά. Είναι γεγονός ότι, λόγω της ραγδαίας εξέλιξης της τεχνολογίας και της άμεσης επιρροής της στην καθημερινότητα των ανθρώπων, συστήματα όπως αυτό, μπορούν να συνδράμουν σημαντικά στην διευκόλυνση και απλούστευση διαδικασιών που κατά τα άλλα είναι χρονοβόρες και περίπλοκες.

## 5.7 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο που είναι και το τελευταίο, συνοψίστηκαν ο τρόπος σχεδίασης και υλοποίησης του συστήματος, έγινε αναφορά στα προβλήματα που προέκυψαν σε αυτό. Επίσης παρουσιάστηκαν οι μετρικές κώδικα του συστήματος στο σύνολο του. Παρουσιάστηκαν επιπλέον οι μελλοντικές επεκτάσεις που θα μπορούσαν να υλοποιηθούν. Τέλος, αναφέρθηκαν τα συμπεράσματα που εξήχθησαν.

## Βιβλιογραφία

[1] Αντίστοιχη εφαρμογή 1: AED Locator.

Ιστότοπος: <https://play.google.com/store/apps/details?id=au.com.aedlocator.android&hl=en>,  
Απρίλιος, 2019

[2] Αντίστοιχη εφαρμογή 2: Permis de sauver. Ιστότοπος:

<https://play.google.com/store/apps/details?id=com.permisdesauver>, Απρίλιος 2019

[3] Αντίστοιχη εφαρμογή 3: AED Alert. Ιστότοπος:

<https://play.google.com/store/apps/details?id=com.stanglobal.stan.cpr>, Απρίλιος 2019

[4] Αντίστοιχη εφαρμογή 4: Staying Alive. Ιστότοπος:

<https://play.google.com/store/apps/details?id=com.mobilehealth.cardiac>, Απρίλιος 2019

[5] Αντίστοιχη εφαρμογή 5: DAEdove. Ιστότοπος:

<https://play.google.com/store/apps/details?id=it.squadranautica.daedove>, Απρίλιος 2019

[6] Αντίστοιχη εφαρμογή 6: Responders. Ιστότοπος:

<https://play.google.com/store/apps/details?id=org.aedmap.swissheart>, Απρίλιος 2019

[7] Αντίστοιχη εφαρμογή 7: SAUV life. Ιστότοπος:

<https://play.google.com/store/apps/details?id=com.sauvlife.app>, Απρίλιος 2019

[8] Προγραμματισμός διαδικτύου (Web Development). Ιστότοπος:

[https://en.wikipedia.org/wiki/Web\\_development](https://en.wikipedia.org/wiki/Web_development), Απρίλιος 2019

[9] Υπολογιστής Εξυπηρετητής (Web Server). Ιστότοπος:

[https://developer.mozilla.org/enUS/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/enUS/docs/Learn/Common_questions/What_is_a_web_server),  
Απρίλιος 2019

[10] Database Management System (DBMS). Ιστότοπος:

<https://en.wikipedia.org/wiki/Database>, Απρίλιος 2019

[11] HyperText Markup Language (HTML). Ιστότοπος:

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp), Απρίλιος 2019

[12] Cascading Style Sheets (CSS). Ιστότοπος: <https://www.w3schools.com/css/>, Απρίλιος  
2019

[13] Javascript (JS). Ιστότοπος: [https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

[US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript), Απρίλιος 2019

- [14] Mysql. Ιστότοπος: <https://searchoracle.techtarget.com/definition/MySQL>, Απρίλιος 2019
- [15] PHP. Ιστότοπος: <http://php.net/docs.php>, Απρίλιος 2019
- [16] Slim. Ιστότοπος: <https://www.slimframework.com/>, Απρίλιος 2019
- [17] Framework. Ιστότοπος: [https://en.wikipedia.org/wiki/Software\\_framework](https://en.wikipedia.org/wiki/Software_framework), Απρίλιος 2019
- [18] Laravel. Ιστότοπος: <https://laravel.com/docs/5.7>, Απρίλιος 2019
- [19] Material-ui. Ιστότοπος: <https://material-ui.com/>, Απρίλιος 2019
- [20] jQuery. Ιστότοπος: <https://www.w3schools.com/jquery/default.asp>, Απρίλιος 2019
- [21] React. Ιστότοπος: [https://en.wikipedia.org/wiki/React\\_\(JavaScript\\_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)), Απρίλιος 2019
- [22] Single Page Application (SPA). Ιστότοπος: [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application), Απρίλιος 2019
- [23] Application Programming Interface (API). Ιστότοπος: [https://en.wikipedia.org/wiki/Application\\_programming\\_interface](https://en.wikipedia.org/wiki/Application_programming_interface), Απρίλιος 2019
- [24] Fetch. Ιστότοπος: [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch), Απρίλιος 2019
- [25] Redux. Ιστότοπος: <https://redux.js.org/introduction/getting-started>, Απρίλιος 2019
- [26] Webpack. Ιστότοπος: <https://webpack.js.org/concepts/>, Απρίλιος 2019
- [27] Babel. Ιστότοπος: <https://babeljs.io/docs/en/>, Απρίλιος 2019
- [28] Pusher API. Ιστότοπος: <https://pusher.com/docs>, Απρίλιος 2019
- [29] Google Maps. Ιστότοπος: <https://developers.google.com/maps/documentation/>, Απρίλιος 2019
- [30] React Native. Ιστότοπος: <https://facebook.github.io/react-native/>, Απρίλιος 2019



## Παράρτημα

Στο παρόν τμήμα, παρουσιάζονται οι βοηθητικές οδηγίες που πρέπει να ακολουθήσει κάποιος, προκειμένου να εγκαταστήσει την διαδικτυακή εφαρμογή στο δικό του σύστημα (localhost) και να την χρησιμοποιεί από εκεί ή την εγκατάσταση του σε shared hosting provider.

### Οδηγίες εγκατάστασης σε τοπικό υπολογιστή (localhost)

Για να επιτευχθεί σωστή εγκατάσταση αρχικά ο χρήστης πρέπει να κατεβάσει από το github με link [https://github.com/alxmrdrd](https://github.com/alxmrdrd/Heartbit-Api) τα εξής project. Το πρώτο είναι το Heartbit-Api με url <https://github.com/alxmrdrd/Heartbit-Api> που απευθύνεται στον πηγαίο κώδικα του backend και το δεύτερο είναι ο πηγαίος κώδικας της διαδικτυακής εφαρμογής Heartbit με url <https://github.com/alxmrdrd/heartbit>. Στη συνέχεια, μέσω terminal ο χρήστης πρέπει να πλοηγηθεί αρχικά στο project του api και να πληκτρολογήσει το ακόλουθο script “./server”. Αξίζει να σημειωθεί ότι απαιτείται το λογισμικό XAMP για την ενεργοποίηση της MySQL και του Apache. Τελευταίο βήμα που είναι απαραίτητο, είναι η πλοήγηση του χρήστη στο φάκελο του Heartbit και να εκτελέσει την ακόλουθη εντολή “`REACT_APP_URL=http://localhost:8080 REACT_APP_GOOGLE_MAPS_API_KEY="INSERT YOUR GOOGLE API KEY" npm start`”. Αμέσως η διαδικτυακή εφαρμογή θα ανοίξει σε μια νέα καρτέλα στον προκαθορισμένο φυλλομετρητή του συστήματος.

### Οδηγίες κινητής εφαρμογής

Για να έχει την δυνατότητα κάποιος να δοκιμάσει την εφαρμογή δεν μένει παρά να κατεβάσει στο κινητό του είτε android είτε iOS την εφαρμογή του Expo. Έπειτα αρκεί να σκανάρει μέσω του κινητού του αν είναι σε iOS περιβάλλον ή μέσω της εφαρμογής του Expo αν είναι σε android το ακόλουθο QRcode. Αμέσως το Expo θα αποκτήσει ένα εκτελέσιμο build της εφαρμογής.



## Οδηγίες εγκατάστασης σε shared hosting provider

Για να επιτευχθεί σωστή εγκατάσταση, απαιτείται πρώτα η global εγκατάσταση στο hosting των composer και node. Ωστόσο επειδή η εγκατάσταση σε hosting provider είναι αρκετά πιο περίπλοκη και είναι σίγουρο ότι θα προκύψουν σφάλματα στην πορεία, παρακάτω θα περιγραφούν τα γενικά βήματα που ακολουθούνται πάντα:

1. Πραγματοποίηση βήματος 3 από προηγούμενη εγκατάσταση σε localhost.
2. Χωρισμός public φακέλου και τοποθέτηση των εσωτερικών αρχείων στον κατάλογο public\_html.
3. Επεξεργασία των αρχείων index.php και server.php ώστε να βρίσκουν τα κατάλληλα paths.
4. Εισαγωγή με cli στο hosting μέσω ssh και εισαγωγή εντολών από βήμα.