



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΦΑΡΜΟΓΗ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ
ΕΝΤΟΠΙΣΜΟΥ, ΠΛΟΗΓΗΣΗΣ ΠΡΟΣ ΚΑΙ
ΧΡΗΣΗΣ ΑΠΙΝΙΔΩΤΩΝ

ΑΡΓΥΡΙΟΥ ΙΩΑΝΝΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΣΤΑΜΟΥΛΗΣ ΓΕΩΡΓΙΟΣ
Καθηγητής

ΣΥΝΕΠΙΒΛΕΠΩΝ

ΤΖΙΡΙΤΑΣ ΝΙΚΟΛΑΟΣ
ΠΔ 407/80

Λαμία Φεβρουάριος 2020



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΦΑΡΜΟΓΗ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ
ΕΝΤΟΠΙΣΜΟΥ, ΠΛΟΗΓΗΣΗΣ ΠΡΟΣ ΚΑΙ
ΧΡΗΣΗΣ ΑΠΙΝΙΔΩΤΩΝ

ΑΡΓΥΡΙΟΥ ΙΩΑΝΝΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΣΤΑΜΟΥΛΗΣ ΓΕΩΡΓΙΟΣ
Καθηγητής

ΣΥΝΕΠΙΒΛΕΠΩΝ

ΤΖΙΡΙΤΑΣ ΝΙΚΟΛΑΟΣ
ΠΔ 407/80

Λαμία Φεβρουάριος 2020

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφική. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 24/02/2020

Ο Δηλών



(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»



Περίληψη

Η ανακοπή καρδιάς αποτελεί ένα από τα σοβαρότερα προβλήματα υγείας, του οποίου η σημασία είναι αντιληπτή ακόμη και από κάποιον χωρίς ιατρικές γνώσεις. Κατά τη διάρκεια της ανακοπής η καρδιά παύει να επιτελεί τον σκοπό της, δηλαδή αδυνατεί να στείλει αίμα στον εγκέφαλο και στα υπόλοιπα όργανα του σώματος. Αυτό συμβαίνει επειδή, είτε ξαφνικά σταματάει να πάλλεται είτε πάλλεται γρηγορότερα, αλλά με ακανόνιστο τρόπο. Η δεύτερη περίπτωση είναι αναστρέψιμη εάν αντιμετωπιστεί εντός λίγων λεπτών. Η αντιμετώπιση πραγματοποιείται αρχικά με την καρδιοαναπνευστική αναζωογόνηση (ΚΑΡΠΑ), ένα μέτρο το οποίο εξασφαλίζει περιορισμένη ποσότητα οξυγονωμένου αίματος στον εγκέφαλο, και στη συνέχεια με χρήση αυτόματου εξωτερικού απινιδωτή (ΑΕΑ). Παρά το γεγονός ότι η Ελλάδα έχει θεσπίσει νόμο ο οποίος υποχρεώνει την εγκατάσταση απινιδωτών σε δημόσιους χώρους στους οποίους συνυπάρχει μεγάλο πλήθος ατόμων και ταυτόχρονα ενθαρρύνει την εθελοντική εγκατάσταση ΑΕΑ σε οποιαδήποτε υπηρεσία, οργανισμό ή ιδιωτική επιχείρηση το επιθυμεί, δεν υπάρχει κάποια ενιαία πηγή από την οποία οι πολίτες μπορούν να ενημερωθούν για τα σημεία ύπαρξης απινιδωτών. Σκοπός της εργασίας αυτής είναι η αξιοποίηση της τεχνολογίας προκειμένου να συνδράμει στη λύση του προβλήματος αυτού με την ανάπτυξη μιας εφαρμογής για κινητές συσκευές. Μέσω της εφαρμογής αυτής, οι χρήστες μπορούν να ενημερωθούν για τα κοντινότερα σημεία ύπαρξης ΑΕΑ με βάση τη τρέχουσα τοποθεσία τους, να λάβουν αναλυτικές πληροφορίες για τον καθένα, καθώς και να πλοηγηθούν με φωνητικές οδηγίες άμεσα σε αυτούς. Επίσης, θεωρήθηκε χρήσιμη η προσθήκη δυνατότητας επέκτασης στη βάση δεδομένων. Γι' αυτόν τον λόγο, διατίθεται ένα περιβάλλον μέσω του οποίου οποιοσδήποτε φορέας ή οργανισμός μπορεί να εγγραφεί και στη συνέχεια να προσθέσει νέους απινιδωτές ή να τροποποιήσει τους ήδη υπάρχοντες που του ανήκουν. Τέλος, το REST API μέσω του οποίου γίνεται η άντληση των δεδομένων, κατασκευάστηκε με τέτοιο τρόπο έτσι ώστε να είναι εύκολα και δημόσια προσβάσιμο, ώστε να προσφέρει σε άλλους προγραμματιστές τη δυνατότητα ανάπτυξης νέων ιδεών, χωρίς την ανάγκη συλλογής των απαραίτητων δεδομένων εκ νέου από τις διάφορες πηγές.

Λέξεις – Κλειδιά

απινιδωτής, API, εφαρμογή



Περιεχόμενα

Περίληψη	vii
Περιεχόμενα.....	viii
Κατάλογος Εικόνων	x
Κατάλογος Πινάκων.....	xi
1. Εισαγωγή.....	1
1.1 Αυτόματος Εξωτερικός Απινιδωτής (ΑΕΑ)	1
1.2 Ευρωπαϊκός αριθμός κλήσης έκτακτης ανάγκης (112).....	1
2. Το API.....	3
2.1 REST API.....	3
2.2 Strapi	4
2.2.1 Τι είναι το Strapi	4
2.2.2 Χαρακτηριστικά του Strapi	4
2.3 Τύποι περιεχομένου και δομικά στοιχεία.....	5
2.3.1 Τύπος περιεχομένου «Aed».....	5
2.3.2 Τύπος περιεχομένου «AedOwner»	7
2.3.3 Τύπος περιεχομένου «AedManager»	8
2.3.4 Τύπος περιεχομένου «AedModel».....	8
2.3.5 Τύπος περιεχομένου «AedRegion».....	9
2.3.6 Τύπος περιεχομένου «AedRegionalUnit»	10
2.3.7 Τύπος περιεχομένου «AedMunicipality»	10
2.3.8 Δομικό στοιχείο «AedLocation».....	11
2.4 API endpoints	12
2.5 Διαχείριση	15
2.5.1 Σελίδα σύνδεσης διαχειριστή.....	15
2.5.2 Περιβάλλον διαχειριστή	16
2.5.3 Προσθήκη νέου απινιδωτή	17
2.5.4 Ρόλοι και άδειες	17
2.5.5 Δημιουργία νέου χρήστη	18
3. Η εφαρμογή	20
3.1 Flutter	20
3.2 Mapbox	20
3.2.2 Directions API	21
3.3 Android Intents	23
3.4 Πακέτα	24
3.4.1 Πακέτο «location».....	24
3.4.2 Πακέτο «flutter_map»	26
3.4.3 Πακέτο «carousel_slider».....	27
3.4.4 Πακέτο «android_intent».....	27
3.4.5 Πακέτο «call_number».....	27
3.5 Κλάσεις - Μοντέλα	28
3.5.1 Κλάση «Aed».....	28
3.5.2 Κλάση «AedOwner»	28
3.5.3 Κλάση «AedManager»	28



3.5.4 Κλάση «AedModel».....	28
3.5.5 Κλάση «AedLocation».....	28
3.6 Συναρτήσεις.....	29
3.6.1 Συνάρτηση «getAllAeds».....	29
3.6.2 Συνάρτηση «getNearestAed».....	29
3.6.3 Συνάρτηση «navigationIntent».....	29
3.6.4 Συνάρτηση «callIntent».....	30
3.7 Καρτέλα «Χάρτης».....	31
3.7.1 Μενού επιλογών απινιδωτή.....	32
3.7.2 Οθόνη πληροφοριών απινιδωτή.....	34
3.8 Καρτέλα «Οδηγίες χρήσης».....	37
3.9 Κουμπί «έκτακτης ανάγκης».....	38
Ιστογραφία.....	40
Παράρτημα Α: Κώδικας εφαρμογής.....	42
Φάκελος «models».....	43
Αρχείο «aed.dart».....	43
Αρχείο «aed_location.dart».....	46
Αρχείο «aed_manager.dart».....	46
Αρχείο «aed_model.dart».....	46
Αρχείο «aed_owner.dart».....	46
Φάκελος «screens».....	47
Αρχείο «aed_details_screen.dart».....	47
Αρχείο «aed_instructions_screen.dart».....	52
Αρχείο «aed_map_screen.dart».....	56
Αρχείο «tabs_screen.dart».....	60
Κύριος φάκελος.....	63
Αρχείο «future.dart».....	63
Αρχείο «intent.dart».....	65
Αρχείο «main.dart».....	66



Κατάλογος Εικόνων

Εικόνα 2-1 Τύπος περιεχομένου «Aed»	7
Εικόνα 2-2 Τύπος περιεχομένου «AedOwner».....	8
Εικόνα 2-3 Τύπος περιεχομένου «AedManager»	8
Εικόνα 2-4 Τύπος περιεχομένου «AedModel»	9
Εικόνα 2-5 Τύπος περιεχομένου «AedRegion»	9
Εικόνα 2-6 Τύπος περιεχομένου «AedRegionalUnit»	10
Εικόνα 2-7 Τύπος περιεχομένου «AedMunicipality»	11
Εικόνα 2-8 Δομικό στοιχείο «AedLocation»	12
Εικόνα 2-9 Σελίδα σύνδεσης διαχειριστή	15
Εικόνα 2-10 Περιβάλλον διαχειριστή	16
Εικόνα 2-11 Προσθήκη νέου απινιδωτή μέσω του περιβάλλοντος διαχειριστή.....	17
Εικόνα 2-12 Διαθέσιμοι ρόλοι	17
Εικόνα 2-13 Δημιουργία νέου χρήστη μέσω του περιβάλλοντος διαχειριστή	18
Εικόνα 3-14 Στιγμιότυπα οθόνης της καρτέλας «Χάρτης»	31
Εικόνα 3-15 Στιγμιότυπο οθόνης του μενού επιλογών απινιδωτή.....	33
Εικόνα 3-16 Στιγμιότυπα οθόνης της οθόνης πληροφοριών απινιδωτή.....	34
Εικόνα 3-17 Στιγμιότυπα οθόνης της οθόνης πληροφοριών απινιδωτή (2)	36
Εικόνα 3-18 Στιγμιότυπα οθόνης της καρτέλας «Οδηγίες χρήσης»	37
Εικόνα 3-19 Στιγμιότυπο οθόνης του κουμπιού «έκτακτης ανάγκης»	38
Εικόνα 3-20 Στιγμιότυπο οθόνης των επιλογών του κουμπιού «έκτακτης ανάγκης»	38



Κατάλογος Πινάκων

Πίνακας 2-1 Πεδία του τύπου περιεχομένου «Aed»	6
Πίνακας 2-2 Πεδία του τύπου περιεχομένου «AedOwner»	7
Πίνακας 2-3 Πεδία του τύπου περιεχομένου «AedManager»	8
Πίνακας 2-4 Πεδία του τύπου περιεχομένου «AedModel»	9
Πίνακας 2-5 Πεδία του τύπου περιεχομένου «AedRegion»	9
Πίνακας 2-6 Πεδία του τύπου περιεχομένου «AedRegionalUnit»	10
Πίνακας 2-7 Πεδία του τύπου περιεχομένου «AedMunicipality»	11
Πίνακας 2-8 Πεδία του δομικού στοιχείου «AedLocation»	12
Πίνακας 2-9 Διαθέσιμα API endpoint	13
Πίνακας 2-10 Παραδείγματα χρήσης των διαθέσιμων API endpoint	14
Πίνακας 2-11 Υποστηριζόμενα προφίλ του Directions API	22
Πίνακας 2-12 Απαιτούμενοι παράμετροι στα αιτήματα του Directions API	22
Πίνακας 2-13 Συμπεριληφθέντα δεδομένα στις απαντήσεις του Directions API.....	23
Πίνακας 2-14 Παραδείγματα χρήσης Android Intent	24
Πίνακας 2-15 Διαθέσιμες συναρτήσεις στο πακέτο «location».....	26



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*



1. Εισαγωγή

1.1 Αυτόματος Εξωτερικός Απινιδωτής (ΑΕΑ)

Ο αυτόματος εξωτερικός απινιδωτής (ΑΕΑ) [\[1\]\[2\]](#) είναι μια φορητή συσκευή η οποία αποτελείται από έναν μικροεπεξεργαστή, ηλεκτρικά κυκλώματα και αυτοκόλλητα επιθέματα ηλεκτροδίων. Μόλις τα ηλεκτρόδια επικολληθούν στον θώρακα του ασθενή, η συσκευή μπορεί να ξεκινήσει να συλλέγει δεδομένα για τον καρδιακό ρυθμό, όπως ακριβώς ένας κλασικός καρδιογράφος. Αν διαγνωστεί κάποιο πρόβλημα που χρειάζεται απινίδωση (ηλεκτρική εκκένωση), αυτή χορηγείται μέσω των επικολλημένων ηλεκτροδίων στην καρδιά. Η απινίδωση αδρανοποιεί στιγμιαία τη δραστηριότητα της καρδιάς, δίνοντάς της την ευκαιρία να ξαναρχίσει να χτυπά αποτελεσματικά με φυσιολογική ηλεκτρική δραστηριότητα.

Κάποιοι απινιδωτές είναι πλήρως αυτόματοι, ενώ κάποιοι άλλοι απαιτούν την παρέμβαση του χειριστή. Πολλοί μπορούν επίσης να δώσουν γραπτές ή προφορικές οδηγίες για καθοδήγηση της διαδικασίας, καθώς και να καταγράφουν σε εσωτερική μνήμη τις ενέργειές τους.

1.2 Ευρωπαϊκός αριθμός κλήσης έκτακτης ανάγκης (112)

Ο τηλεφωνικός αριθμός 112 [\[3\]\[4\]\[5\]](#) είναι ένας κοινός αριθμός έκτακτης ανάγκης, ο οποίος ισχύει για όλα τα κράτη μέλη της Ευρωπαϊκής Ένωσης και όχι μόνο. Σκοπός του αριθμού αυτού είναι να διευκολύνει την επικοινωνία των πολιτών με τις εθνικές υπηρεσίες έκτακτης ανάγκης. Οποιοσδήποτε βρίσκεται εντός της Ε.Ε. μπορεί να απευθυνθεί στον αριθμό 112 για να ζητήσει βοήθεια ή να αναφέρει κάποιο έκτακτο περιστατικό.

Όλα τα κράτη μέλη της Ε.Ε. είναι υποχρεωμένα να παρέχουν κλήσεις στο 112, χωρίς τηλεφωνική χρέωση, από οποιοδήποτε σταθερό ή κινητό τηλέφωνο και από όλες τις δημόσιες τηλεφωνικές συσκευές. Σε περίπτωση που η περιοχή δεν καλύπτεται από το δίκτυο κινητής τηλεφωνίας του καλούντος, η κλήση πραγματοποιείται μέσω άλλων δικτύων που καλύπτουν την περιοχή (υπηρεσία εθνικής και διεθνούς περιαγωγής).

Το άριστα εκπαιδευμένο προσωπικό μπορεί να ανακατευθύνει άμεσα την κλήση στην υπεύθυνη εθνική υπηρεσία ή να μεσολαβήσει για να μεταφέρει αυτό το αίτημα του καλούντος στις κατάλληλες υπηρεσίες, ανάλογα με την κρισιμότητα της κατάστασης.



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Τέλος, η κλήση στον αριθμό 112 παρέχει τη δυνατότητα εντοπισμού της γεωγραφικής θέσης του καλούντος σε περίπτωση έλλειψης γνώσης αυτής.



2. Το API

2.1 REST API

Το REST (Representational State Transfer) [\[6\]\[7\]](#) είναι ένα αρχιτεκτονικό στυλ λογισμικού το οποίο ορίζει ένα σύνολο από αρχές σχεδίασης διαδικτυακών υπηρεσιών. Όπως είναι φανερό και από το όνομά του, σε μια REST διαδικτυακή υπηρεσία υπάρχει ένας διακομιστής ο οποίος επιστρέφει στον πελάτη μια αναπαράσταση της κατάστασης της ζητούμενης πληροφορίας. Οι διαδικτυακές υπηρεσίες που συμμορφώνονται σύμφωνα με τη τεχνολογία REST ονομάζονται «RESTful» και παρέχουν διαλειτουργικότητα μεταξύ συστημάτων υπολογιστών στο διαδίκτυο.

Συνεπώς, οι RESTful διεπαφές προγραμματισμού εφαρμογών (API) βασίζονται στη τεχνολογία REST και χρησιμοποιούνται για την επικοινωνία μεταξύ εφαρμογών. Η επικοινωνία βασίζεται στο πρωτόκολλο HTTP (Hypertext Transfer Protocol). Συγκεκριμένα, η μέθοδος «GET» που ορίζει το πρωτόκολλο χρησιμοποιείται για την ανάκτηση δεδομένων, η μέθοδος «POST» για τη δημιουργία δεδομένων, η μέθοδος «PUT» για ενημέρωση και η μέθοδος «DELETE» για διαγραφή αυτών.

Η αναπαράσταση των δεδομένων συνήθως γίνεται σε μορφή JSON (JavaScript Object Notation), αλλά υπάρχουν και περιπτώσεις που η αναπαράσταση γίνεται σε μορφή XML (Extensible Markup Language) ή HTML (Hypertext Markup Language). Και οι τρεις προηγούμενες μορφές δεδομένων αφορούν την κωδικοποίηση εγγράφων σε μορφή που είναι ταυτόχρονα κατανοητή από τον άνθρωπο και τους ηλεκτρονικούς υπολογιστές.

Η επικοινωνία με τα API γίνεται μέσω των διάφορων «endpoint» που διαθέτουν. Πρακτικά, τα API endpoint είναι πολλά διαφορετικά URL (Uniform Resource Locator), χάρη στα οποία γίνεται διαχωρισμός μεταξύ των πληροφοριών που μπορεί να προσφέρει το API. Ένα API μπορεί να διαθέτει μόνο ένα ή εκατοντάδες endpoint. Ανάλογα με το ποια είναι η ζητούμενη πληροφορία, το αίτημα πρέπει να προορίζεται προς το ένα συγκεκριμένο που έχει καθοριστεί να επιστρέφει την πληροφορία αυτή.



2.2 Strapi

2.2.1 Τι είναι το Strapi

Το Strapi [\[8\]\[9\]\[10\]\[11\]](#) είναι ένα «headless» σύστημα διαχείρισης περιεχομένου (ΣΔΠ, Content Management System, CMS) ανοιχτού κώδικα βασισμένο στο Node.js - ένα περιβάλλον που επιτρέπει την εκτέλεση κώδικα JavaScript εκτός των περιηγητών ιστού (browsers) για τους οποίους προοριζόταν αρχικά.

Ο όρος «headless» βασίζεται στην ιδέα ύπαρξης μιας διεπαφής για την προσθήκη περιεχομένου και ενός RESTful API που είναι υπεύθυνο για την παροχή του περιεχομένου αυτού. Σε αντίθεση με τα παραδοσιακά ΣΔΠ (CMS), όπως είναι το WordPress, τα headless συστήματα δε διαθέτουν κάποιο μέσο, όπως μια ιστοσελίδα, για άμεση προβολή του περιεχομένου στον χρήστη. Ευθύνη των headless συστημάτων αποτελούν μόνο η αποθήκευση και η παροχή δομημένου περιεχομένου.

Η ιδέα γύρω από τη δημιουργία του Strapi είναι να παρέχει τη δυνατότητα κατασκευής ενός πρακτικού, έτοιμου για χρήση API, σε λίγες ώρες αντί για ολόκληρες εβδομάδες, χάρη στο εύχρηστο περιβάλλον και το πλήρως διαμορφώσιμο API που διαθέτει.

2.2.2 Χαρακτηριστικά του Strapi

Μερικά από τα κύρια και καινοτόμα χαρακτηριστικά του Strapi είναι τα εξής:

- **Πανεύκολη διαχείριση περιεχομένου:** Μετά τον καθορισμό της δομής του περιεχομένου, η διαχείρισή του μπορεί να γίνει από οποιονδήποτε χρήστη, με ή χωρίς εμπειρία, χάρη στο εύχρηστο γραφικό περιβάλλον διαχείρισής του.
- **Ευέλικτη δομή περιεχομένου:** Το Strapi προσφέρει εκτός από τους βασικούς τύπους πεδίων, και άλλους πιο σύνθετους, όπως τύπο πολυμέσων, email και κωδικών πρόσβασης. Πιο έμπειροι χρήστες μπορούν ακόμη να κατασκευάσουν δικούς τους τύπους περιεχομένου ανάλογα με τις ανάγκες τους.
- **Επαλήθευση και άδειες:** Χάρη στην υποστήριξη ρόλων, η πρόσβαση μιας ομάδας σε κάποιο API endpoint μπορεί να χορηγηθεί ή να ανακληθεί πανεύκολα και γρήγορα.
- **Διαμορφώσιμο API:** Με πρόσβαση στα αρχεία κώδικα του project, το API που κατασκευάζεται μπορεί να προσφέρει στους χρήστες του περισσότερα και ιδιαίτερα χαρακτηριστικά.



- **Μεταφόρτωση αρχείων:** Υπάρχει υποστήριξη μεταφόρτωσης όλων των τύπων αρχείων, τόσο στον ίδιο διακομιστή που φιλοξενείται το Strapi όσο και σε εξωτερικούς παρόχους, όπως AWS S3, Cloudflare, κλπ.
- **Βάσεις δεδομένων:** Είτε πρέπει να χρησιμοποιηθεί σχεσιακή είτε μη σχεσιακή βάση δεδομένων, το Strapi έχει κατασκευαστεί να είναι συνεργάσιμο και με τους δύο τύπους. Προσφέρεται υποστήριξη για τις βάσεις δεδομένων MongoDB, PostgreSQL, MySQL, MariaDB και για την SQLite η οποία χρησιμοποιείται από προεπιλογή.
- **Ενσωματωμένο σύστημα email:** Με κάθε καινούριο project, εγκαθίσταται από προεπιλογή και ένας διακομιστής SMTP (Simple Mail Transfer Protocol) για διαχείριση των email (εγγραφής, ανάκτησης κωδικού πρόσβασης, κλπ.), χωρίς την ανάγκη επιπλέον κώδικα.

2.3 Τύποι περιεχομένου και δομικά στοιχεία

Πριν δημιουργηθεί το περιεχόμενο που θα παρέχει το API, πρέπει πρώτα να δημιουργηθούν οι διάφοροι πιθανοί τύποι περιεχομένου στους οποίους θα βασιστεί. Για το συγκεκριμένο project, έχουν δημιουργηθεί οι παρακάτω τύποι περιεχομένου και τα δομικά στοιχεία.

2.3.1 Τύπος περιεχομένου «Aed»

Περιγραφή: Συγκρατεί όλες τις χρήσιμες πληροφορίες για τις καταχωρήσεις απινιδωτών.

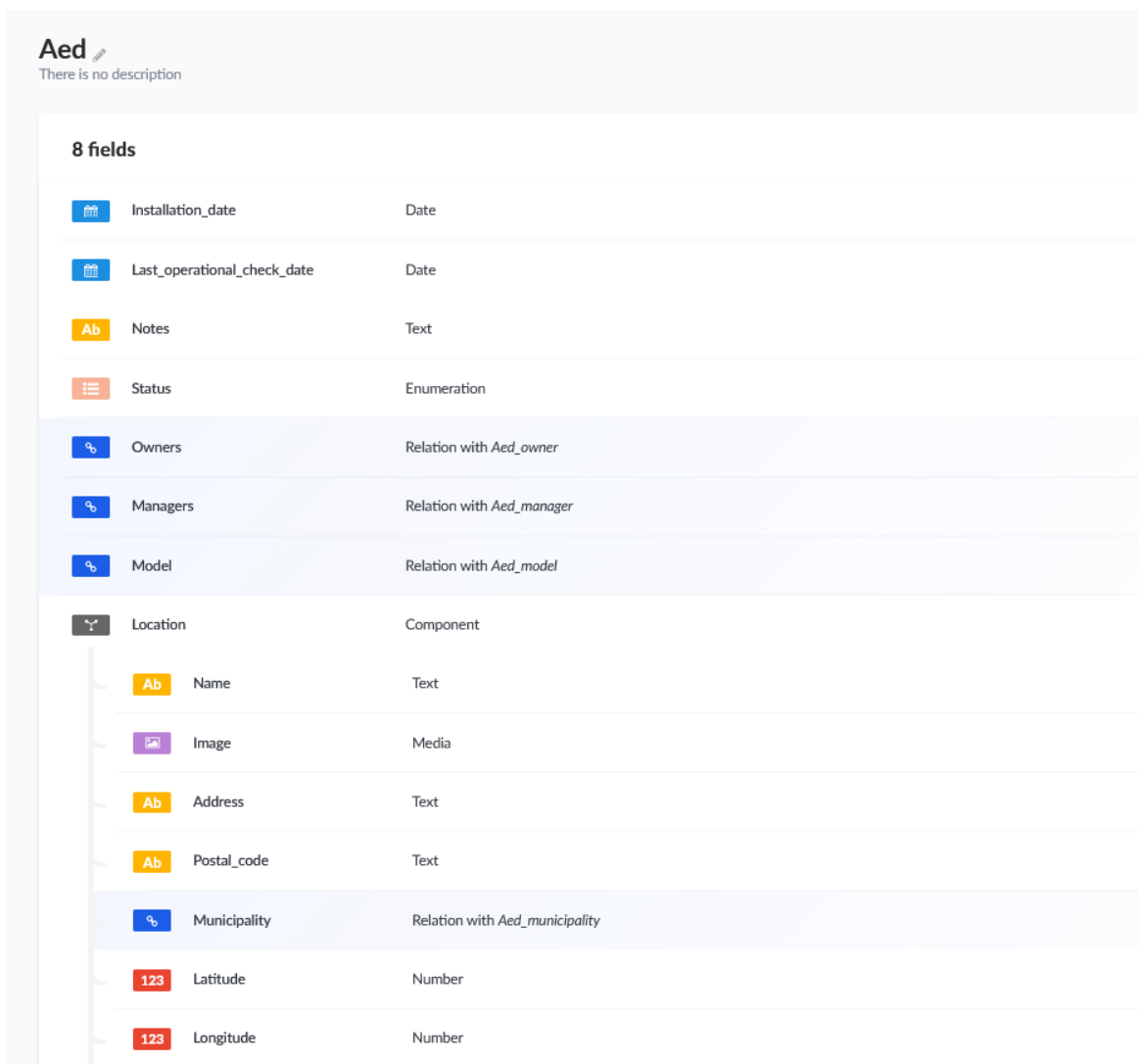
Πεδία:

Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Ημερομηνία εγκατάστασης	Ημερομηνία	OXI	OXI
Ημερομηνία τελευταίου ελέγχου λειτουργίας	Ημερομηνία	OXI	OXI
Σημειώσεις	Κείμενο	OXI	OXI



Κατάσταση	Απαρίθμηση Δυνατές τιμές: available, in_use, out_of_order	ΝΑΙ	ΟΧΙ
Ιδιοκτήτες	Συσχέτιση με τον τύπο περιεχομένου «AedOwner» Σχέση: Ένα προς πολλά. Ένας απινιδωτής μπορεί να έχει περισσότερους από έναν ιδιοκτήτες.	ΝΑΙ	ΟΧΙ
Υπεύθυνοι	Συσχέτιση με τον τύπο περιεχομένου «AedManager» Σχέση: Ένα προς πολλά. Ένας απινιδωτής μπορεί να έχει περισσότερους από έναν υπεύθυνους για αυτόν.	ΝΑΙ	ΟΧΙ
Μοντέλο	Συσχέτιση με τον τύπο περιεχομένου «AedModel» Σχέση: Ένα προς ένα. Ένας απινιδωτής αντιστοιχίζεται με ένα μόνο μοντέλο.	ΝΑΙ	ΟΧΙ
Τοποθεσία	Δομικό στοιχείο «AedLocation»	ΝΑΙ	ΟΧΙ

Πίνακας 2-1 Πεδία του τύπου περιεχομένου «Aed»



Εικόνα 2-1 Τύπος περιεχομένου «Aed»

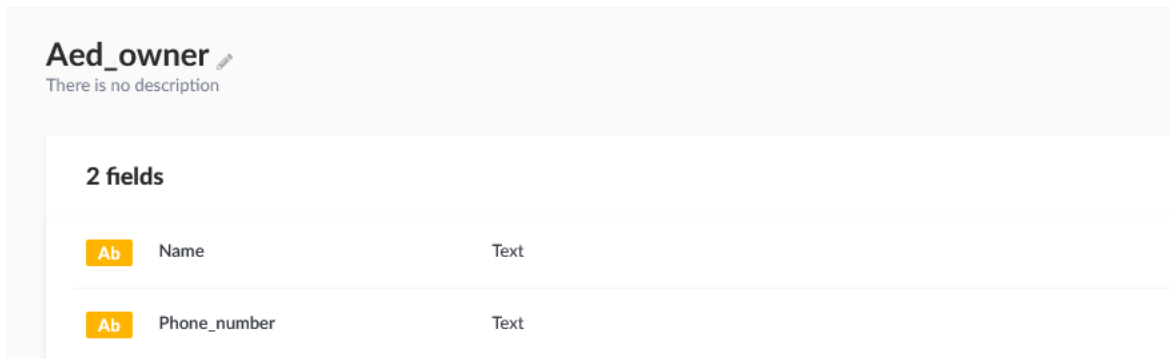
2.3.2 Τύπος περιεχομένου «AedOwner»

Περιγραφή: Αναπαριστά τους ιδιοκτήτες ενός ή περισσότερων απινιδωτών.

Πεδία:

Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Όνομα	Κείμενο	ΝΑΙ	ΝΑΙ
Αριθμός τηλέφωνο	Κείμενο	ΝΑΙ	ΝΑΙ

Πίνακας 2-2 Πεδία του τύπου περιεχομένου «AedOwner»



Εικόνα 2-2 Τύπος περιεχομένου «AedOwner»

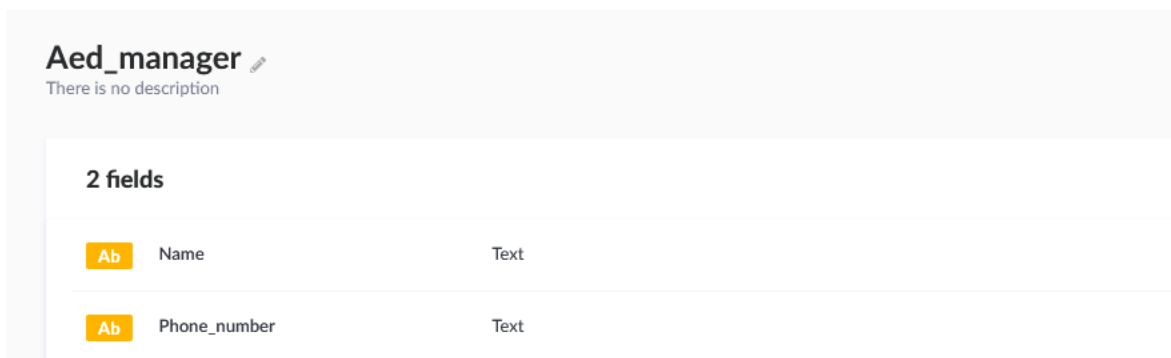
2.3.3 Τύπος περιεχομένου «AedManager»

Περιγραφή: Αναπαριστά τους υπεύθυνους ενός ή περισσότερων απινιδωτών.

Πεδία:

Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Όνομα	Κείμενο	ΝΑΙ	ΝΑΙ
Αριθμός τηλεφώνου	Κείμενο	ΝΑΙ	ΝΑΙ

Πίνακας 2-3 Πεδία του τύπου περιεχομένου «AedManager»



Εικόνα 2-3 Τύπος περιεχομένου «AedManager»

2.3.4 Τύπος περιεχομένου «AedModel»

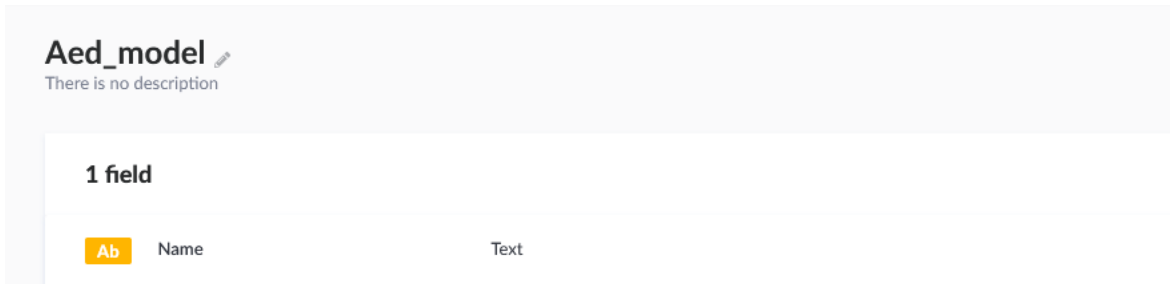
Περιγραφή: Αναπαριστά τα διάφορα μοντέλα απινιδωτών.

Πεδία:



Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Όνομα	Κείμενο	ΝΑΙ	ΝΑΙ

Πίνακας 2-4 Πεδία του τύπου περιεχομένου «AedModel»



Εικόνα 2-4 Τύπος περιεχομένου «AedModel»

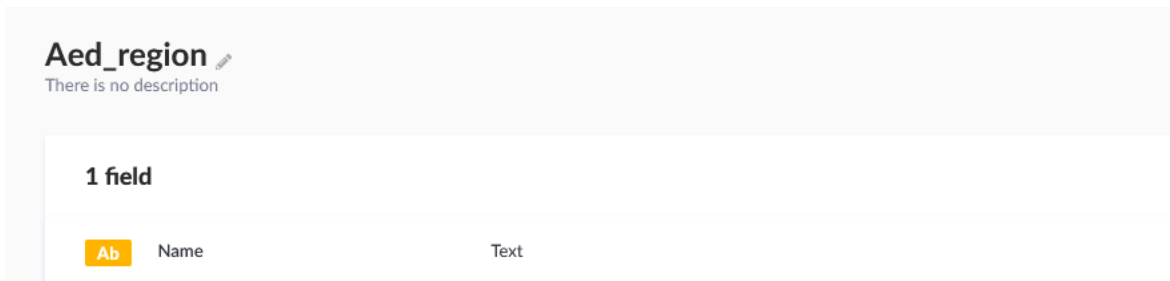
2.3.5 Τύπος περιεχομένου «AedRegion»

Περιγραφή: Αναπαριστά τις περιφέρειες της Ελλάδας (Ανατολική Μακεδονία και Θράκη, Θεσσαλία, κλπ.).

Πεδία:

Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Όνομα	Κείμενο	ΝΑΙ	ΝΑΙ

Πίνακας 2-5 Πεδία του τύπου περιεχομένου «AedRegion»



Εικόνα 2-5 Τύπος περιεχομένου «AedRegion»



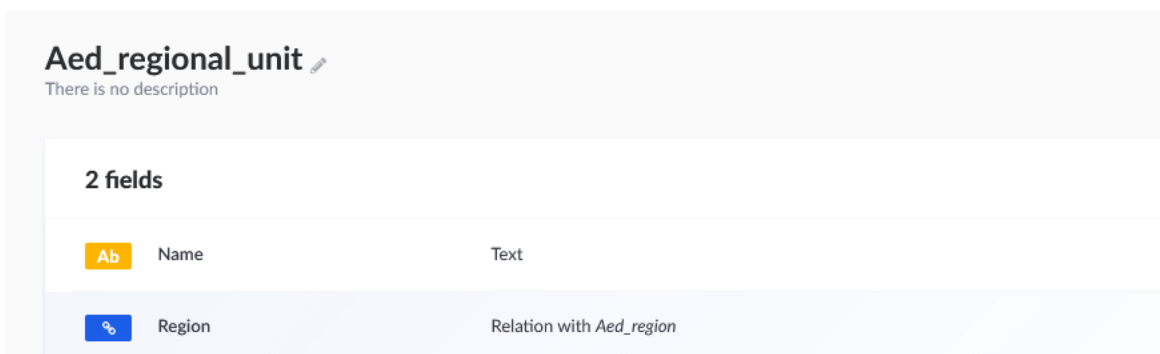
2.3.6 Τύπος περιεχομένου «AedRegionalUnit»

Περιγραφή: Αναπαριστά τις περιφερειακές ενότητες της Ελλάδας (Καβάλα, Ξάνθη, Λάρισα, Μαγνησία, κλπ.).

Πεδία:

Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Όνομα	Κείμενο	ΝΑΙ	ΝΑΙ
Περιφέρεια	Συσχέτιση με τον τύπο περιεχομένου «AedRegion» Σχέση: Ένα προς ένα. Μία περιφερειακή ενότητα ανήκει σε μία μόνο περιφέρεια.	ΝΑΙ	ΟΧΙ

Πίνακας 2-6 Πεδία του τύπου περιεχομένου «AedRegionalUnit»



Εικόνα 2-6 Τύπος περιεχομένου «AedRegionalUnit»

2.3.7 Τύπος περιεχομένου «AedMunicipality»

Περιγραφή: Αναπαριστά τους δήμους της Ελλάδας (Ελασσόνα, Λάρισα, Φάρσαλα, κλπ.).

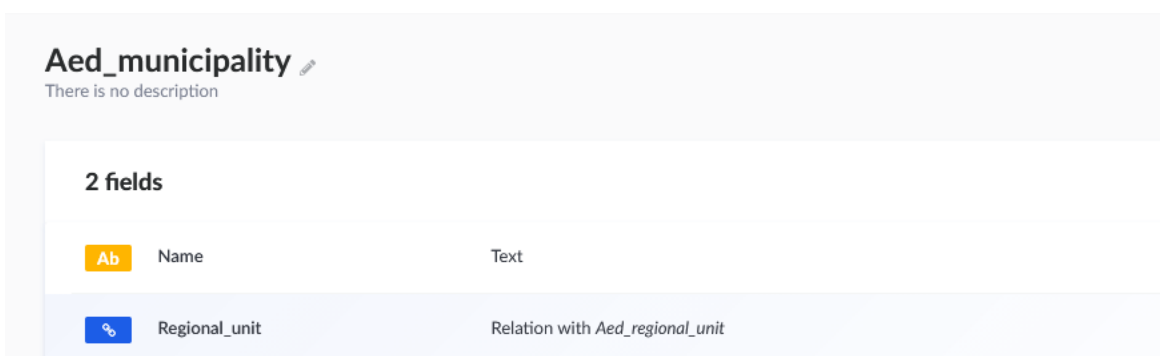
Πεδία:

Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Όνομα	Κείμενο	ΝΑΙ	ΝΑΙ



Περιφερειακή ενότητα	Συσχέτιση με τον τύπο περιεχομένου «AedRegionalUnit» Σχέση: Ένα προς ένα. Ένας δήμος ανήκει σε μία μόνο περιφερειακή ενότητα.	ΝΑΙ	ΟΧΙ
-------------------------	---	-----	-----

Πίνακας 2-7 Πεδία του τύπου περιεχομένου «AedMunicipality»



Εικόνα 2-7 Τύπος περιεχομένου «AedMunicipality»

2.3.8 Δομικό στοιχείο «AedLocation»

Περιγραφή: Αναπαριστά τις τοποθεσίες – σημεία εγκατάστασης των απινιδωτών.

Πεδία:

Όνομα	Τύπος	Απαιτούμενο πεδίο	Μοναδική τιμή πεδίου
Όνομα	Κείμενο	ΝΑΙ	ΟΧΙ
Εικόνα	Πολυμέσο	ΟΧΙ	ΝΑΙ
Διεύθυνση	Κείμενο	ΝΑΙ	ΟΧΙ
Ταχυδρομικός κώδικας	Κείμενο	ΝΑΙ	ΟΧΙ
Δήμος	Συσχέτιση με τον τύπο περιεχομένου «AedMunicipality»	ΝΑΙ	ΟΧΙ



	Σχέση: Ένα προς ένα. Μία τοποθεσία αντιστοιχίζεται με έναν μόνο δήμο.		
Γεωγραφικό πλάτος	Αριθμός (πραγματικός) Ελάχιστη τιμή: -90 Μέγιστη τιμή: 90	NAI	OXI
Γεωγραφικό μήκος	Αριθμός (πραγματικός) Ελάχιστη τιμή: -180 Μέγιστη τιμή: 180	NAI	OXI

Πίνακας 2-8 Πεδία του δομικού στοιχείου «AedLocation»

Location
There is no description

7 fields

- Ab** Name Text
- Image** Image Media
- Ab** Address Text
- Ab** Postal_code Text
- Municipality** Municipality Relation with Aed_municipality
- 123** Latitude Number
- 123** Longitude Number

Εικόνα 2-8 Δομικό στοιχείο «AedLocation»

2.4 API endpoints

Το Strapi δημιουργεί από προεπιλογή τα παρακάτω API endpoint. Μετά από κάποιο αίτημα προς αυτά, αποστέλλεται πίσω στον πελάτη μία αναπαράσταση της ζητούμενης πληροφορίας σε μορφή JSON.



Μέθοδος αιτήματος	Διαδρομή	Περιγραφή
GET	/ {τύπος περιεχομένου}	Επιστρέφει μια λίστα με όλα τα καταχωρημένα δεδομένα του καθορισμένου τύπου περιεχομένου.
GET	/ {τύπος περιεχομένου} / : {αναγνωριστικό}	Επιστρέφει την καταχώρηση του καθορισμένου τύπου περιεχομένου που έχει το παρεχόμενο αναγνωριστικό.
GET	/ {τύπος περιεχομένου} / count	Επιστρέφει το πλήθος των καταχωρήσεων του καθορισμένου τύπου περιεχομένου.
POST	/ {τύπος περιεχομένου}	Δημιουργεί μία νέα καταχώρηση στον καθορισμένο τύπο περιεχομένου.
DELETE	/ {τύπος περιεχομένου} / : {αναγνωριστικό}	Διαγράφει την καταχώρηση του καθορισμένου τύπου περιεχομένου που έχει το παρεχόμενο αναγνωριστικό.
PUT	/ {τύπος περιεχομένου} / : {αναγνωριστικό}	Ενημερώνει την καταχώρηση του καθορισμένου τύπου περιεχομένου που έχει το παρεχόμενο αναγνωριστικό.

Πίνακας 2-9 Διαθέσιμα API endpoint

Παραδείγματα:



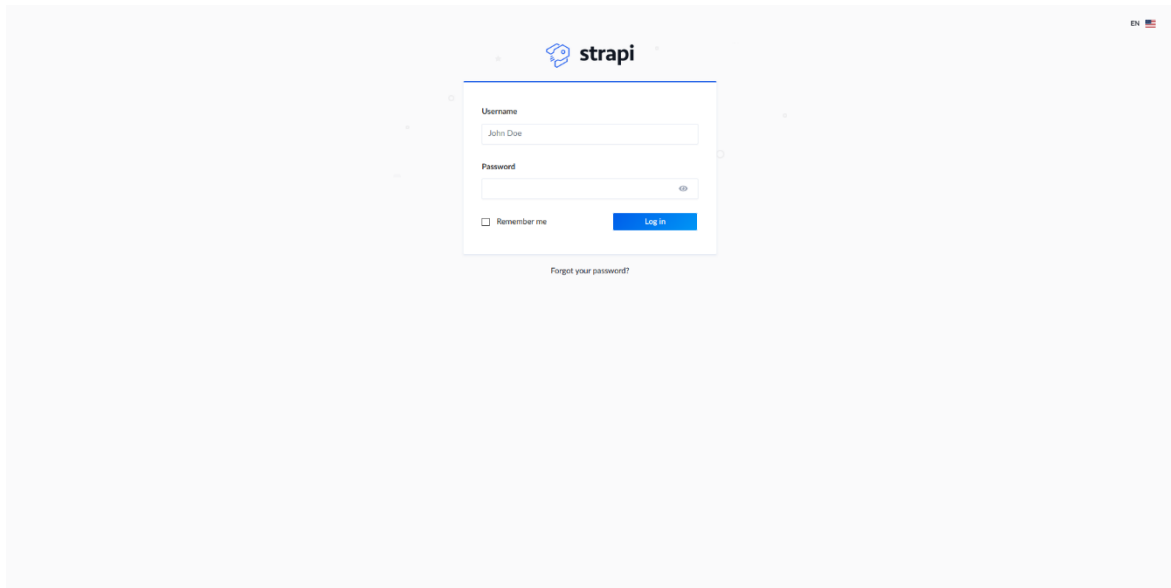
Μέθοδος αιτήματος	Διαδρομή	Περιγραφή
GET	/aeds	Επιστρέφει μια λίστα με όλους τους καταχωρημένους απινιδωτές.
GET	/aed-models/1	Επιστρέφει το καταχωρημένο μοντέλο απινιδωτή με αναγνωριστικό «1».
GET	/aed-owners/count	Επιστρέφει το πλήθος των καταχωρημένων ιδιοκτητών απινιδωτών.
POST	/aeds	Δημιουργεί μια νέα καταχώρηση απινιδωτή. Σε κάθε τέτοιο αίτημα πρέπει να συμπεριλαμβάνονται οι απαιτούμενες πληροφορίες.
DELETE	/aed-managers/1	Διαγράφει τον καταχωρημένο υπεύθυνο απινιδωτών με αναγνωριστικό «1».
PUT	/aeds/5	Ενημερώνει τον καταχωρημένο απινιδωτή με αναγνωριστικό «5» σύμφωνα με τις νέες πληροφορίες που συμπεριλαμβάνονται στο αίτημα.

Πίνακας 2-10 Παραδείγματα χρήσης των διαθέσιμων API endpoint



2.5 Διαχείριση

2.5.1 Σελίδα σύνδεσης διαχειριστή

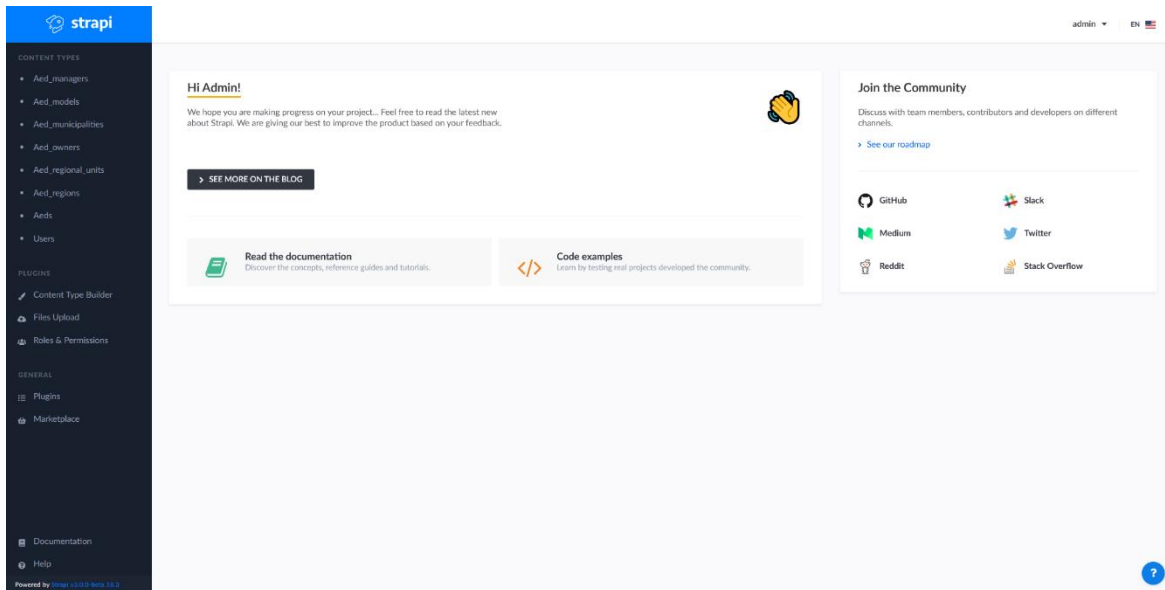


Εικόνα 2-9 Σελίδα σύνδεσης διαχειριστή

Η σελίδα σύνδεσης διαχειριστή είναι προσβάσιμη προσθέτοντας στον τελικό σύνδεσμο τη διαδρομή «**/admin**» μετά τη διεύθυνση του διακομιστή που φιλοξενεί το Strapi. Μετά από επιτυχή σύνδεση, ο χρήστης ανακατευθύνεται στο [περιβάλλον διαχειριστή](#) του Strapi.



2.5.2 Περιβάλλον διαχειριστή



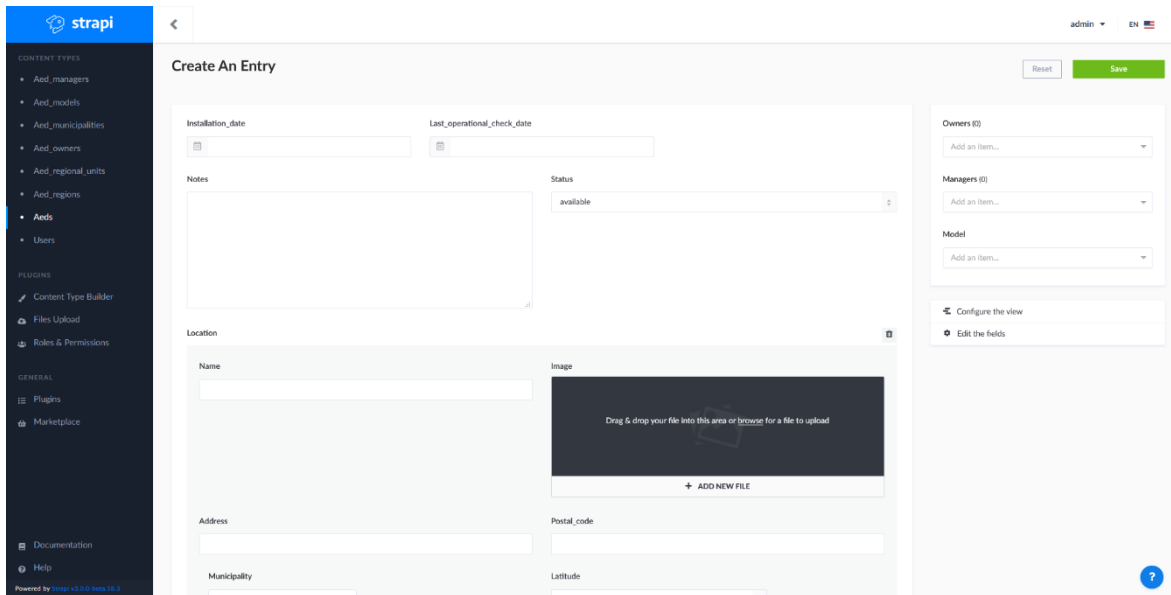
Εικόνα 2-10 Περιβάλλον διαχειριστή

Μέσω του περιβάλλοντος διαχειριστή ένας διαχειριστής έχει τη δυνατότητα να:

- **διαχειριστεί** τους τύπους περιεχομένου, τα δομικά στοιχεία, καθώς και το περιεχόμενο που δημιουργείται με χρήση αυτών,
- **μεταφορτώσει** νέα αρχεία ή να **επεξεργαστεί** τα ήδη υπάρχοντα
- **διαμορφώσει** τους ρόλους και τις άδειες των χρηστών,
- **εγκαταστήσει** ή να **ρυθμίσει** τα εγκατεστημένα πρόσθετα



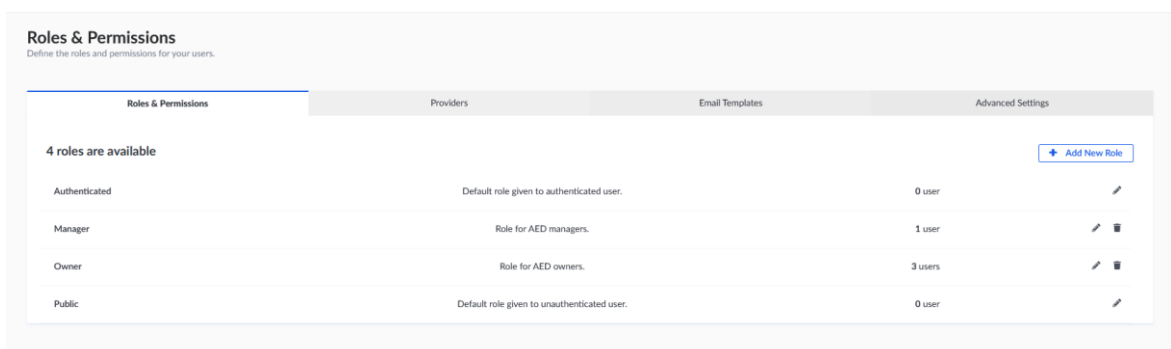
2.5.3 Προσθήκη νέου απινιδωτή



Εικόνα 2-11 Προσθήκη νέου απινιδωτή μέσω του περιβάλλοντος διαχειριστή

Αφού δημιουργηθεί ο κατάλληλος τύπος περιεχομένου (στη συγκεκριμένη περίπτωση ο [τύπος περιεχομένου «Aed»](#)), οι διαχειριστές έχουν τη δυνατότητα προσθήκης νέου περιεχομένου μέσω της επιλογής που προστίθεται στο μενού της αριστερής μεριάς του [περιβάλλοντος διαχειριστή](#) με το όνομα του τύπου.

2.5.4 Ρόλοι και άδειες



Εικόνα 2-12 Διαθέσιμοι ρόλοι

Από προεπιλογή υπάρχουν δύο (2) έτοιμοι δημιουργημένοι ρόλοι από το Strapi:

- **Επαληθευμένος (Authenticated) χρήστης:** Οι χρήστες οι οποίοι έχουν επαληθεύσει την εγγραφή τους μέσω email.



- **Δημόσια (Public) χρήση:** Οι χρήστες του API που επιθυμούν να το χρησιμοποιήσουν χωρίς να έχουν εγγραφεί.

Και στους δύο (2) παραπάνω ρόλους έχει επιτραπεί μόνο η χρήση της μεθόδου «GET» στα διαθέσιμα [API endpoint](#) για λόγους ασφαλείας.

Επιπλέον, έχουν δημιουργηθεί οι εξής ρόλοι με τις ακόλουθες άδειες:

- **Ιδιοκτήτης (Owner) απινιδωτή:** Ο ρόλος αυτός έχει δημιουργηθεί για τους ιδιοκτήτες απινιδωτών, στους οποίους επιτρέπεται η δυνατότητα χρήσης όλων των διαθέσιμων μεθόδων στα διαθέσιμα [API endpoint](#), προκειμένου να μπορούν να εισάγουν, διαγράφουν και να ενημερώνουν τους απινιδωτές.
- **Υπεύθυνος (Manager) απινιδωτή:** Αυτός ο ρόλος απευθύνεται στους υπεύθυνους απινιδωτών, στους οποίους δίνεται άδεια χρήσης μόνο των μεθόδων «GET» και «PUT», ώστε να τους επιτρέπεται να ενημερώνουν τα στοιχεία των ήδη καταχωρημένων απινιδωτών, αλλά ταυτόχρονα να τους στερείται το δικαίωμα εισαγωγής νέων ή διαγραφή υπαρχόντων.

2.5.5 Δημιουργία νέου χρήστη

The screenshot shows the Strapi admin interface for creating a new user. The left sidebar lists content types like 'Aid_managers', 'Aid_models', etc. The main area is titled 'Create An Entry' and contains a form with the following fields: Username, Email, Password, and Confirmed Password. There are also toggle switches for 'Blocked' and 'Role' (set to 'Add an item...'). Buttons for 'Reset' and 'Save' are visible at the top right of the form.

Εικόνα 2-13 Δημιουργία νέου χρήστη μέσω του περιβάλλοντος διαχειριστή

Από προεπιλογή, υπάρχουν δύο (2) τρόποι εγγραφής ενός νέου χρήστη:

- **Μέσω του API endpoint «/auth/local/register»:** Με χρήση της μεθόδου «POST» στο endpoint αυτό, ο χρήστης έχει τη δυνατότητα δημιουργίας λογαριασμού στην υπηρεσία, συμπεριλαμβάνοντας υποχρεωτικά στο αίτημα το όνομα χρήστη, το e-mail και τον κωδικό πρόσβασης που επιθυμεί.
- **Μέσω του [περιβάλλοντος διαχειριστή](#):** Οι διαχειριστές της υπηρεσίας έχουν τη δυνατότητα δημιουργίας λογαριασμών μέσω της φόρμας που απεικονίζεται στην παραπάνω εικόνα.



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Προς το παρόν, θεωρήθηκε σκόπιμη η απενεργοποίηση των εγγραφών μέσω του API endpoint «**/auth/local/register**», καθώς η δημιουργία λογαριασμού δεν ωφελεί σε τίποτα έναν απλό χρήστη. Ταυτόχρονα, ένας ιδιοκτήτης ή υπεύθυνος απινιδωτή θα ήταν υποχρεωμένος να αναμένει την ανάθεση του ρόλου του από κάποιον διαχειριστή. Έτσι, ο μόνος τρόπος που απομένει είναι η δημιουργία λογαριασμών χρηστών από τους διαχειριστές, οι οποίοι μπορούν να αναθέσουν άμεσα τον κατάλληλο ρόλο στον νέο χρήστη.



3. Η εφαρμογή

3.1 Flutter

Το Flutter [\[12\]\[13\]\[14\]](#) είναι ένα ανοιχτού κώδικα κιτ ανάπτυξης λογισμικού (SDK) διαθέσιμο από τη Google. Χρησιμοποιείται κυρίως για την ανάπτυξη «native» εφαρμογών για κινητές συσκευές με λειτουργικό σύστημα Android ή iOS με βάση έναν κοινό κώδικα. Με άλλα λόγια, χρησιμοποιείται μία γλώσσα προγραμματισμού και ένας κώδικας για την παραγωγή δύο διαφορετικών εφαρμογών. Ακόμη, παρέχεται σε πρώιμο στάδιο υποστήριξη για την ανάπτυξη εφαρμογών που προορίζονται για λειτουργικά συστήματα σταθερών υπολογιστών (Windows, Mac, Linux), αλλά και για τη δημιουργία ιστοσελίδων. Ο όρος «native» σημαίνει ο τελικός κώδικας της εφαρμογής που παράγεται είναι γραμμένος στην αντίστοιχη γλώσσα προγραμματισμού της κάθε πλατφόρμας (π.χ. Java ή Kotlin για Android και Objective-C ή Swift για iOS).

Το Flutter αποτελείται από δύο (2) βασικά συστατικά στοιχεία:

- Ένα **κιτ ανάπτυξης λογισμικού (SDK)**: Μία συλλογή από εργαλεία που βοηθούν τους προγραμματιστές στην ανάπτυξη των εφαρμογών. Σε αυτά συμπεριλαμβάνονται και εργαλεία μεταγλώττισης για την παραγωγή εγγενή κώδικα της εκάστοτε πλατφόρμας προορισμού.
- Ένα **framework**: Μία συλλογή από επαναχρησιμοποιήσιμα γραφικά στοιχεία διεπαφής χρήστη (κουμπιά, φόρμες κειμένου, κλπ.) για τη διαμόρφωση του γραφικού περιβάλλοντος σύμφωνα με τις ανάγκες της αναπτυσσόμενης εφαρμογής.

Η γλώσσα προγραμματισμού που χρησιμοποιεί το κιτ είναι η Dart, μία σχετικά νέα γλώσσα προγραμματισμού με αυξημένη δημοτικότητα, η οποία αναπτύσσεται επίσης από τη Google σε συνεχή συνεργασία με την ομάδα του Flutter.

3.2 Mapbox

Το Mapbox [\[15\]\[16\]](#) είναι ένας ευρέως διαδεδομένος πάροχος προσαρμοσμένων διαδικτυακών χαρτών και είναι ο βασικός αντίπαλος του Google Maps. Οι ποικίλες διεπαφές προγραμματισμού εφαρμογών (API), τα κιτ ανάπτυξης λογισμικού (SDK) και τα συνεχώς ενημερωμένα δεδομένα χαρτών που διαθέτει, προσφέρουν στους προγραμματιστές τα εργαλεία για την κατασκευή ιστοσελίδων και εφαρμογών με



καλύτερες δυνατότητες χαρτογράφησης, πλοήγησης και αναζήτησης, σε αντίθεση με τους ανταγωνιστές του.

Τα δεδομένα τους προέρχονται κυρίως από πηγές ανοιχτών δεδομένων, όπως η NASA και το OpenStreetMap [\[17\]](#) - έναν χάρτη που αναπτύσσεται εξολοκλήρου από τη συνεισφορά μιας τεράστιας κοινότητας εθελοντών.

Το Mapbox επιλέχθηκε ως πάροχος των χαρτών στην [καρτέλα «Χάρτης»](#) της εφαρμογής σε συνεργασία με το [πακέτο «flutter_map»](#), λόγω των αξιόλογων API που προσφέρει δωρεάν προς χρήση για μεγάλο πλήθος αιτημάτων.

3.2.2 Directions API

Το Directions API [\[18\]](#) είναι ένα από τα πολλά API που διατίθεται από την υπηρεσία πλοήγησης του [Mapbox](#). Χάρη σε αυτό οι χρήστες μπορούν να:

- υπολογίσουν τις βέλτιστες διαδρομές οδήγησης, βάδισης και ποδηλασίας, λαμβάνοντας υπόψη την κυκλοφορία και πιθανά περιστατικά που έχουν συμβεί στον δρόμο,
- λάβουν οδηγίες πλοήγησης από την αφετηρία μέχρι τον προορισμό τους,
- συμπεριλάβουν στη διαδρομή τους ενδιάμεσες στάσεις

Σε αυτό το API υποστηρίζονται τέσσερα (4) διαφορετικά προφίλ δρομολόγησης:

Προφίλ	Περιγραφή
mapbox/driving-traffic	Δρομολόγηση με χρήση αυτοκινήτου. Αυτό το προφίλ λαμβάνει υπόψη τις τρέχουσες συνθήκες κυκλοφορίας καθώς και το ιστορικό τους για την αποφυγή καθυστερήσεων.
mapbox/driving	Δρομολόγηση με χρήση αυτοκινήτου. Αυτό το προφίλ επιστρέφει τις ταχύτερες διαδρομές προτιμώντας τους δρόμους υψηλής ταχύτητας, όπως αυτοκινητοδρόμους.
mapbox/walking	Δρομολόγηση με βάδιση. Αυτό το προφίλ δείχνει τη συντομότερη διαδρομή χρησιμοποιώντας πεζοδρόμια και μονοπάτια.
mapbox/cycling	Δρομολόγηση με χρήση ποδηλάτου. Αυτό το προφίλ δείχνει διαδρομές που είναι σύντομες και ασφαλέστερες για τους



	ποδηλάτες, αποφεύγοντας αυτοκινητοδρόμους και προτιμώντας δρόμους με ποδηλατοδρόμους.
--	---

Πίνακας 2-11 Υποστηριζόμενα προφίλ του Directions API

Το τρέχον endpoint του Directions API είναι το εξής: `/directions/v5/{προφίλ δρομολόγησης}/{γεωγραφικές συντεταγμένες}`

Στα αιτήματα απαιτείται να συμπεριλαμβάνονται οι εξής δύο (2) παράμετροι:

Παράμετρος	Περιγραφή
profile	Το επιθυμητό προφίλ δρομολόγησης. Ένα από τα τέσσερα προφίλ που περιγράφηκαν προηγουμένως.
coordinates	Μία λίστα από γεωγραφικές συντεταγμένες που πρέπει να συμπεριληφθούν στη διαδρομή, χωρισμένες μεταξύ τους με ελληνικό ερωτηματικό (;). Η σειρά τους καθορίζει τη σειρά επίσκεψης.

Πίνακας 2-12 Απαιτούμενοι παράμετροι στα αιτήματα του Directions API

Οι απαντήσεις στα αιτήματα δίνονται σε μορφή JSON και διαθέτουν τα εξής δεδομένα:

Στοιχείο	Περιγραφή
code	Μια συμβολοσειρά που χαρακτηρίζει την κατάσταση της απάντησης. Στις κανονικές, έγκυρες απαντήσεις η συμβολοσειρά έχει τιμή «Ok».
waypoints	Μία λίστα με όλες τις ενδιάμεσες στάσεις που ζητήθηκε να συμπεριληφθούν στη διαδρομή. Η σειρά τους στη λίστα είναι ανάλογη με τη σειρά που τους δόθηκε στο αίτημα. Κάθε στοιχείο της λίστας συμπεριλαμβάνει τις γεωγραφικές συντεταγμένες της στάσης, καθώς και το όνομα του δρόμου ή της διεύθυνσης στην οποία αντιστοιχούν.
routes	Μία λίστα δύο (2) το μέγιστο διαδρομών που μπορεί να ακολουθήσει ο χρήστης. Η λίστα είναι ταξινομημένη κατά φθίνουσα σειρά σύστασης και περιέχει τον εκτιμώμενο χρόνο ταξιδιού, την απόσταση της



	διαδρομής, καθώς και άλλες πιθανώς χρήσιμες πληροφορίες για τους ταξιδιώτες.
--	--

Πίνακας 2-13 Συμπεριληφθέντα δεδομένα στις απαντήσεις του Directions API

Αιτήματα στο Directions API γίνονται από το [κουμπί «έκτακτης ανάγκης»](#) και κατ' επέκταση από τη [συνάρτηση «getNearestAed»](#) για να ληφθούν οι αποστάσεις των απινιδωτών από τη τρέχουσα τοποθεσία του χρήστη, προκειμένου να βρεθεί ο κοντινότερος απινιδωτής. Το προφίλ δρομολόγησης που χρησιμοποιείται στην εφαρμογή είναι το «**mapbox/walking**» και η απόσταση λαμβάνεται από στοιχείο «**routes**» της απάντησης.

Ο τελικός σύνδεσμος των αιτημάτων διαμορφώνεται ως εξής:

`https://api.mapbox.com/directions/v5/mapbox/walking/{γεωγραφικό μήκος τοποθεσίας χρήστη},{γεωγραφικό πλάτος τοποθεσίας χρήστη};{γεωγραφικό μήκος απινιδωτή},{γεωγραφικό πλάτος απινιδωτή}?access_token={κλειδί πρόσβασης}`

3.3 Android Intents

Οι «προθέσεις» (intents) ^[19] αποτελούν ένα χαρακτηριστικό του λειτουργικού συστήματος Android και είναι μια αφηρημένη περιγραφή ενεργειών που πρέπει να εκτελεστούν. Χρησιμοποιούνται για την εκκίνηση μιας δραστηριότητας (της ίδιας ή κάποιας άλλης εγκατεστημένης εφαρμογής) ή για την επικοινωνία με μια υπηρεσία που εκτελείται στο παρασκήνιο.

Η βασική δομή μιας πρόθεσης αποτελείται από:

- την **ενέργεια (action)**, δηλαδή τη βασική ενέργεια που πρέπει να εκτελεστεί (ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, κλπ.)
- τα **δεδομένα (data)**, δηλαδή τα δεδομένα πάνω στα οποία θα εκτελεστεί η καθορισμένη ενέργεια (action)

Παραδείγματα προθέσεων:

Ενέργεια	Δεδομένα	Περιγραφή
----------	----------	-----------



ACTION_VIEW	content://contacts/people/1	Εμφανίζει πληροφορίες για την καταχώρηση με αναγνωριστικό «1» της λίστας επαφών.
ACTION_DIAL	content://contacts/people/1	Ανοίγει την προεπιλεγμένη εφαρμογή διαχείρισης τηλεφωνικών κλήσεων με τον τηλεφωνικό αριθμό της καταχώρησης με αναγνωριστικό «1» της λίστας επαφών συμπληρωμένο.
ACTION_VIEW	tel:123	Ανοίγει την προεπιλεγμένη εφαρμογή διαχείρισης τηλεφωνικών κλήσεων με τον αριθμό «123» συμπληρωμένο.
ACTION_DIAL	tel:123	Ανοίγει την προεπιλεγμένη εφαρμογή διαχείρισης τηλεφωνικών κλήσεων με τον αριθμό «123» συμπληρωμένο.
ACTION_EDIT	content://contacts/people/1	Ανοίγει την καταχώρηση με αναγνωριστικό «1» της λίστας επαφών για επεξεργασία των πληροφοριών της.
ACTION_VIEW	content://contacts/people/	Εμφανίζει μια λίστα με όλες τις καταχωρήσεις της λίστας επαφών.

Πίνακας 2-14 Παραδείγματα χρήσης Android Intent

Προθέσεις χρησιμοποιούνται στις συναρτήσεις [«navigationIntent»](#) και [«callIntent»](#) οι οποίες αναλύονται παρακάτω.

3.4 Πακέτα

3.4.1 Πακέτο «location»

Το πακέτο «location» [\[20\]](#) αναλαμβάνει τη λήψη τοποθεσίας της συσκευής τόσο στο λειτουργικό σύστημα Android όσο και στο iOS.

Συναρτήσεις:



Τύπος επιστροφής	Όνομα	Περιγραφή
Future<bool>	requestPermission	Ζητάει από τον χρήστη την άδεια εντοπισμού τοποθεσίας από την εφαρμογή. Επιστρέφει τιμή αληθείας ανάλογα με την απάντηση του χρήστη.
Future<bool>	hasPermission	Επιστρέφει τιμή αληθείας ανάλογα με τη τρέχουσα κατάσταση της άδειας εντοπισμού τοποθεσίας από την εφαρμογή.
Future<bool>	serviceEnabled	Επιστρέφει τιμή αληθείας ανάλογα με το αν η υπηρεσία τοποθεσίας είναι ενεργοποιημένη ή αν ο χρήστης την απενεργοποίησε χειροκίνητα.
Future<bool>	requestService	Εμφανίζει ένα παράθυρο διαλόγου ειδοποίησης το οποίο ζητάει από τον χρήστη να ενεργοποιήσει την υπηρεσία τοποθεσίας.
Future<bool>	changeSettings	Επιτρέπει την αλλαγή ρυθμίσεων των μελλοντικών αιτημάτων. Η παράμετρος « accuracy » περιγράφει την ακρίβεια των αιτημάτων λήψης τοποθεσίας. Η παράμετρος « interval » αφορά το χρονικό διάστημα μεταξύ των ενημερώσεων τοποθεσίας σε μιλιδευτερόλεπτα. Η παράμετρος « distanceFilter » αφορά την ελάχιστη μετατόπιση σε μέτρα που πρέπει να πραγματοποιηθεί ώστε να ληφθεί η επόμενη ενημέρωση.



Future<LocationData>	getLocation	Επιτρέπει τη στιγμιαία λήψη της τοποθεσίας του χρήστη. Ζητάει από τον χρήστη την άδεια εντοπισμού τοποθεσίας εάν δεν έχει ήδη χορηγηθεί.
Stream<LocationData>	onLocationChanged	Επιστρέφει τη νέα τοποθεσία του χρήστη κάθε φορά που πραγματοποιείται ενημέρωση σύμφωνα με τα κριτήρια που ορίζονται με τη συνάρτηση «changeSettings».

Πίνακας 2-15 Διαθέσιμες συναρτήσεις στο πακέτο «location»

Από το πακέτο αυτό χρησιμοποιείται συγκεκριμένα η συνάρτηση «**getLocation**» κάθε φορά που γίνεται εναλλαγή στην [καρτέλα «Χάρτης»](#), ώστε ο χρήστης να είναι ενήμερος για τη θέση του στον χάρτη και να μπορεί να εκτιμήσει την απόστασή του από κοντινούς απινιδωτές.

3.4.2 Πακέτο «flutter_map»

Το πακέτο «flutter_map» ^[21] αποτελεί την υλοποίηση της βιβλιοθήκης Leaflet ^[22] στη γλώσσα προγραμματισμού Dart για χρήση εντός των [Flutter](#) εφαρμογών. Η βιβλιοθήκη Leaflet είναι η κυρίαρχη βιβλιοθήκη ανοιχτού κώδικα γραμμένη στη γλώσσα προγραμματισμού JavaScript για την αναπαράσταση διαδραστικών χαρτών σε παντός τύπου εφαρμογές, είτε προορίζονται για προσωπικούς υπολογιστές είτε για κινητές συσκευές. Η βιβλιοθήκη διαθέτει όλα τα χαρακτηριστικά χαρτογράφησης τα οποία μπορεί να χρειαστεί κάποιος προγραμματιστής για να υλοποιήσει τις ιδέες του, ενώ παράλληλα είναι εύκολα επεκτάσιμη με πρόσθετα τρίτων, καθιστώντας την εξαιρετικά ευέλικτη.

Χρησιμοποιώντας ως πάροχο χαρτών το [Mapbox](#), το πακέτο αυτό είναι υπεύθυνο για την αναπαράσταση του χάρτη, καθώς και για την εμφάνιση των εικονιδίων που υποδεικνύουν την τοποθεσία του χρήστη και των απινιδωτών στην [καρτέλα «Χάρτης»](#).



3.4.3 Πακέτο «carousel_slider»

Το πακέτο αυτό ^[23] επιτρέπει την παρουσίαση ενός συνόλου εικόνων ή άλλων γραφικών στοιχείων στον χρήστη με όμορφο και διαδραστικό τρόπο. Η αναπαράσταση των βημάτων της διαδικασίας απινίδωσης με εικόνες και κείμενο στην [καρτέλα «Οδηγίες χρήσης»](#) γίνεται χάρη στις δυνατότητες που προσφέρει το πακέτο αυτό.

3.4.4 Πακέτο «android_intent»

Το πακέτο «android_intent» ^[24] διευκολύνει τη δημιουργία και διαχείριση [Android Intent](#) εντός των [Flutter](#) εφαρμογών. Χρήση αυτού του πακέτου γίνεται εντός της [συνάρτησης «navigationIntent»](#) η οποία εσωτερικά επιχειρεί να εκκινήσει κάποια εφαρμογή χαρτών της Google σε λειτουργία πλοήγησης με σκοπό την πλοήγηση του χρήστη στον επιλεγμένο απινιδωτή.

3.4.5 Πακέτο «call_number»

Το πακέτο αυτό ^[25] διαθέτει μία και μοναδική συνάρτηση, ονόματι «callNumber», η οποία δέχεται ως παράμετρο έναν τηλεφωνικό αριθμό σε μορφή συμβολοσειράς. Η ασύγχρονη αυτή συνάρτηση εσωτερικά δημιουργεί ένα [Android Intent](#) με ενέργεια (action) την «ACTION_CALL» και δεδομένα (data) την εξής συμβολοσειρά: «tel:{αριθμός τηλεφώνου}». Το κείμενο «tel:» που προηγείται αποτελεί ένα URI (Uniform Resource Identifier), όπως είναι το «https://» και το «mailto:», το οποίο ακολουθείται από έναν τηλεφωνικό αριθμό και έχει καθιερωθεί να χρησιμοποιείται για ενέργειες σχετικές με τη διαχείριση τηλεφωνικών κλήσεων.

Συγκεκριμένα, το λειτουργικό σύστημα Android προσφέρει δύο (2) ενέργειες (actions) για το συγκεκριμένο URI. Η ενέργεια «ACTION_DIAL» ανοίγει την υπεύθυνη εφαρμογή της συσκευής για τη διαχείριση τηλεφωνικών κλήσεων με τον τηλεφωνικό συμπληρωμένο, ώστε ο χρήστης να μπορεί άμεσα να τον καλέσει. Αντίθετα, η ενέργεια «ACTION_CALL» εκτελεί άμεσα την κλήση χωρίς την διαμεσολάβηση του χρήστη. Η τελευταία είναι και αυτή που χρησιμοποιεί το πακέτο και κατ' επέκταση η εφαρμογή. Προϋπόθεση για να λειτουργήσει η ενέργεια «ACTION_CALL» είναι να έχει χορηγηθεί στην εφαρμογή η άδεια «CALL_PHONE» από τον χρήστη.



3.5 Κλάσεις - Μοντέλα

3.5.1 Κλάση «Aed»

Η κλάση «Aed» αναπαριστά τον [τύπο περιεχομένου «Aed»](#) εντός της εφαρμογής.

Διαθέτει δύο (2) getters, οι οποίοι λειτουργούν σαν συναρτήσεις. Ο πρώτος («**statusText**») είναι υπεύθυνος για την μετατροπή των κωδικοποιημένων λέξεων που αφορούν την κατάσταση των απινιδωτών («available», «in_use» ή «out_of_order») σε ελληνικό κείμενο. Ο δεύτερος («**statusColor**») επιστρέφει το κατάλληλο χρώμα (πράσινο, πορτοκαλί ή κόκκινο) ανάλογα με την κατάσταση του απινιδωτή, το οποίο χρησιμοποιείται για τον χρωματισμό των καρδιών που αναπαριστούν τα σημεία ύπαρξης απινιδωτών στην [καρτέλα «Χάρτης»](#).

Επίσης, διαθέτει έναν ειδικό constructor ονόματι «**fromJson**» ο οποίος καλείται μέσα από τη [συνάρτηση «getAllAeds»](#) μετά την επικοινωνία της με το API. Ο constructor αυτός δέχεται ως παράμετρο τα δεδομένα του [τύπου περιεχομένου «Aed»](#) σε μορφή JSON και φροντίζει να τα μετατρέψει σε μορφή κατανοητή για την εφαρμογή, δηλαδή σε αντικείμενα των διαθέσιμων κλάσεων.

3.5.2 Κλάση «AedOwner»

Αναπαριστά τον [τύπο περιεχομένου «AedOwner»](#) εντός της εφαρμογής.

3.5.3 Κλάση «AedManager»

Αναπαριστά τον [τύπο περιεχομένου «AedManager»](#) εντός της εφαρμογής.

3.5.4 Κλάση «AedModel»

Αναπαριστά τον [τύπο περιεχομένου «AedModel»](#) εντός της εφαρμογής.

3.5.5 Κλάση «AedLocation»

Αναπαριστά το [δομικό περιεχομένου «AedLocation»](#) εντός της εφαρμογής.



3.6 Συναρτήσεις

3.6.1 Συνάρτηση «getAllAeds»

Η συνάρτηση αυτή βρίσκεται στο αρχείο [«future.dart»](#) και κάνει χρήση των διαθέσιμων [API endpoint](#) του [τύπου περιεχομένου «Aed»](#). Πιο συγκεκριμένα, χρησιμοποιεί το endpoint με διαδρομή [«/aeds»](#) και μέθοδο αιτήματος «GET», το οποίο επιστρέφει μία λίστα με όλους τους απινιδωτές της βάσης δεδομένων σε μορφή JSON. Στη συνέχεια καλεί τον constructor [«fromJson»](#) της [κλάσης «Aed»](#) για τη δημιουργία μίας λίστας αντικειμένων της τελευταίας κλάσης.

Χρήση αυτής της συνάρτησης γίνεται κατά το άνοιγμα της εφαρμογής ή κατά την εναλλαγή του χρήστη στην [καρτέλα «Χάρτης»](#), ώστε να συλλεχθούν όλα τα δεδομένα των διαθέσιμων απινιδωτών από το API.

3.6.2 Συνάρτηση «getNearestAed»

Η συνάρτηση «getNearestAed» αποτελεί περιεχόμενο του αρχείου [«future.dart»](#) και είναι υπεύθυνη για την επιλογή του απινιδωτή που βρίσκεται πιο κοντά στην τοποθεσία του χρήστη. Αυτό το επιτυγχάνει με χρήση του [Directions API](#) του [Mapbox](#).

3.6.3 Συνάρτηση «navigationIntent»

Η συνάρτηση «navigationIntent» συμπεριλαμβάνεται στο αρχείο [«intent.dart»](#), είναι ασύγχρονη και είναι υπεύθυνη για τη δημιουργία ενός [Android Intent](#) με ενέργεια (action) την [«ACTION_VIEW»](#) και δεδομένα (data) την εξής συμβολοσειρά: [«google.navigation:q={γεωγραφικό πλάτος απινιδωτή},{γεωγραφικό μήκος απινιδωτή}»](#). Το κείμενο [«google.navigation:»](#) που προηγείται, αποτελεί ένα URI (Uniform Resource Identifier), όπως είναι το [«tel:»](#) που περιγράφηκε παραπάνω, και έχει προστεθεί στις εφαρμογές χαρτών της Google ώστε να μπορούν να κληθούν για εκτέλεση από άλλες εφαρμογές, με σκοπό την άμεση παροχή της υπηρεσίας πλοήγησης.

Η παράμετρος [«q»](#) μπορεί να δεχτεί εκτός από γεωγραφικές συντεταγμένες και τη διεύθυνση προορισμού σε μορφή κειμένου. Επίσης, υποστηρίζονται οι προαιρετικές παράμετροι [«mode»](#) και [«avoid»](#) για επιλογή της μεθόδου μετακίνησης (με αυτοκίνητο, περπατώντας, κλπ.) και προσπάθεια αποφυγής στοιχείων στη διαδρομή (διόδια, αυτοκινητοδρόμους, κλπ.) αντίστοιχα.



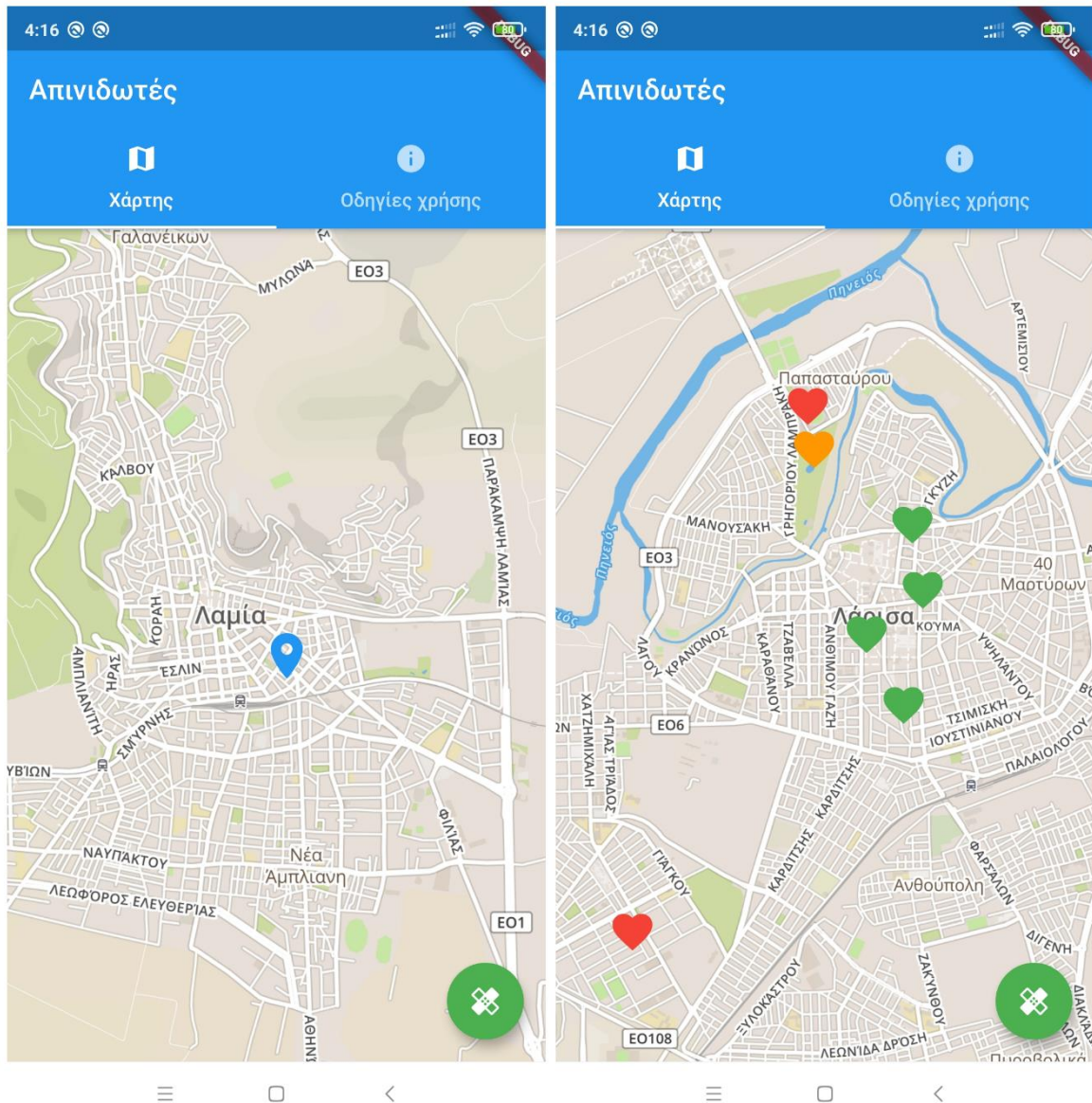
Παράδειγμα: `google.navigation:q=38.9122289,22.426225&mode=w`, όπου η παράμετρος `mode=w` δηλώνει ότι επιθυμούμε να πλοηγηθούμε στις δοσμένες γεωγραφικές συντεταγμένες περπατώντας.

Η συνάρτηση που περιγράφεται χρησιμοποιεί μόνο την παράμετρο `q` με τις ακριβείς γεωγραφικές συντεταγμένες των απινιδωτών και μπορεί να κληθεί από το [μενού επιλογών απινιδωτή](#), από την [οθόνη πληροφοριών απινιδωτή](#), καθώς και από το [κουμπί «έκτακτης ανάγκης»](#).

3.6.4 Συνάρτηση `callIntent`

Η ασύγχρονη συνάρτηση `callIntent` αποτελεί τη δεύτερη συνάρτηση του αρχείου [intent.dart](#) και κάνει χρήση της μοναδικής συνάρτησης `callNumber` του [πακέτου call number](#). Η συνάρτηση αυτή χρησιμοποιείται στην [οθόνη πληροφοριών απινιδωτή](#) και στο [κουμπί «έκτακτης ανάγκης»](#).

3.7 Καρτέλα «Χάρτης»






Εικόνα 3-14 Στιγμιότυπα οθόνης της καρτέλας «Χάρτης»

Η καρτέλα «Χάρτης» είναι η προεπιλεγμένη προς εμφάνιση κατά το άνοιγμα της εφαρμογής. Κατά τη μετάβαση σε αυτή την καρτέλα, πραγματοποιείται κλήση των ασύγχρονων συναρτήσεων «[getLocation](#)» του [πακέτου «location»](#) και «[getAllAeds](#)», οι οποίες είναι υπεύθυνες για τη λήψη της τοποθεσίας του χρήστη και όλων των διαθέσιμων απινιδωτών μέσω του API αντίστοιχα. Μετά την επιτυχή κλήση και των δύο, καλούνται οι συναρτήσεις «[addLocationMarker](#)» και «[addAedMarkers](#)» οι οποίες αναλαμβάνουν την προσθήκη των κατάλληλων εικονιδίων και γεωγραφικών συντεταγμένων στη μεταβλητή-



λίστα «**markers**». Η λίστα αυτή εμπεριέχει όλα τα σημεία στα οποία πρέπει να προστεθούν σημάδια στον χάρτη που πρόκειται να δημιουργηθεί αμέσως μετά με χρήση του [πακέτου «flutter_map»](#). Συγκεκριμένα, η τοποθεσία του χρήστη αναπαρίσταται με το μπλε εικονίδιο που εμφανίζεται στο κέντρο του αριστερού στιγμιότυπου οθόνης. Παράλληλα, οι απινιδωτές αναπαρίστανται με πράσινη, πορτοκαλί ή κόκκινη καρδιά ανάλογα με τη τρέχουσα κατάσταση του καθενός:

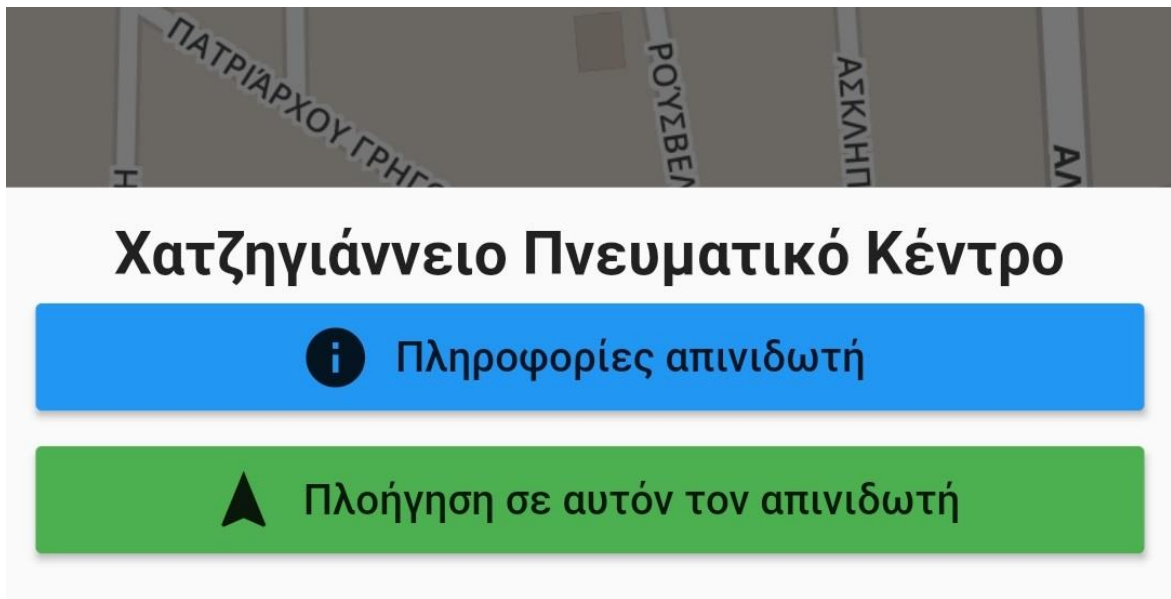
-  (πράσινη καρδιά) : Ο απινιδωτής είναι διαθέσιμος
-  (πορτοκαλί καρδιά) : Ο απινιδωτής είναι σε χρήση
-  (κόκκινη καρδιά) : Ο απινιδωτής είναι εκτός λειτουργίας

Τέλος, όπως προαναφέρθηκε γίνεται χρήση του [πακέτου «flutter_map»](#), για την αναπαράσταση του χάρτη με επίκεντρο τη τρέχουσα τοποθεσία του χρήστη.

Σημείωση: Τη στιγμή που δοκιμαζόταν η εφαρμογή για τη λειτουργικότητά της δεν υπήρχαν γνωστά σημεία εγκατάστασης κοντά στη τοποθεσία του χρήστη. Αυτός είναι ο λόγος που στο αριστερό στιγμιότυπο οθόνης δεν εμφανίζονται απινιδωτές (με το εικονίδιο της καρδιάς) κοντά στο εικονίδιο τοποθεσίας του χρήστη.

3.7.1 Μενού επιλογών απινιδωτή

Το πάτημα του χρήστη επάνω στο εικονίδιο μιας καρδιάς (που αντιστοιχεί σε σημείο ύπαρξης απινιδωτή), προκαλεί στο παρασκήνιο την μεγέθυνση του χάρτη στο σημείο εγκατάστασης του απινιδωτή, ώστε γίνει πιο σαφής στον χρήστη η τοποθεσία στην οποία πρέπει να μεταβεί. Ταυτόχρονα, εμφανίζεται στην οθόνη το όνομα του σημείου εγκατάστασης και δύο (2) επιλογές.

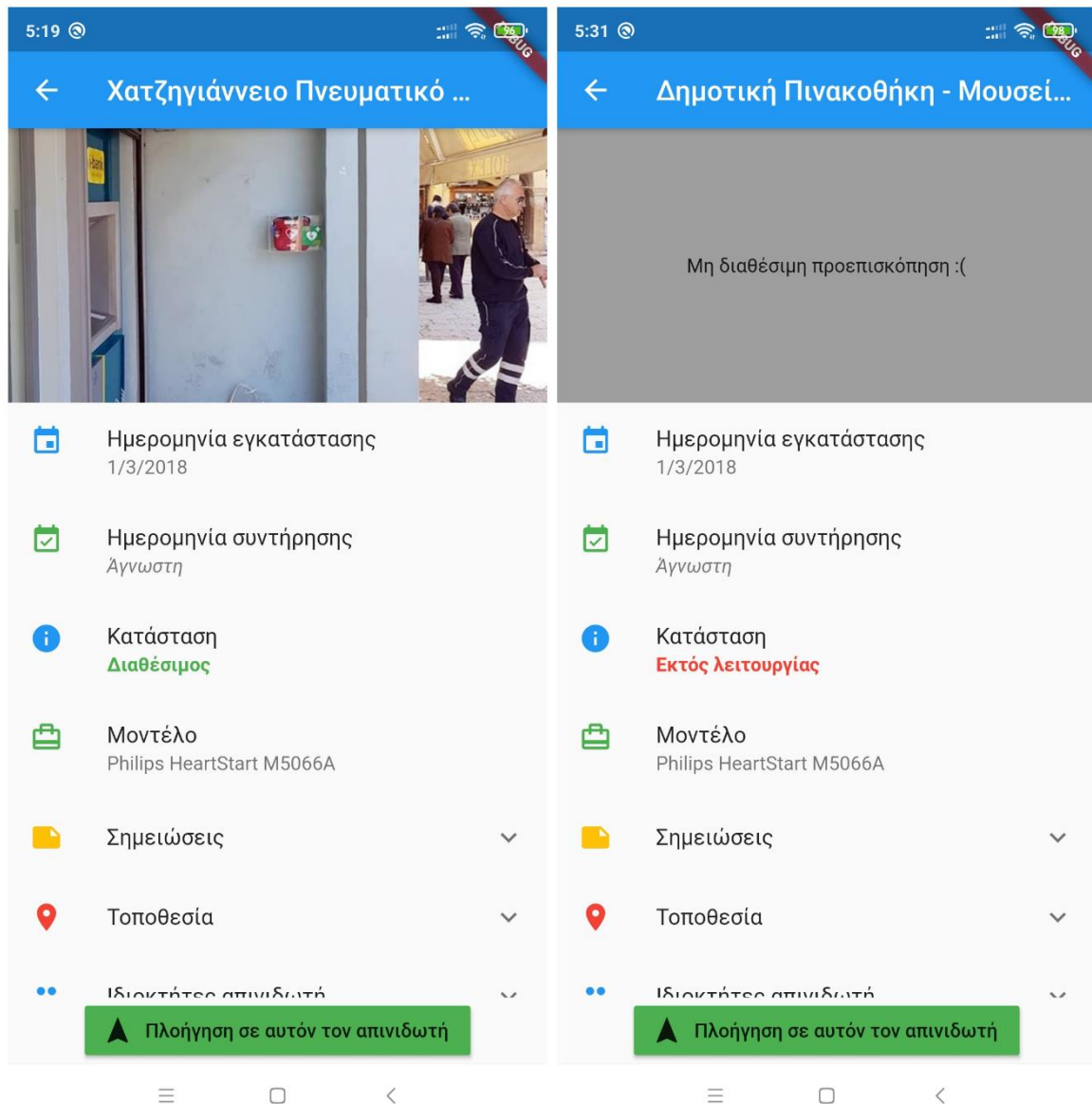


Εικόνα 3-15 Στιγμιότυπο οθόνης του μενού επιλογών απινιδωτή

Το πρώτο μπλε κουμπί «Πληροφορίες απινιδωτή» πλοηγεί τον χρήστη στην [οθόνη πληροφοριών](#) του επιλεγμένου απινιδωτή στην οποία μπορεί να βρει πολλές χρήσιμες πληροφορίες σχετικά με αυτόν.

Το δεύτερο πράσινο κουμπί «Πλοήγηση σε αυτόν τον απινιδωτή» πραγματοποιεί κλήση της [συνάρτησης <navigationIntent>](#) η οποία, όπως προαναφέρθηκε, «ζητάει» από το λειτουργικό σύστημα την εκτέλεση οποιασδήποτε εγκατεστημένης εφαρμογής χαρτών της Google σε λειτουργία πλοήγησης προς τις γεωγραφικές συντεταγμένες του συγκεκριμένου απινιδωτή.

3.7.2 Οθόνη πληροφοριών απινιδωτή



Εικόνα 3-16 Στιγμιότυπα οθόνης της οθόνης πληροφοριών απινιδωτή

Στην οθόνη αυτή ο χρήστης μπορεί να βρει τις εξής πληροφορίες ενός απινιδωτή:

- **Εικόνα προεπισκόπησης:** Εμφανίζεται μια φωτογραφία από το σημείο εγκατάστασης του απινιδωτή που πιθανώς έχει μεταφορτώσει ο διαχειριστής του στον διακομιστή του [Strapi](#). Σε περίπτωση που δεν υπάρχει διαθέσιμη φωτογραφία, εμφανίζεται το μήνυμα «Μη διαθέσιμη προεπισκόπηση».
- **Ημερομηνία εγκατάστασης:** Εμφανίζει την ημερομηνία εγκατάστασης του απινιδωτή στη συγκεκριμένη τοποθεσία. Εάν δεν έχει οριστεί από τον διαχειριστή του, εμφανίζεται το κείμενο «Άγνωστη».



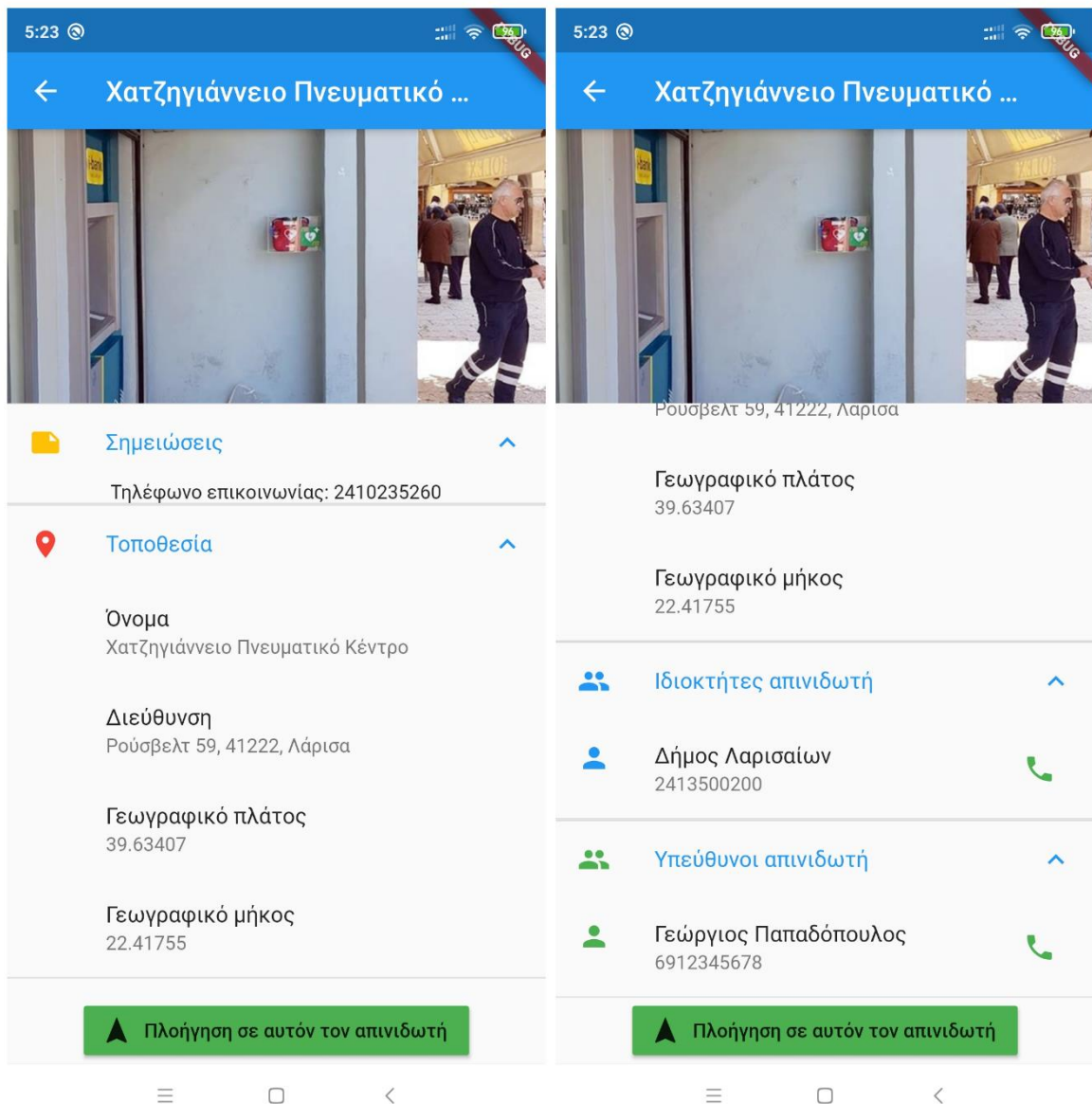
Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού, πλοήγησης προς και χρήσης απινιδωτών

- **Ημερομηνία συντήρησης:** Εμφανίζει τη τελευταία ημερομηνία συντήρησης του απινιδωτή. Εάν δεν έχει οριστεί ή δεν έχει ενημερωθεί από τον διαχειριστή του, εμφανίζεται το κείμενο «Άγνωστη».
- **Κατάσταση:** Εμφανίζεται το κείμενο «Διαθέσιμος», «Σε χρήση» ή «Εκτός λειτουργίας» με συγκεκριμένο χρώμα ανάλογα με την τρέχουσα κατάσταση του απινιδωτή.
- **Μοντέλο:** Η μάρκα και το μοντέλο του επιλεγμένου απινιδωτή.
- **Σημειώσεις:** Εμφανίζει ένα σύντομο κείμενο στο οποίο ο διαχειριστής του απινιδωτή μπορεί να αναφέρει οτιδήποτε επιθυμεί και θεωρεί χρήσιμο, όπως π.χ. ένα τηλέφωνο επικοινωνίας ή κάποια ιδιαιτερότητα του απινιδωτή που πρέπει να προσέξει ο χρήστης.
- **Τοποθεσία:** Εμφανίζει το όνομα του σημείου εγκατάστασης, τη διεύθυνση, τον ταχυδρομικό κώδικα και το όνομα της πόλης στην οποία βρίσκεται ο απινιδωτής. Επίσης αναφέρονται το γεωγραφικό πλάτος και το γεωγραφικό μήκος του σημείου εγκατάστασης χάρη στα οποία πραγματοποιείται με ακρίβεια η πλοήγηση του χρήστη στην τοποθεσία που πρέπει να μεταβεί.
- **Ιδιοκτήτες απινιδωτή:** Εμφανίζει μια λίστα με όλους τους ιδιοκτήτες του συγκεκριμένου απινιδωτή. Η λίστα αποτελείται από ονόματα και τηλεφωνικούς αριθμούς. Για λόγους ευκολίας, δεξιά από κάθε ιδιοκτήτη υπάρχει ένα πράσινο εικονίδιο τηλεφώνου, πατώντας το οποίο καλείται η [συνάρτηση «callIntent»](#), η οποία πραγματοποιεί άμεση κλήση στον τηλεφωνικό αριθμό του αντίστοιχου ιδιοκτήτη που εμφανίζεται κάτω από το όνομά του.
- **Υπεύθυνοι απινιδωτή:** Εμφανίζει μια λίστα με όλους τους υπεύθυνους του συγκεκριμένου απινιδωτή. Η λίστα αυτή, όπως και η λίστα ιδιοκτητών, αποτελείται από ονόματα και τηλεφωνικούς αριθμούς. Και πάλι για λόγους ευκολίας, υπάρχει ένα πράσινο εικονίδιο τηλεφώνου το οποίο επιτελεί τη λειτουργία που περιγράφηκε πριν.

Τέλος, στο κάτω μέρος της οθόνης εμφανίζεται ένα πράσινο κουμπί «**Πλοήγηση σε αυτόν τον απινιδωτή**» που εκτελεί την ίδια λειτουργία με το ομώνυμο κουμπί του [μενού επιλογών απινιδωτή](#).

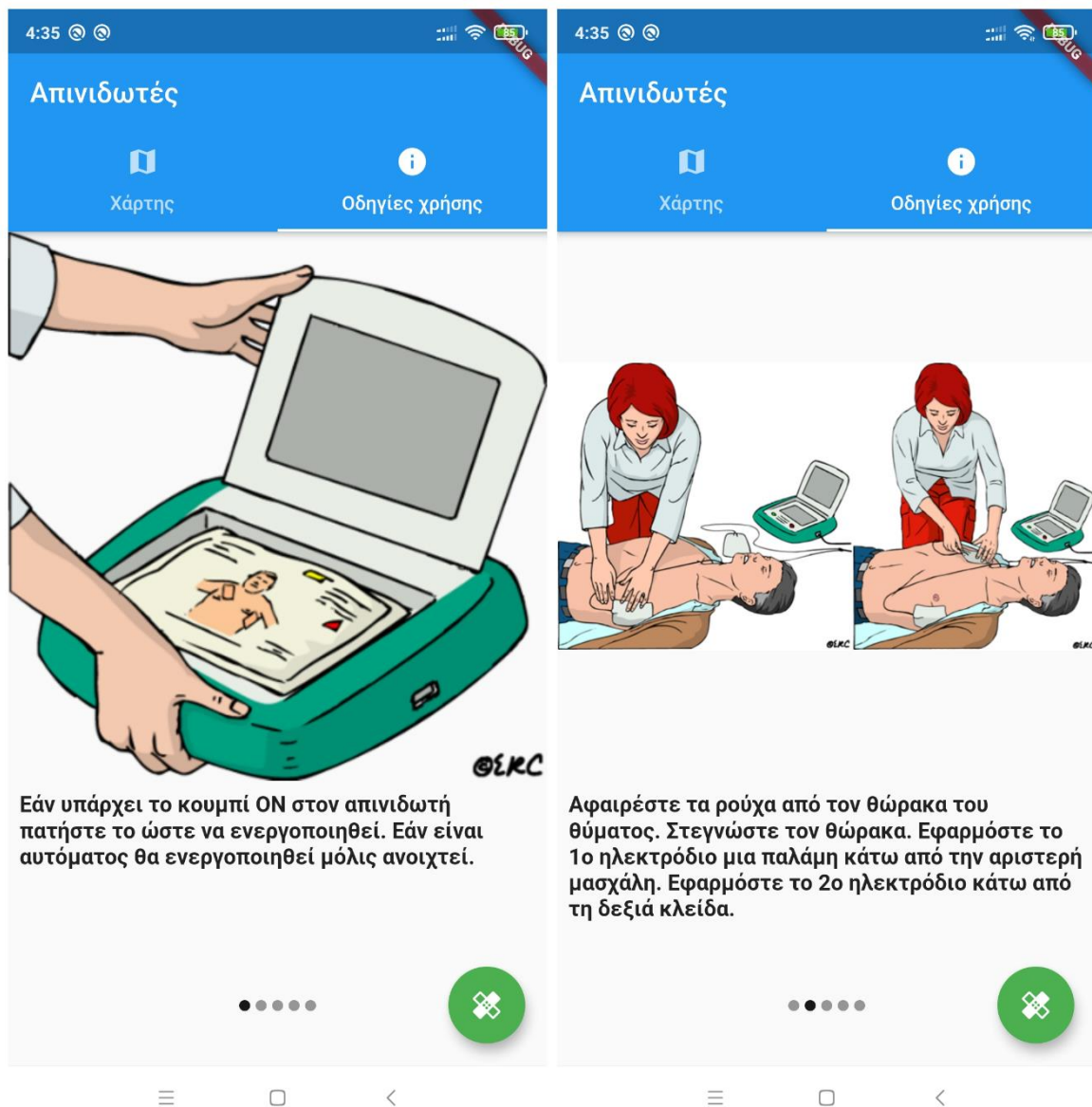


Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού, πλοήγησης προς και χρήσης απινιδωτών



Εικόνα 3-17 Στιγμιότυπα οθόνης της οθόνης πληροφοριών απινιδωτή (2)

3.8 Καρτέλα «Οδηγίες χρήσης»



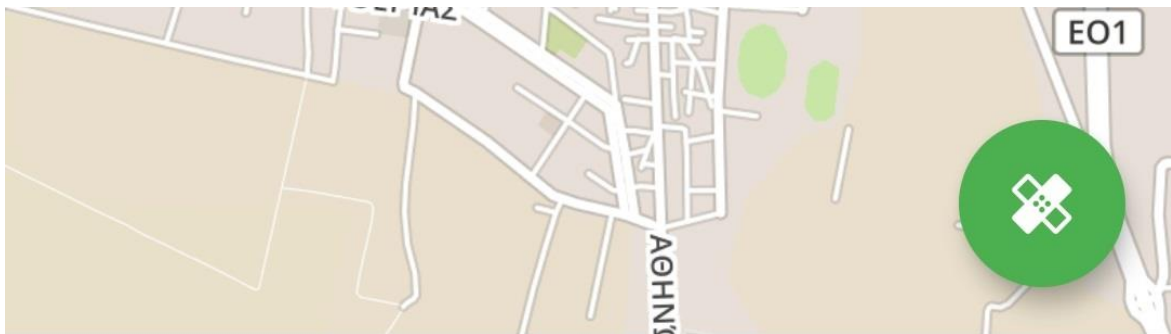
Εικόνα 3-18 Στιγμιότυπα οθόνης της καρτέλας «Οδηγίες χρήσης»

Για την ενσωμάτωση της καρτέλας «Οδηγίες χρήσης» στην εφαρμογή χρησιμοποιήθηκε το [πακέτο «carousel_slider»](#) που περιγράφηκε παραπάνω. Η καρτέλα αυτή παρουσιάζει τη διαδικασία απινίδωσης σε πέντε (5) απλά βήματα. Κάθε βήμα αποτελείται από μια εικόνα αναπαράστασης και από μια σύντομη περιγραφή εκτέλεσης. Ο χρήστης μπορεί να μεταβεί από ένα βήμα στο προηγούμενο ή στο επόμενο του σύροντας το δάχτυλό του προς τα δεξιά ή αριστερά αντίστοιχα. Η καρτέλα αυτή προστέθηκε με σκοπό την υπενθύμιση των βημάτων απινίδωσης στους ήδη εκπαιδευμένους πάνω στις πρώτες βοήθειες οποιαδήποτε



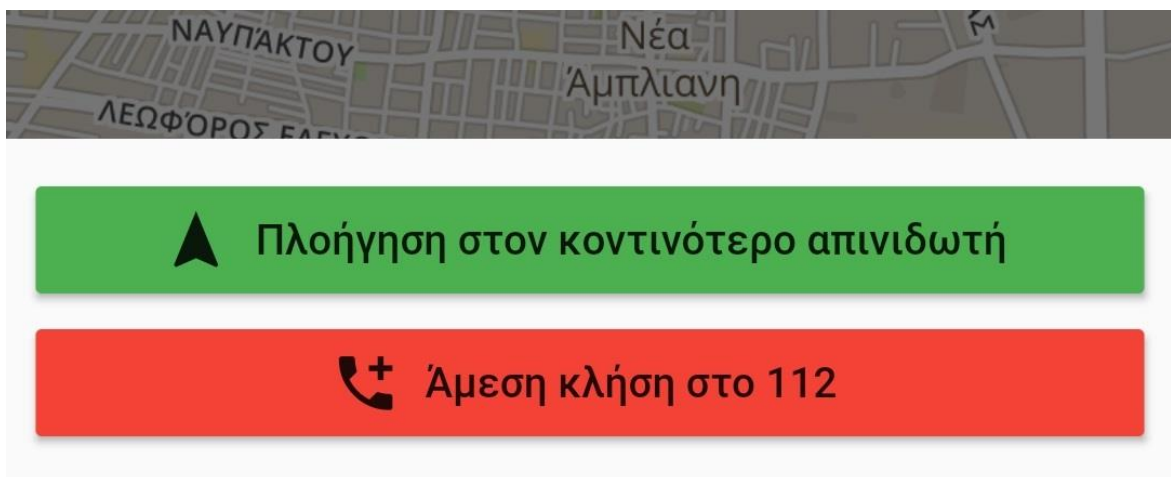
στιγμή το επιθυμούν. Ωστόσο, λήφθηκε επίσης υπόψη η χειρότερη περίπτωση εκτέλεσης της διαδικασίας από κάποιον ανεκπαιδευτο. Σε μια τέτοια περίπτωση, είναι προφανώς προτιμότερο να ξεκινήσει ο ίδιος τη διαδικασία, διαβάζοντας από την εφαρμογή και εκτελώντας βήμα-βήμα τη διαδικασία, καθώς η αναμονή του ΕΚΑΒ ή οποιουδήποτε άλλου εκπαιδευμένου ατόμου μπορεί να στοιχίσει στο θύμα τη ζωή του.

3.9 Κουμπί «έκτακτης ανάγκης»



Εικόνα 3-19 Στιγμιότυπο οθόνης του κουμπιού «έκτακτης ανάγκης»

Το κουμπί «έκτακτης ανάγκης» προσφέρει στον χρήστη δύο (2) επιλογές.



Εικόνα 3-20 Στιγμιότυπο οθόνης των επιλογών του κουμπιού «έκτακτης ανάγκης»

Το πρώτο πράσινο κουμπί «Πλοήγηση στον κοντινότερο απινιδωτή» πραγματοποιεί αρχικά κλήση της ασύγχρονης [συνάρτησης «getNearestAed»](#). Μόλις λάβει από αυτήν αποτέλεσμα (δηλαδή το αντικείμενο της [κλάσης «Aed»](#) που απέχει την μικρότερη απόσταση σε σχέση με την τρέχουσα τοποθεσία του χρήστη), εκτελεί κλήση της



Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού, πλοήγησης προς και χρήσης απινιδωτών

[συνάρτησης «navigationIntent»](#). Η τελευταία με τη σειρά της «ζητάει» από το λειτουργικό σύστημα την εκτέλεση οποιασδήποτε εγκατεστημένης εφαρμογής χαρτών της Google, η οποία θα ανοίξει κατευθείαν σε λειτουργία πλοήγησης προς τις γεωγραφικές συντεταγμένες του απινιδωτή που επέστρεψε η [συνάρτηση «getNearestAed»](#).

Το δεύτερο κόκκινο κουμπί «**Άμεση κλήση στο 112**» πραγματοποιεί κλήση της [συνάρτησης «callIntent»](#) με όρισμα τον [ευρωπαϊκό αριθμό κλήσης έκτακτης ανάγκης \(112\)](#) σε μορφή συμβολοσειράς. Αυτή με τη σειρά της «ζητάει» από το λειτουργικό σύστημα να εκτελέσει την υπεύθυνη εφαρμογή για τη διαχείριση τηλεφωνικών κλήσεων, ώστε να πραγματοποιήσει άμεση κλήση στον αριθμό 112 χωρίς την ανάγκη διαμεσολάβησης του χρήστη. Αυτή η δυνατότητα προστέθηκε για τον λόγο ότι ακόμη και τα ελάχιστα δευτερόλεπτα που χρειάζονται για μια απλή τηλεφωνική μπορούν να αποβούν μοιραία για το θύμα.



Ιστογραφία

- [1] Καρδιακή ανακοπή, Αυτόματοι Απινιδωτές και ο Δήμος Μεγαλόπολης. (24 Μαΐου 2017). Ανακτήθηκε από <https://www.arcadiaportal.gr/news/kardiaki-anakopi-aytomatoi-apinidotes-kai-o-dimos-megalopolis>
- [2] Τι είναι και πώς λειτουργεί ο αυτόματος απινιδωτής. Ανακτήθηκε από https://orthomedicare.com.gr/blog?journal_blog_post_id=39
- [3] 112 | Digital Single Market. Ανακτήθηκε από <https://ec.europa.eu/digital-single-market/en/112>
- [4] 112 (emergency telephone number). (28 Ιανουαρίου 2020). Ανακτήθηκε από [https://en.wikipedia.org/wiki/112_\(emergency_telephone_number\)](https://en.wikipedia.org/wiki/112_(emergency_telephone_number))
- [5] Υπηρεσία Επικοινωνιών Εκτάκτου Ανάγκης 112. Ανακτήθηκε από <https://www.civilprotection.gr/el/ypiresia-epikoinonion-ektaktou-anagkis-112>
- [6] Representational state transfer. (28 Ιανουαρίου 2020). Ανακτήθηκε από https://en.wikipedia.org/wiki/Representational_state_transfer
- [7] Avraham, S. B. (5 Σεπτεμβρίου 2017). What is REST - A Simple Explanation for Beginners, Part 1: Introduction. Retrieved from <https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-1-introduction-b4a072f8740f>
- [8] Strapi - Open source Node.js Headless CMS . Ανακτήθηκε από <https://strapi.io/>
- [9] Strapi open source Node.js Headless CMS Frequently Asked Questions. Ανακτήθηκε από <https://strapi.io/faq>
- [10] GitHub - strapi/strapi: Open source Node.js Headless CMS to easily build customisable APIs. Ανακτήθηκε από <https://github.com/strapi/strapi>
- [11] Strapi Documentation. Ανακτήθηκε από <https://strapi.io/documentation/3.0.0-beta.x/getting-started/introduction.html>
- [12] Flutter - Beautiful native apps in record time. Ανακτήθηκε από <https://flutter.dev/>
- [13] Flutter (software). (27 Ιανουαρίου 2020). Ανακτήθηκε από [https://en.wikipedia.org/wiki/Flutter_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software))



- [14] Thomas, G. (29 Ιανουαρίου 2020). What is Flutter and Why You Should Learn It in 2020. Ανακτήθηκε από <https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>
- [15] Mapbox. Ανακτήθηκε από <https://www.mapbox.com/>
- [16] Mapbox. (26 Νοεμβρίου 2019). Ανακτήθηκε από <https://en.wikipedia.org/wiki/Mapbox>
- [17] OpenStreetMap. (n.d.). Ανακτήθηκε από <https://www.openstreetmap.org/>
- [18] Navigation service | API | Mapbox. Ανακτήθηκε από <https://docs.mapbox.com/api/navigation/#directions>
- [19] Common Intents | Android Developers. Ανακτήθηκε από <https://developer.android.com/guide/components/intents-common>
- [20] GitHub - Lyokone/flutterlocation: A Flutter plugin to easily handle realtime location in iOS and Android. Provides settings for optimizing performance or battery. Ανακτήθηκε από <https://github.com/Lyokone/flutterlocation>
- [21] GitHub - johnpryan/flutter_map: A Flutter map widget inspired by Leaflet. Ανακτήθηκε από https://github.com/johnpryan/flutter_map
- [22] Leaflet - a JavaScript library for interactive maps. Ανακτήθηκε από <https://leafletjs.com/>
- [23] GitHub - serenader2014/flutter_carousel_slider: A flutter carousel widget, support infinite scroll, and custom child widget. Ανακτήθηκε από https://github.com/serenader2014/flutter_carousel_slider
- [24] plugins/packages/android_intent at master · flutter/plugins · GitHub. Ανακτήθηκε από https://github.com/flutter/plugins/tree/master/packages/android_intent
- [25] GitHub - sreesdas/flutter_call_number_plugin: A Flutter plugin to initiate calls directly from the app itself, without opening a dialer. Ανακτήθηκε από https://github.com/sreesdas/flutter_call_number_plugin



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Παράρτημα Α: Κώδικας εφαρμογής



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Φάκελος «models»

Αρχείο «aed.dart»



```
import 'package:flutter/material.dart';
import 'package:latlong/latlong.dart';
import 'package:intl/intl.dart';

import './aed_location.dart';
import './aed_manager.dart';
import './aed_model.dart';
import './aed_owner.dart';

class Aed {
  final int id;
  final String installationDate;
  final String lastOperationalCheckDate;
  final String notes;
  final String status;
  final AedModel model;
  final AedLocation location;
  final List<AedOwner> owners;
  final List<AedManager> managers;

  String get statusText {
    switch (status) {
      case 'available':
        return 'Διαθέσιμος';
      case 'in_use':
        return 'Σε χρήση';
      case 'out_of_order':
        return 'Εκτός λειτουργίας';
      default:
        return 'Άγνωστη';
    }
  }

  Color get statusColor {
    switch (status) {
      case 'available':
        return Colors.green;
      case 'in_use':
        return Colors.orange;
      case 'out_of_order':
        return Colors.red;
      default:
        return Colors.grey;
    }
  }

  Aed({
    this.id,
    this.installationDate,
    this.lastOperationalCheckDate,
    this.notes,
    this.status,
    this.model,
    this.location,
    this.managers,
    this.owners,
  });

  factory Aed.fromJson(Map<String, dynamic> json) {
    return Aed(
      id: json['id'],

```




```
installationDate: json['installation_date'] != null
    ? DateFormat('d/M/y').format(
        DateTime.parse(
            json['installation_date'],
        ),
    )
    : '',
lastOperationalCheckDate: json['last_operational_check_date']
!= null
    ? DateFormat('d/M/y').format(
        DateTime.parse(
            json['last_operational_check_date'],
        ),
    )
    : '',
notes: json['notes'] ?? '',
status: json['status'],
model: AedModel(
    name: json['model']['name'],
),
location: AedLocation(
    name: json['location']['name'],
    address: json['location']['address'],
    postalCode: json['location']['postal_code'],
    municipalityName: json['location']['municipality']['name'],
    latLng: LatLng(
        json['location']['latitude'],
        json['location']['longitude'],
    ),
    image: json['location']['image'],
),
owners: json['owners']
    .map<AedOwner>(
        (owner) => AedOwner(
            name: owner['name'],
            phoneNumber: owner['phone_number'],
        ),
    )
    .toList(),
managers: json['managers']
    .map<AedManager>(
        (manager) => AedManager(
            name: manager['name'],
            phoneNumber: manager['phone_number'],
        ),
    )
    .toList(),
);
}
}
```



Αρχείο «aed_location.dart»

```
import 'package:latlong/latlong.dart';

class AedLocation {
  final String name;
  final String address;
  final String postalCode;
  final String municipalityName;
  final LatLng latLng;
  final Map<String, Object> image;

  AedLocation({
    this.name,
    this.address,
    this.postalCode,
    this.municipalityName,
    this.latLng,
    this.image,
  });
}
```

Αρχείο «aed_manager.dart»

```
class AedManager {
  final String name;
  final String phoneNumber;

  AedManager({
    this.name,
    this.phoneNumber,
  });
}
```

Αρχείο «aed_model.dart»

```
class AedModel {
  final String name;

  AedModel({this.name});
}
```

Αρχείο «aed_owner.dart»

```
class AedOwner {
  final String name;
  final String phoneNumber;

  AedOwner({
    this.name,
    this.phoneNumber,
  });
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Φάκελος «screens»

Αρχείο «aed_details_screen.dart»



```
import 'package:flutter/material.dart';

import '../models/aed.dart';
import '../intent.dart';

class AedDetailsScreen extends StatelessWidget {
  static const route = '/aed-details';

  @override
  Widget build(BuildContext context) {
    final selectedAed = ModalRoute.of(context).settings.arguments as
Aed;

    Widget getDate(String date) {
      return date.isNotEmpty
        ? Text(date)
        : Text(
            'Άγνωστη',
            style: TextStyle(fontStyle: FontStyle.italic),
          );
    }

    Widget getNotes(String notes) {
      return notes.isNotEmpty
        ? Text(notes)
        : Text(
            'Δεν υπάρχουν σημειώσεις',
            style: TextStyle(fontStyle: FontStyle.italic),
          );
    }

    return Scaffold(
      appBar: AppBar(
        title: Text('${selectedAed.location.name}'),
      ),
      body: Column(
        children: <Widget>[
          Container(
            height: 200,
            width: double.infinity,
            child: selectedAed.location.image != null
              ? Image.network(
                  'http://{SERVER_IP}:{SERVER_PORT}/${selectedAed.location.image['url']}
                )',
                fit: BoxFit.cover,
              )
            : Center(
                child: Text('Μη διαθέσιμη προεπισκόπηση :('),
              ),
            color: Colors.grey,
          ),
          Expanded(
            child: ListView(
              children: <Widget>[
                ListTile(
                  leading: Icon(
                    Icons.event,
                    color: Colors.blue,
                  ),
                  title: Text('Ημερομηνία εγκατάστασης'),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

```
        subtitle: getDate(selectedAed.installationDate),
    ),
    ListTile(
      leading: Icon(
        Icons.event_available,
        color: Colors.green,
      ),
      title: Text('Ημερομηνία συντήρησης'),
      subtitle:
        getDate(selectedAed.lastOperationalCheckDate),
    ),
    ListTile(
      leading: Icon(
        Icons.info,
        color: Colors.blue,
      ),
      title: Text('Κατάσταση'),
      subtitle: Text(
        selectedAed.statusText,
        style: TextStyle(
          color: selectedAed.statusColor,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
    ListTile(
      leading: Icon(
        Icons.card_travel,
        color: Colors.green,
      ),
      title: Text('Μοντέλο'),
      subtitle: Text(selectedAed.model.name),
    ),
    ExpansionTile(
      leading: Icon(
        Icons.note,
        color: Colors.amber,
      ),
      title: Text('Σημειώσεις'),
      children: <Widget>[
        Padding(
          padding: EdgeInsets.symmetric(vertical: 10.0),
          child: getNotes(selectedAed.notes),
        ),
      ],
    ),
    ExpansionTile(
      leading: Icon(
        Icons.location_on,
        color: Colors.red,
      ),
      title: Text('Τοποθεσία'),
      children: <Widget>[
        ListTile(
          leading: Padding(padding:
EdgeInsets.all(16.0)),
          title: Text('Όνομα'),
          subtitle: Text(selectedAed.location.name),
        ),
        ListTile(
          leading: Padding(padding:
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

```
EdgeInsets.all(16.0)),
    title: Text('Διεύθυνση'),
    subtitle: Text(
        '${selectedAed.location.address}',
    ),
    ${selectedAed.location.postalCode},
    ${selectedAed.location.municipalityName}'),
),
ListTile(
    leading: Padding(padding:
EdgeInsets.all(16.0)),
    title: Text('Γεωγραφικό πλάτος'),
    subtitle: Text(
        '${selectedAed.location.latLng.latitude}',
    ),
),
),
ListTile(
    leading: Padding(padding:
EdgeInsets.all(16.0)),
    title: Text('Γεωγραφικό μήκος'),
    subtitle: Text(
        '${selectedAed.location.latLng.longitude}',
    ),
),
),
],
),
ExpansionTile(
    leading: Icon(
        Icons.people,
        color: Colors.blue,
    ),
    title: Text('Ιδιοκτήτες απινιδωτή'),
    children: selectedAed.owners
        .map<Widget>(
            (owner) => ListTile(
                leading: Icon(
                    Icons.person,
                    color: Colors.blue,
                ),
                title: Text('${owner.name}'),
                subtitle: Text('${owner.phoneNumber}'),
                trailing: IconButton(
                    icon: Icon(
                        Icons.call,
                        color: Colors.green,
                    ),
                    onPressed: () => callIntent('1314'),
                ),
            ),
        )
        .toList(),
),
ExpansionTile(
    leading: Icon(
        Icons.people,
        color: Colors.green,
    ),
    title: Text('Υπεύθυνοι απινιδωτή'),
    children: selectedAed.managers
        .map<Widget>(
            (manager) => ListTile(
                leading: Icon(
```



Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού, πλοήγησης προς και χρήσης απινιδωτών

```
        Icons.person,
        color: Colors.green,
    ),
    title: Text('${manager.name}'),
    subtitle: Text('${manager.phoneNumber}'),
    trailing: IconButton(
        icon: Icon(
            Icons.call,
            color: Colors.green,
        ),
        onPressed: () => callIntent('1314'),
    ),
),
),
.toList(),
),
],
),
),
RaisedButton.icon(
    color: Colors.green,
    icon: Icon(Icons.navigation),
    label: Text('Πλοήγηση σε αυτόν τον απινιδωτή'),
    onPressed: () =>
navigationIntent(selectedAed.location.latLng),
),
],
),
);
}
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Αρχείο «aed_instructions_screen.dart»



```
import 'package:flutter/material.dart';
import 'package:carousel_slider/carousel_slider.dart';

List<T> map<T>(List list, Function handler) {
  List<T> result = [];
  for (var i = 0; i < list.length; i++) {
    result.add(handler(i, list[i]));
  }

  return result;
}

class AedInstructionsScreen extends StatefulWidget {
  static const route = '/aed-instructions';

  @override
  _AedInstructionsScreenState createState() =>
    _AedInstructionsScreenState();
}

class _AedInstructionsScreenState extends
  State<AedInstructionsScreen> {
  static const List<Map<String, Object>> instructions = [
    {
      "step": 1,
      "image": "https://i.ibb.co/slkt4tt/step1.png",
      "text":
        "Εάν υπάρχει το κουμπί ON στον απινιδωτή πατήστε το ώστε να
        ενεργοποιηθεί. Εάν είναι αυτόματος θα ενεργοποιηθεί μόλις ανοιχτεί.",
    },
    {
      "step": 2,
      "image": "https://i.ibb.co/Wcq1pBr/step2.png",
      "text":
        "Αφαιρέστε τα ρούχα από τον θώρακα του θύματος. Στεγνώστε
        τον θώρακα. Εφαρμόστε το 1ο ηλεκτρόδιο μια παλάμη κάτω από την
        αριστερή μασχάλη. Εφαρμόστε το 2ο ηλεκτρόδιο κάτω από τη δεξιά
        κλείδα.",
    },
    {
      "step": 3,
      "image": "https://i.ibb.co/PgDMbhf/step3.png",
      "text":
        "Όταν τα ηλεκτρόδια τοποθετηθούν, κρατήστε απόσταση από το
        θύμα όσο ο ΑΕΑ αναλύει τον καρδιακό ρυθμό. Βεβαιωθείτε ότι κανείς δεν
        αγγίζει το θύμα κατά τη διάρκεια της ανάλυσης.",
    },
    {
      "step": 4,
      "image": "https://i.ibb.co/0918mHd/step4.png",
      "text":
        "Εάν ο ΑΕΑ υποδεικνύει απινίδωση βεβαιωθείτε ότι κανείς δεν
        ακουμπά το θύμα. Φωνάξτε δυνατά «Απομακρυνθείτε! Πραγματοποιείται
        απινίδωση.». Πατήστε το κουμπί απινίδωσης.",
    },
    {
      "step": 5,
      "image": "https://i.ibb.co/YT5zCtc/step5.png",
      "text":
        "Εάν δεν υπάρχει αντίδραση από το θύμα, συνεχίστε ΚΑΡΠΑ
        (30:2). Αν κατά τη διάρκεια παροχής ΚΑΡΠΑ παρατηρήσετε αντίδραση,
```



```
σταματήστε την ΚΑΡΠΑ και ελέγξτε για φυσιολογική αναπνοή.",
    },
  ],
  int _current = 0;

  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        CarouselSlider(
          items: instructions.map(
            (step) {
              return Column(
                children: <Widget>[
                  Container(
                    child: Image.network(
                      step['image'],
                      width: 400,
                      height: 400,
                    ),
                  ),
                  Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: Text(
                      step['text'],
                      style: TextStyle(
                        fontSize: 16.0,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                ],
              );
            },
          ).toList(),
          height: 550,
          viewportFraction: 1.0,
          enableInfiniteScroll: false,
          autoPlay: false,
          enlargeCenterPage: false,
          onPageChanged: (index) {
            setState(() => _current = index);
          },
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: map<Widget>( instructions, (index, url) {
            return Container(
              width: 8.0,
              height: 8.0,
              margin: EdgeInsets.symmetric(vertical: 10.0,
horizontal: 2.0),
              decoration: BoxDecoration(
                shape: BoxShape.circle,
                color: _current == index
                  ? Color.fromRGBO(0, 0, 0, 0.9)
                  : Color.fromRGBO(0, 0, 0, 0.4)),
            );
          },
        ),
      )
    );
  }
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

}
)
}
}



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Αρχείο «aed_map_screen.dart»



```
import 'package:flutter/material.dart';
import 'package:latlong/latlong.dart';
import 'package:location/location.dart';
import 'package:flutter_map/flutter_map.dart';

import '../models/aed.dart';
import '../screens/aed_details_screen.dart';
import '../future.dart';
import '../intent.dart';

const accessToken = {MAPBOX_ACCESS_TOKEN};
LocationData userLocation;
List<Aed> aedList;

class AedMapScreen extends StatefulWidget {
  static const route = '/aed-map';

  @override
  _AedMapScreenState createState() => _AedMapScreenState();
}

class _AedMapScreenState extends State<AedMapScreen> {
  var location = Location();
  var mapController = MapController();
  List<Marker> markers = [];

  void _addLocationMarker() {
    markers.add(
      Marker(
        width: 40,
        height: 40,
        point: LatLng(
          userLocation.latitude,
          userLocation.longitude,
        ),
        builder: (ctx) => Container(
          child: IconButton(
            icon: Icon(Icons.location_on),
            color: Colors.blue,
            iconSize: 40.0,
            onPressed: () => {},
          ),
        ),
      ),
    );
  }

  void _addAedMarkers() {
    aedList.forEach((aed) {
      markers.add(
        Marker(
          width: 40,
          height: 40,
          point: aed.location.latLng,
          builder: (ctx) => Container(
            child: IconButton(
              icon: Icon(Icons.favorite),
              color: aed.statusColor,
              iconSize: 35.0,
              onPressed: () {
                mapController.move(aed.location.latLng, 17.0);
              }
            ),
          ),
        ),
      );
    });
  }
}
```



```
        _showAedOptions(aed);
    },
    ),
),
);
});
}

void _showAedOptions(Aed aed) {
    showModalBottomSheet(
        context: context,
        builder: (builder) {
            return Padding(
                padding: const EdgeInsets.all(10.0),
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.min,
                    crossAxisAlignment: CrossAxisAlignment.stretch,
                    children: <Widget>[
                        Center(
                            child: Text(
                                aed.location.name,
                                style: TextStyle(
                                    fontSize: 20,
                                    fontWeight: FontWeight.bold,
                                ),
                            ),
                        ),
                        RaisedButton.icon(
                            color: Colors.blue,
                            icon: Icon(Icons.info),
                            label: Text('Πληροφορίες απινιδωτή'),
                            onPressed: () => Navigator.pushNamed(
                                context, AedDetailsScreen.route,
                                arguments: aed),
                        ),
                        RaisedButton.icon(
                            color: Colors.green,
                            icon: Icon(Icons.navigation),
                            label: Text('Πλοήγηση σε αυτόν τον απινιδωτή'),
                            onPressed: () =>
navigationIntent(aed.location.latLng),
                        ),
                    ],
                ),
            );
        },
    );
}

@override
Widget build(BuildContext context) {
    return FutureBuilder(
        future: Future.wait([
            location.getLocation(),
            getAllAeds(),
        ]),
        builder: (context, snapshot) {
            if (snapshot.hasData) {
                userLocation = snapshot.data[0];
                aedList = snapshot.data[1];
            }
        }
    );
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

```
_addLocationMarker();
_addAedMarkers();

return FlutterMap(
  mapController: mapController,
  options: MapOptions(
    center: LatLng(
      userLocation.latitude,
      userLocation.longitude,
    ),
    zoom: 14.0,
  ),
  layers: [
    TileLayerOptions(
      urlTemplate:
"https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}@2x.png?access_token
={accessToken}",
      additionalOptions: {
        'accessToken': accessToken,
        'id': 'mapbox.streets',
      },
    ),
    MarkerLayerOptions(markers: markers),
  ],
);
} else if (snapshot.hasError) {
  return Text("${snapshot.error}");
}

return Padding(
  padding: const EdgeInsets.symmetric(horizontal: 10.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.center,
    children: <Widget>[
      const Text('Γίνεται λήψη της τοποθεσίας σας και των
απινιδωτών.'),
      const Text('Παρακαλώ περιμένετε...'),
      const Padding(
        padding: const EdgeInsets.all(20.0),
        child: CircularProgressIndicator(),
      ),
    ],
  ),
);
},
);
}
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Αρχείο «tabs_screen.dart»



```
import 'package:flutter/material.dart';
import 'package:latlong/latlong.dart';

import './aed_instructions_screen.dart';
import './aed_map_screen.dart';
import '../future.dart';
import '../intent.dart';

class TabsScreen extends StatefulWidget {
  @override
  _TabsScreenState createState() => _TabsScreenState();
}

class _TabsScreenState extends State<TabsScreen> {

  void showEmergencyOptions() {
    showModalBottomSheet(
      context: context,
      builder: (builder) {
        return Padding(
          padding: const EdgeInsets.all(10.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: <Widget>[
              RaisedButton.icon(
                color: Colors.green,
                icon: Icon(Icons.navigation),
                label: Text('Πλοήγηση στον κοντινότερο απινιδωτή'),
                onPressed: () {
                  LatLng(userLocation.latitude, userLocation.longitude), aedList)
                    .then((aed) =>
                      navigationIntent(aed.location.latLng));
                },
              RaisedButton.icon(
                color: Colors.red,
                icon: Icon(Icons.add_call),
                label: Text('Άμεση κλήση στο 112'),
                onPressed: () => callIntent('112'),
              ),
            ],
          ),
        );
      },
    );
  }

  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 2,
      child: Scaffold(
        appBar: AppBar(
          title: Text('Απινιδωτές'),
          bottom: TabBar(
            tabs: <Widget>[
              Tab(
                icon: Icon(Icons.map),
                text: 'Χάρτης',
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

```
        Tab(  
            icon: Icon(Icons.info),  
            text: 'Οδηγίες χρήσης',  
        ),  
    ],  
),  
),  
body: TabBarView(  
    physics: NeverScrollableScrollPhysics(),  
    children: <Widget>[  
        AedMapScreen(),  
        AedInstructionsScreen(),  
    ],  
),  
floatingActionButton: FloatingActionButton(  
    onPressed: showEmergencyOptions,  
    child: Icon(Icons.healing),  
    backgroundColor: Colors.green,  
),  
),  
);  
}  
}
```



*Ιωάννης Αργυρίου, Εφαρμογή για κινητές συσκευές εντοπισμού,
πλοήγησης προς και χρήσης απινιδωτών*

Κύριος φάκελος

Αρχείο «future.dart»



```
import 'dart:async';
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:latlong/latlong.dart';

import 'models/aed.dart';

Future<List<Aed>> getAllAeds() async {
  final response = await
http.get('http://{SERVER_IP}:{SERVER_PORT}/aeds/');

  if (response.statusCode == 200) {
    // If the call to the server was successful, parse the JSON.
    return (json.decode(response.body) as Iterable)
      .map((i) => Aed.fromJson(i))
      .toList();
  } else {
    // If that call was not successful, throw an error.
    throw Exception('Failed to get AED list');
  }
}

Future<Aed> getNearestAed(
  String accessToken,
  LatLng source,
  List<Aed> aedList,
) async {
  var minDistance = double.infinity;
  Aed minDistanceAed;

  for (Aed aed in aedList) {
    final response = await http.get(
      'https://api.mapbox.com/directions/v5/mapbox/driving-
traffic/${source.longitude},${source.latitude};${aed.location.latLng.
longitude},${aed.location.latLng.latitude}?access_token=$accessToken'
    );

    if (response.statusCode == 200) {
      // If the call to the server was successful, parse the JSON.
      final aedDistance =
json.decode(response.body)['routes'][0]['distance'];

      if (aedDistance < minDistance) {
        minDistance = aedDistance;
        minDistanceAed = aed;
      }
    } else {
      // If that call was not successful, throw an error.
      throw Exception('Failed to get AED\'s (id: ${aed.id}) distance
from user location');
    }
  }

  return minDistanceAed;
}
```



Αρχείο «intent.dart»

```
import 'package:latlong/latlong.dart';
import 'package:android_intent/android_intent.dart';
import 'package:call_number/call_number.dart';

void navigationIntent(LatLng endPoint) async {
  final AndroidIntent intent = AndroidIntent(
    action: 'action_view',
    data: Uri.encodeFull(

'google.navigation:q=${endPoint.latitude},${endPoint.longitude}',
    ),
  );

  await intent.launch();
}

void callIntent(String phoneNumber) async {
  await CallNumber().callNumber(phoneNumber);
}
```



Αρχείο «main.dart»

```
import 'package:flutter/material.dart';

import 'screens/aed_details_screen.dart';
import 'screens/aed_instructions_screen.dart';
import 'screens/aed_map_screen.dart';
import 'screens/tabs_screen.dart';

void main() => runApp(MyApp());

class MyApp extends StatefulWidget {
  MyApp({Key key}) : super(key: key);

  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Εφαρμογή Απινιδωτών',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      initialRoute: '/',
      routes: {
        '/': (ctx) => TabsScreen(),
        AedMapScreen.route: (ctx) => AedMapScreen(),
        AedDetailsScreen.route: (ctx) => AedDetailsScreen(),
        AedInstructionsScreen.route: (ctx) =>
AedInstructionsScreen(),
      },
    );
  }
}
```