

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΜΕ ΤΗ ΧΡΗΣΗ ΑΡΙ ΚΑΙ GOOGLE
MAPS ΜΕ ΣΚΟΠΟ ΤΗΝ ΕΥΡΕΣΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ
ΣΗΜΕΙΩΝ ΔΕΙΓΜΑΤΟΛΗΨΙΑΣ ΓΙΑ ΔΙΕΝΕΡΓΕΙΑ RAPID TEST
ΜΕΣΩ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΟΥ ΥΠΟΥΡΓΕΙΟΥ ΥΓΕΙΑΣ.**

Κωνσταντίνος Μιχαήλ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη εφαρμογής με τη χρήση API και Google Maps με σκοπό την εύρεση και παρουσίαση των σημείων δειγματοληψίας για διενέργεια Rapid Test μέσω του προγράμματος του Υπουργείου Υγείας.

Κωνσταντίνος Μιχαήλ

Επιβλέπων Καθηγητής

Δρ. Χρυσάνθου Γιώργος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2022

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου για την διπλωματική εργασία, Δρ. Γιώργο Χρυσάνθου που με καθοδήγησε να ασχοληθώ με το συγκεκριμένο θέμα και σεβάστηκε την επιθυμία μου να δουλέψω με συγκεκριμένα πράγματα όπως τη χρήση API's και χρησιμοποίηση της γλώσσας προγραμματισμού Java, όπως επίσης και τις κατευθυντήριες γραμμές που μου προσφέρθηκαν για την συγγραφή της πτυχιακής μου.

Επίσης, ιδιαίτερες ευχαριστίες στην Σάβια Μαχλουζαρίδη, για την συνεχή βοήθεια που μου παρείχε από την αρχή της πτυχιακής μου εργασίας, εμπλουτίζοντας με ιδέες και συνεχή ανταπόκριση κατά την διάρκεια της υλοποίησης.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για και ιδιαίτερα τους γονείς μου, Μυριάνθη και Χριστόδουλο, για όλη τους την υποστήριξη κατά την διάρκεια των σπουδών μου, καθώς και σε κάθε επιλογή που έχω κάνει.

Περίληψη

Τα τελευταία 2.5 χρόνια ο όρος COVID-19 έχει εισέλθει στην ζωή μας και έχουν παρθεί κάποια μέτρα για την αντιμετώπιση του ιού. Σε αυτά τα μέτρα είναι και η συνεχής διενέργεια Rapid Test των πολιτών για καλύτερη αντιμετώπιση του ιού. Το Γραφείο Τύπου και Πληροφοριών της Κύπρου, μέσω του προγράμματος του Υπουργείου Υγείας παρουσιάζει καθημερινά τα σημεία δειγματοληψίας για διενέργεια Rapid Test που περιλαμβάνονται σε αυτό το πρόγραμμα.

Επίσης, τα τελευταία χρόνια ο όρος προγραμματισμός σε πραγματικό χρόνο εισέρχεται ολοένα στο λεξιλόγιο μας. Αυτό οφείλεται κυρίως στην ανάγκη των προγραμματιστών για αυτοματοποιημένη παρουσίαση πληροφοριών χρησιμοποιώντας data storage και επαναχρησιμοποίηση με σκοπό την παρουσίαση των πληροφοριών σε μικρό χρονικό διάστημα. Στη δική μου περίπτωση δεν ήταν δυστυχώς εφικτό να είναι όλες οι λειτουργίες με αυτό το τρόπο.

Επομένως, στην προσπάθεια μου για αυτοματοποιημένη παροχή πληροφοριών στους πολίτες και εύκολη πρόσβαση σε αυτές, με τις ιδέες της ομάδας με την οποία δούλεψα αναπτύξαμε μία εφαρμογή για έξυπνα τηλέφωνα η οποία θα απεικονίζει στους πολίτες τα σημεία δειγματοληψίας που περιλαμβάνονται στο πρόγραμμα και θα προτρέπει τον χρήστη να επιλέξει το ποιο κοντινό του σημείο δειγματοληψίας. Επίσης, ο χρήστης θα μπορεί να δει όλα τα υπόλοιπα σημεία και να επιλέξει αν επιθυμεί κάποιο άλλο, όπως και να αλλάξει τον τρόπο μεταφοράς του στο σημείο. Με αυτό τον τρόπο ο χρήστης της εφαρμογής μπορεί εύκολα να μεταβεί στη ποιο κοντινή του τοποθεσία και να διενεργήσει Rapid Test για προστασία από τον COVID-19.

Στην παρούσα εργασία περιγράφονται οι αποφάσεις που λήφθηκαν κατά την υλοποίηση της εφαρμογής, μεθοδολογίες που χρησιμοποιήσα στη υλοποίηση, προβλήματα που αντιμετώπισα κατά την διάρκεια της υλοποίησης, τα αποτελέσματα της εφαρμογής αλλά και μελλοντικές πιθανές βελτιώσεις που θα μπορούσαν να βοηθήσουν στην μεγαλύτερη ανάπτυξη της εφαρμογής. Η εφαρμογή μου δουλεύει μόνο σε τηλέφωνα με λογισμικό Android.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή.....	1
1.1 Γενική Εισαγωγή	1
1.2 Περιγραφή και κίνητρο	2
1.3 Στόχος εργασίας και συνεισφορά	3
1.4 Σχετικές εφαρμογές	3
1.5 Δομή εργασίας	4
 Κεφάλαιο 2 Θεωρητικό υπόβαθρο.....	 5
2.1 Προγραμματισμός σε πραγματικό χρόνο	5
2.2 Στατικός προγραμματισμός	6
2.3 HTTP Requests/Responses	6
2.4 API's	10
2.5 HTML	11
 Κεφάλαιο 3 Μεθοδολογία.....	 12
3.1 Επισκόπηση	12
3.2 Android Studio – Google Maps	14
3.3 Twitter API	14
3.4 Google Maps API	15
3.5 HTML Website Parsing	15
3.6 SQLite	16
 Κεφάλαιο 4 Υλοποίηση.....	 17
4.1 Δημιουργία βάσης εφαρμογής	17
4.1.1 Δημιουργία χάρτη	17
4.1.2 Προβλήματα που αντιμετωπίστηκαν	18
4.2 Εύρεση συγκεκριμένου tweet	19
4.2.1 Φιλτράρισμα	20

4.2.2	Εύρεση URL από tweet	20
4.2.3	Προβλήματα που αντιμετωπίστηκαν	21
4.3	Εύρεση πληροφοριών από URL	23
4.3.1	HTML Parsing	23
4.3.2	Προβλήματα που αντιμετωπίστηκαν	25
4.4	Εύρεση τοποθεσιών	26
4.4.1	Εύρεση τοποθεσιών χρησιμοποιώντας Google API's	26
4.4.2	Προβλήματα που αντιμετωπίστηκαν	28
4.5	Εύρεση κοντινότερου σημείου δειγματοληψίας	29
4.5.1	Εύρεση δρομολόγιων	29
4.5.2	Εύρεση κοντινότερου σημείου δειγματοληψίας	30
4.5.3	Διαχώριση κοντινότερου σημείου δειγματοληψίας	31
4.6	Αποθήκευση πληροφοριών σε βάση δεδομένων	32
4.6.1	Δομή βάσης δεδομένων	33
4.6.2	Διαγραφή αχρείαστων σημείων δειγματοληψίας	34
Κεφάλαιο 5	Εκτίμηση και αποτελέσματα.....	35
5.1	Ερωτηματολόγιο	35
5.1.1	Παρουσίαση ερωτηματολογίου	35
5.1.2	Ανάλυση ερωτηματολογίου	37
5.2	Αποτελέσματα Ερωτηματολογίου	41
5.2.1	Ανάλυση αποτελεσμάτων	41
5.2.2	Συμπεράσματα αποτελεσμάτων	47
Κεφάλαιο 6	Συμπεράσματα	50
6.1	Σχολιασμός αποτελεσμάτων	50
6.1.1	Αποτελέσματα από Twitter API	51
6.1.2	Αποτελέσματα από Google API	51
6.1.3	Αποτελέσματα από HTML Web Parsing	52
6.1.4	Αποτελέσματα από Android Studio	53
6.1.5	Αποτελέσματα από βάση δεδομένων	54
6.1.6	Αποτελέσματα εφαρμογής σε σχέση με ταχύτητα	54

6.2 Συμπεράσματα	54
6.3 Μελλοντική εργασία	55
Βιβλιογραφία	57
Παράρτημα Α.....	A-9

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή	1
1.2 Περιγραφή και κίνητρο	2
1.3 Στόχος εργασίας και συνεισφορά	3
1.4 Σχετικές εφαρμογές	3
1.5 Δομή εργασίας	4

1.1 Γενική Εισαγωγή

Τα τελευταία χρόνια η τεχνολογία έχει γίνει αναπόσπαστο κομμάτι της καθημερινότητας μας. Με την πρόοδο της τεχνολογίας αναπτύσσονται κάθε μέρα νέες συσκευές, νέες εφαρμογές οι οποίες αποσκοπούν στην διευκόλυνση της ζωής των ανθρώπων παρέχοντας στους χρήστες αυτοματοποιημένες διαδικασίες καθώς και διάφορες άλλες ευκολίες που μπορούν να παρέχουν στους χρήστες χρησιμοποιώντας διεπιφάνειες. Σήμερα, σε αντίθεση με ποιο παλιές εποχές, η ανάγκη για πληροφορίες είναι αυξημένη και ταυτόχρονα η ευκολία που παρέχονται οι πληροφορίες είναι εξίσου σημαντική. Οι χρήστες πλέον, μπορούν με μεγάλη ευκολία να ψάξουν για οποιαδήποτε πληροφορία χρειάζονται μέσω των μηχανών αναζήτησης, με εντυπωσιακά αποτελέσματα.

Με την ανάπτυξη τεχνολογιών όπως τα GPS, Satellite Maps μπορούν να προσφέρουν τεράστια βοηθήματα ως προς τις μετακινήσεις των ανθρώπων, αποφυγή ατυχημάτων και παρουσίαση ευκολιών στους χρήστες αναλόγως συνθηκών. Η ανάπτυξη του συγκεκριμένου τομέα στο συγκεκριμένο επιστημονικό πλαίσιο έχει καταφέρει να επιλύσει προβλήματα στις ζωές των ανθρώπων που κάποτε θεωρούνταν απίθανα, με πλέον άμεση εφαρμογή στη πραγματική ζωή.

Προβλήματα όπως λανθασμένες τοποθεσίες, δυσκολίες εύρεσης μιας τοποθεσίας που παλαιότερα θα χρειάζονταν χρόνο για να λυθούν, τώρα δεν υπάρχουν πλέον με τη χρήση

των GPS και Satellite Maps καθώς δίνουν πρόσβαση στο χρήστη σε ακριβείς πληροφορίες.

Όλα τα ποιο πάνω, σε συνδυασμό με μια εφαρμογή κινητών τηλεφώνων είναι ικανά να προσφέρουν σε πολλούς ανθρώπους διευκολύνσεις, χωρίς περιττές αναζητήσεις και κινήσεις.

1.2 Περιγραφή και κίνητρο

Από την αρχή της πτυχιακής μου εργασίας, ήθελα να βρω θέμα το οποίο θα έχει κοινό όφελος και μπορεί να βοηθήσει πολλούς πολίτες. Προσπάθησα να σκεπτώ προβλήματα τα οποία πολλοί άνθρωποι βιώνουν καθημερινά, και να προσφέρω μέσω της εφαρμογής μου λύσεις. Λόγω της κατάστασης με τον CO VID-19 και των διαφόρων μέτρων που εφάρμοσε η πολιτεία στα πλαίσια της καταπολέμησης του ιού, πολλοί πολίτες θα χρειάζονται ανά τακτικά διαστήματα διενέργεια Rapid Test. Έτσι αποφάσισα ότι είναι μια καλή ιδέα να συνδυάσω τις διενέργειες Rapid Test με μια εφαρμογή που θα μπορούσε να προσφέρει στους χρήστες της τα σημεία όπου γίνονται διενέργειες Rapid Test με όλες τις πληροφορίες που χρειάζονται, για εύκολη πρόσβαση των χρηστών σε αυτές τις πληροφορίες με το πάτημα ενός κουμπιού.

Εμπνευσμένος από αυτή την ιδέα και με σκοπό να βοηθήσουμε άτομα που θα χρειάζονταν συχνά διενέργεια Rapid Test, ανέπτυξα την εφαρμογή που βοηθά τον χρήστη στην εύρεση των κέντρων διενέργειας Rapid Test, χρησιμοποιώντας πληροφορίες οι οποίες δίνονται καθημερινώς από το Γραφείο Τύπου και Πληροφοριών, και εμπλουτίζοντας τις τοποθετώντας τις τοποθεσίες σε ένα χάρτη όπου ο χρήστης θα μπορεί να δει όλες τις διαδρομές, όλες τις πληροφορίες και ταυτόχρονα να επιλέξει και διαφορετικά κέντρα συγκρίνοντας ό,τι επιθυμεί.

Η εφαρμογή που αναπτύχθηκε χρησιμοποιεί Google Maps API, Twitter API με σκοπό την εύρεση των πληροφοριών, και στη συνέχεια την αξιοποίηση των πληροφοριών χρησιμοποιώντας χάρτες που προσφέρει η Google.

1.3 Στόχος εργασίας και Συνεισφορά

Αυτή η εργασία αποσκοπεί στην ανάπτυξη μιας εφαρμογής κινητών Android που μπορεί να βοηθήσει άτομα για εύκολη πρόσβαση σε πληροφορίες που χρειάζονται για την διενέργεια Rapid Test μέσω του προγράμματος του Υπουργείου Υγείας.

Λόγω της επιθυμίας μου για υλοποίηση μιας εφαρμογής με βάση τον προγραμματισμό σε πραγματικό χρόνο που θα βοηθούσε όλους τους πολίτες με ένα θέμα επίκαιρο όπως ο υιός COVID-19, πλησίασα την ομάδα Smart City του ερευνητικού κέντρου RISE. Η ερευνητική ομάδα η οποία ήδη ήθελε να αξιοποιήσει την πλατφόρμα του Twitter, την πόλη, καθώς και πληροφορίες που συνεχώς αλλάζουν, με βοήθησε να σκεπτώ και να αξιοποιήσω τα σημεία δειγματοληψίας Rapid Test τα οποία αλλάζουν καθημερινώς. Επιπρόσθετα, ένα μέλος της ομάδας του Smart City ήθελε να συμπεριλάβει την δυνατότητα της δρομολόγησης για τα σημεία δειγματοληψίας, λόγω του ότι αυτό συμπίπτει με ένα από τους λόγους του Smart City, να είναι αποδοτικό και βοηθητικό.

Επίσης, μέσω αυτής της εργασίας γίνεται η προσπάθεια για εύκολη πρόσβαση των χρηστών σε διάφορες πληροφορίες που προσφέρουν διάφοροι κυβερνητικοί φορείς στην Κύπρο.

1.4 Σχετικές εφαρμογές

Μια σχετική εφαρμογή που με ενέπνευσε προς την υλοποίηση της πτυχιακής ήταν από το πανεπιστήμιο του Heidelberg. Συγκεκριμένα, μετά από την σκέψη μου και τη συζήτηση με την ομάδα του Smart City, ανακαλύψαμε αυτή τη εφαρμογή μέσα στις προσπάθειες μας να βρούμε OpenStreetMap services που είχαν δρομολόγια για ποδηλάτες επειδή η Google δεν υποστηρίζει αυτή την λειτουργία για την Κύπρο. Επομένως, είχαμε βρει ότι το πανεπιστήμιο του Heidelberg «εκμεταλλεύτηκε» την κατάσταση με τον υιό, και υλοποίησαν μια διαδικτυακή εφαρμογή που είχε ως βάση το OpenStreetMap.

Η εφαρμογή τους δίνει την ευκαιρία στο χρήστη να βρει το ποιο κοντινό σε εκείνον κέντρο εμβολιασμού όπως και επίσης κέντρα διενέργειας Rapid Test. Υπήρχαν επίσης

πληροφορίες και για υπόλοιπα κέντρα εμβολιασμού και διενέργειας Rapid Test εκτός και από το ποιο κοντινό.

Η εφαρμογή του υποστηρίζει μόνο κάποιες συγκεκριμένες χώρες, όπως Γερμανία , Ελβετία, και Γαλλία [5].

1.5 Δομή εργασίας

Στο δεύτερο κεφάλαιο γίνονται περιγραφές για όρους και μεθοδολογίες οι οποίες χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Ακόμα, γίνεται αναφορά στο προγραμματισμό σε πραγματικό χρόνο, στο στατικό προγραμματισμό, στα HTTP Request/Responses καθώς και στη χρήση διαφόρων API's. Τέλος, γίνεται αναφορά στη σχετική βιβλιογραφία που χρησιμοποιήθηκε για ανάπτυξη αυτής της εφαρμογής.

Στο τρίτο κεφάλαιο γίνεται αναφορά στη μεθοδολογία της εργασίας, μια περιγραφή δηλαδή των διαφόρων τεχνικών και αλγορίθμων που χρησιμοποιήσαμε.

Στο τέταρτο κεφάλαιο περιγράφουμε την υλοποίηση της εφαρμογής , όλα τα στάδια της ανάπτυξης καθώς και διάφορα προβλήματα τα οποία αντιμετωπίσαμε κατά την διάρκεια της υλοποίησης.

Στο πέμπτο κεφάλαιο γίνεται παρουσίαση και ανάλυση του ερωτηματολογίου και στη συνέχεια εξάγουμε συμπεράσματα από το ερωτηματολόγιο.

Στο έκτο και τελευταίο κεφάλαιο δίνονται συμπεράσματα σχετικά με την υλοποίηση μας όπως και επίσης πιθανούς τρόπους βελτίωσης της υλοποίησης στο μέλλον.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Προγραμματισμός σε πραγματικό χρόνο	5
2.2 Στατικός προγραμματισμός	6
2.3 HTTP Requests/Responses	6
2.4 API's	10
2.5 HTML	11

2.1 Προγραμματισμός σε πραγματικό χρόνο

Ο όρος προγραμματισμός σε πραγματικό χρόνο χρησιμοποιείται ολοένα και περισσότερο στις σύγχρονες υλοποιήσεις στο τομέα της πληροφορικής καθώς θεωρείται ένα μεγάλο βήμα ως προς την βελτίωση της απόδοσης αλγορίθμων που χρησιμοποιούνται για την επίλυση κάποιου προβλήματος[3].

Προγραμματισμός σε πραγματικό χρόνο είναι η επιστημονική έννοια της Πληροφορικής που χρησιμοποιεί δεδομένα σε πραγματικό χρόνο, για παράδειγμα από ένα συμβάν σε μια ανταπόκριση ενός συστήματος. Ο σκοπός του προγραμματισμού σε πραγματικό χρόνο είναι η ανταπόκριση σε σύντομο χρονικό διάστημα ενός προγράμματος, χρησιμοποιώντας δεδομένα σε πραγματικό χρόνο.

Ένα σύστημα σε πραγματικό χρόνο περιγράφεται ως το σύστημα που ελέγχει ένα περιβάλλον στο οποίο λαμβάνει δεδομένα, τα επεξεργάζεται και επιστρέφει τα αποτελέσματα με σχετική ταχύτητα.

Στη δική μας περίπτωση, χρησιμοποιούμε προγραμματισμό σε πραγματικό χρόνο δημιουργώντας μια ακολουθία από ενέργειες με μεταβλητές οι οποίες δεν είναι από την αρχή αρχικοποιημένες και ενδέχεται να αλλάζουν καθημερινά με στόχο ο χρήστης να μην επεμβαίνει δίνοντας εντολές, όπως και επίσης με δεδομένα τα οποία καθημερινά

αλλάζουν με αποτέλεσμα την παρουσίαση πληροφοριών που μπορεί να είναι διαφορετικές σε καθημερινή βάση.

2.2 Στατικός προγραμματισμός

Ο όρος στατικός προγραμματισμός χρησιμοποιείται στη πληροφορική από την αρχή του τομέα της πληροφορικής. Είναι η διαδικασία εκτέλεσης ενός υπολογισμού με κάποια ακολουθία ενεργειών, οι οποίες δεν γίνονται αυτοματοποιημένα και ίσως χρειάζονται οδηγίες/εντολές από τον χρήστη για λήψη κάποιων αποφάσεων. Επίσης αποτελείται από μεταβλητές οι οποίες πρέπει να είναι γνωστές πριν την διαδικασία του compiling process[1].

Παραδείγματα στατικού προγραμματισμού υπάρχουν πάρα πολλά, καθώς πάρα πολλά προβλήματα της πληροφορικής λύνονται με την χρήση του στατικού προγραμματισμού.

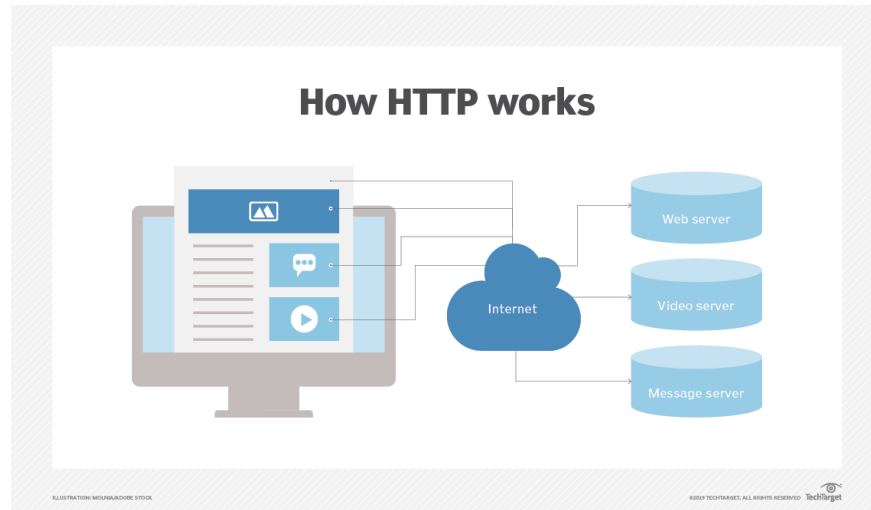
Στην δική μας περίπτωση, χρησιμοποιούμε στατικό προγραμματισμό δημιουργώντας μια ακολουθία από ενέργειες οι οποίες είναι προκαθορισμένες, με μεταβλητές οι οποίες δεν μπορούν να αλλάξουν κατά την διάρκεια του αλγόριθμου. Τυχόν αλλαγές στις συγκεκριμένες ενέργειες είναι πιθανών να επιφέρουν καταστροφικές συνέπειες στην εφαρμογή μας.

2.3 HTTP Requests/Responses

Το HTTP (Hypertext Transfer Protocol) είναι ένα πρωτόκολλο που βρίσκεται στο application layer. Το HTTP είναι η βάση της επικοινωνίας των δεδομένων στο διαδίκτυο όπου οι χρήστες μπορούν με ευκολία να έχουν πρόσβαση σε άλλες πληροφορίες με την χρήση των Web Browser. Ακόμα, το HTTP είναι ένα πρωτόκολλο το οποίο χρησιμοποιείται σε συνδυασμό με το TCP/IP σετ πρωτοκόλλων. Το HTTP αναπτύχθηκε από τον Tim Berners-Lee το 1989[4].

Χρησιμοποιώντας το HTTP μπορούμε να μεταφέρουμε διάφορα αρχεία, όπως κείμενα, εικόνες, ήχο, βίντεο και πληροφορίες μέσω του διαδικτύου. Μέσω του πρωτοκόλλου μπορούν τα διάφορα αρχεία που αναφέραμε πιο πάνω να μεταφέρονται από clients και

servers. Ένας client ο οποίος αναζητά η χρειάζεται πληροφορίες για κάποιο σκοπό, στέλνει ένα αίτημα request στον server για τις πληροφορίες που χρειάζεται, και ο server θα απαντήσει με ένα response δίνοντας πίσω στο χρήστη της πληροφορίες που ζήτησε.



Σχήμα 2.1 Απλή αναπαράσταση του HTTP Protocol[9].

Τα request στο πρωτόκολλο HTTP γίνονται από τους χρήστες (clients) και αποστέλλονται με μια συγκεκριμένη μορφή προς τον παραλήπτη (συνήθως τον server) ο οποίος πρέπει να απαντήσει στον client με τις ανάλογες πληροφορίες.

Κάθε request αποτελείται από κάποιες πληροφορίες όπως την έκδοση του HTTP Protocol (HTTP or HTTP/2), ένα URL το οποίο αναφέρεται σε κάποιο πόρο στο διαδίκτυο, μια HTTP μέθοδο, ένα HTTP header το οποίο περιέχει πληροφορίες σχετικά με το browser που χρησιμοποιείται και τι είδους δεδομένα χρειάζεται το request, και τέλος ένα HTTP Body το οποίο περιέχει πληροφορίες που ίσως να χρειαστεί ο server σε περίπτωση κάποιου authentication (username/password), αρχεία ή μικρές πληροφορίες που μπορεί να χρειαστεί να υποβληθούν στη σελίδα.

Τα request methods του πρωτοκόλλου είναι ιδιαίτερα σημαντικά καθώς θα καθορίσουν το είδος του request. Οι μέθοδοι που χρησιμοποιούνται για την αποστολή ενός request είναι:

- GET: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι απλά αναζητεί πληροφορίες και δεν αποσκοπεί στην αποστολή πληροφοριών στον server.

- HEAD: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι απλά θέλει να μάθει την κατάσταση που βρίσκεται ο server.
- POST: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι θέλει να στείλει κάποιες πληροφορίες στον server οι οποίες θα πρέπει να χρησιμοποιηθούν. Για παράδειγμα, χρησιμοποιείται σε περίπτωση που θέλει να ολοκληρώσει μια συναλλαγή, ή να τοποθετήσει ένα σχόλιο σε κάποιο internet forum.
- PUT: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι θέλει να στείλει κάποιες πληροφορίες, οι οποίες πρέπει να αντικαταστήσουν τυχόν άλλες πληροφορίες στο ίδιο URL που το αποστέλλει. Χρησιμοποιείται δηλαδή για ενέργειες όπως edit κάποιας πληροφορίας του profile μας σε κάποια ιστοσελίδα.
- DELETE: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι επιθυμεί τη διαγραφή πληροφοριών στο συγκεκριμένο URL που το αποστέλλει.
- OPTIONS: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι επιθυμεί να δει ποια HTTP Methods υποστηρίζει ο server.
- CONNECT: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι επιθυμεί να ζητήσει από το ενδιαμέσο, να δημιουργήσει ένα TCP/IP Tunnel προς τον origin server.
- TRACE: Με τη χρήση αυτής της μεθόδου, ο χρήστης υποδηλώνει ότι θέλει από τον χρήστη να του επιστρέψει το request που παρέλαβε. Με αυτό το τρόπο μπορεί να δει αν έγιναν αλλαγές στο request από ενδιάμεσους του δικτύου.

```
GET / HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Σχήμα 2.2 Ένα client request παράδειγμα

Τα responses στο πρωτόκολλο HTTP γίνονται από τους παραλήπτες (server) και αποστέλλονται με μια συγκεκριμένη μορφή προς τον χρήστη (client) και περιέχουν πληροφορίες που σχετίζονται με το request που έκανε προηγουμένως ο client.

Κάθε response αποτελείται από ένα status line και το response header. Το status line περιέχει την έκδοση του πρωτοκόλλου όπως και ένα status response code που υποδεικνύει το status του response. Το response header προσδιορίζει το τρόπο με τον οποίο είναι δομημένο το response.

Το status line μπορεί να πάρει τις ακόλουθες τιμές :

- 1xx-Informational: Υποδεικνύει ότι το request έχει ληφθεί και γίνεται η διαδικασία.
- 2xx-Successful: Υποδεικνύει ότι το request έχει ληφθεί με επιτυχία, έχει γίνει κατανοητό και δεκτό.
- 3xx-Redirection: Υποδεικνύει ότι πρέπει να γίνουν επιπλέον ενέργειες για ολοκλήρωση του request.
- 4xx-Client Error: Υποδεικνύει ότι υπάρχει πρόβλημα στο request. Πιθανώς κακή σύνταξη η δεν μπορεί να πραγματοποιηθεί.
- 5xx-Server Error: Υποδεικνύει ότι υπάρχει πρόβλημα από τη πλευρά του server σε ένα σωστό request.

Μερικές πιθανές δομές του response header (content-type) :

- Text/html
- Text/xml
- Application/json (Στην δική μας περίπτωση)
- Video/mpeg
- Video/mp4

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 155
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
  <head>
    <title>An Example Page</title>
  </head>
  <body>
    <p>Hello World, this is a very simple HTML document.</p>
  </body>
</html>
```

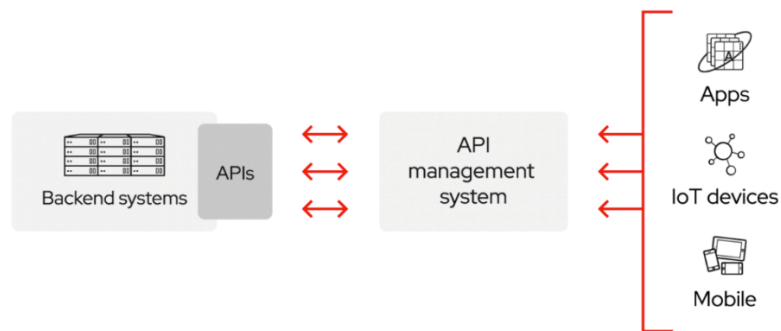
Σχήμα 2.3 Ένα server response παράδειγμα

2.4 API's

Ο όρος API (Application Programming Interface) χρησιμοποιείται όλο και περισσότερο στις σύγχρονες εφαρμογές πληροφορικής. Έχει ως σκοπό την παροχή υπηρεσιών διαφόρων υλοποιήσεων από διαφορετικά λογισμικά χωρίς απαραίτητα να ξέρει ο χρήστης πώς έχουν δημιουργηθεί[11]. Με αυτό το τρόπο αξιοποιούνται διάφοροι πόροι καθώς ευκολύνει πολύ την δημιουργία λογισμικών, κάνοντας οικονομία σε χρόνο και χρήμα. Ακόμα, παρέχει ασφάλεια καθώς μπορούν να χρησιμοποιηθούν οι υλοποιήσεις οποιουδήποτε λογισμικού χωρίς να υπάρχει γνώση της backend υλοποίησης.

Τα API's χρησιμοποιούνται πλέον για διευκόλυνση των υπηρεσιών σε πάρα πολλές μας καθημερινές ενέργειες. Για παράδειγμα, το Google Maps που χρησιμοποιείται σε καθημερινή χρήση από πολλούς χρήστες. Επίσης ένα άλλο παράδειγμα είναι κάθε φορά που αποστέλλουμε ένα μήνυμα στην εφαρμογή Messenger, χρησιμοποιείται ένα API για να σταλεί η πληροφορία.

Για να γίνει αυτή η συνεννόηση μεταξύ δύο λογισμικών, χρησιμοποιείται ένα προκαθορισμένο endpoint (subroutine, methods, request) το οποίο χρησιμοποιεί HTTP Request και περιέχει τις απαραίτητες πληροφορίες για την διεκπεραίωση του αιτήματος. Αυτά τα API's που χρησιμοποιούν HTTP Requests ονομάζονται και REST APIs. Μετά την αποστολή του request, ο server που διαχειρίζεται το API επιστρέφει ένα response με τις απαραίτητες πληροφορίες που ζήτησε ο χρήστης σε διάφορες μορφές όπως HTML, XML, JSON. Μετά, ο client που έχει στείλει το request είναι υπεύθυνος για το πως θα διαχειριστεί της πληροφορίες που πήρε από το response. Ο client μπορεί να μην χρειάζεται όλες τις πληροφορίες, έτσι με parsing που προσφέρεται εύκολα σε δομές όπως το XML και JSON μπορεί να φιλτράρει τις πληροφορίες και να χρησιμοποιήσει μόνο όσες επιθυμεί.



Σχήμα 2.4 Σχεδιάγραμμα παραδείγματος API[8]

2.5 HTML

Ο όρος HTML (Hypertext Markup Language) είναι πλέον ευρέως χρησιμοποιημένος καθώς είναι ο τρόπος με τον οποίο περιγράφεται το περιεχόμενο και η δομή της πληροφορίας σε μια ιστοσελίδα. Η γλώσσα HTML προτάθηκε από τον Tim Berners-Lee το 1992 με την αρχική έκδοση HTML 1.0 και αργότερα συστήθηκε η έκδοση HTML 2.0 με περισσότερες και καλύτερες δυνατότητες[6].

Με τη χρήση της γλώσσας HTML δίνει τη δυνατότητα για ενσωμάτωση υπερσυνδέσμων σε μια ιστοσελίδα. Η δομή της είναι απλή και περιγράφει την λογική διάταξη μιας σελίδας.

Αποτελείται από ένα HTML element το οποίο έχει ένα starting tag και ένα closing tag. Επίσης, ένα element συχνά περιέχει και άλλα elements. Μέσα στο html element υπάρχει το head element το οποίο περιέχει πληροφορίες σχετικά με το αρχείο. Μετά από το head element υπάρχει το body element στο οποίο βρίσκεται το περιεχόμενο της σελίδας.

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Example</title>
5      <link rel="stylesheet" href="sty
6    </head>
7    <body>
8      <h1>
9        <a href="/">Header</a>
10     </h1>
11     <nav>
12       <a href="one/">One</a>
13       <a href="two/">Two</a>
14       <a href="three/">Three</a>
15     </nav>

```

Σχήμα 2.5 Απλό παράδειγμα HTML αρχείου

Σήμερα, χρησιμοποιούμε την έκδοση HTML5 η οποία μας δίνει δυνατότητες για ιστιακές εφαρμογές με περισσότερη υποστήριξη αλληλεπιδραστικότητας, βίντεο, γραφικών, εφέ και APIs. Επίσης, προσαρμόζεται εύκολα σε κάθε συσκευή, όπως Η/Υ, κινητό, τάμπλετ, ακόμα και τηλεόραση. Τέλος, είναι ανοικτή πλατφόρμα με δωρεάν αδειοδότηση.

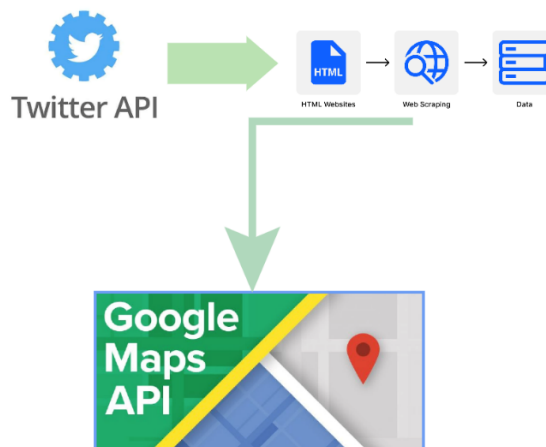
Κεφάλαιο 3

Μεθοδολογία

3.1 Επισκόπηση	12
3.2 Android Studio – Google Maps	14
3.3 Twitter API	14
3.4 Google Maps API	15
3.5 HTML Website Parsing	15
3.6 SQLite	16

3.1 Επισκόπηση

Η μεθοδολογία μου στηρίζεται κυρίως στη χρησιμοποίηση δύο API's, σε συνδυασμό με HTML Website Parsing. Έχω ως βάση ένα Google Maps χάρτη στο Android Studio.



Σχήμα 3.1 Επισκόπηση Μεθοδολογίας

Αρχικά, θα χρησιμοποιήσω το Twitter API με σκοπό το φιλτράρισμα των tweets του Γραφείου Τύπου και Πληροφοριών (υπάρχουν δύο χρήστες, @PIOcy που είναι το Ελληνικό και @CyprusPIO το οποίο είναι το Αγγλικό). Όταν βρω το συγκεκριμένο tweet που αναζητώ, ανακτώ από αυτό το URL που περιέχεται το οποίο με οδηγεί στην ιστοσελίδα του Γραφείου Τύπου όπου και υπάρχουν οι πληροφορίες για τα σημεία δειγματοληψίας που χρειάζομαι.

Στη συνέχεια, στην ιστοσελίδα που ανάκτησα ποιο πάνω χρησιμοποιώ HTML Website Parsing με σκοπό την ανάκτηση των πληροφοριών που εμπεριέχονται στη ιστοσελίδα. Από την ιστοσελίδα χρειάζομαι τις πληροφορίες για τους δικαιούχους, καθώς και τα σημεία δειγματοληψίας τα οποία βρίσκονται πιο κάτω σε μια μορφή table. Οι ποιο πάνω πληροφορίες ανακτώνται στην μορφή αντικειμένων (objects).

Στη βάση των αποφάσεων του Υπουργικού Συμβουλίου, και με στόχο την αναχαίτιση της διασποράς του ιού SARS-CoV-2, στους χώρους όπου η πρόσβαση προϋποθέτει την κατοχή SafePass, τα άτομα ηλικίας 6 ετών και άνω πρέπει να πληρούν μια από τις πιο κάτω προϋποθέσεις:

1. Πιστοποιητικό πλήρους εμβολιασμού.
2. Αποδεικτικό ότι ένα άτομο έχει νοσήσει από την ασθένεια COVID-19 τους τελευταίους 3 μήνες.
3. Πιστοποιητικό αρνητικού rapid test με ισχύ 7 ημερών για άτομα 6-11 ετών.
4. Πιστοποιητικό αρνητικού PCR ή rapid test με ισχύ 72 ωρών για άτομα 12-17 ετών.
5. Πιστοποιητικό αρνητικού test, PCR με ισχύ 72 ωρών ή rapid test με ισχύ 48 ωρών για άτομα 18 ετών και άνω, και τουλάχιστον μια δόση εμβολίου εκεί όπου απαιτείται βάσει Διατάγματος.
6. Πιστοποιητικό αρνητικού rapid test με ισχύ 24 ωρών για άτομα 12 ετών και άνω εκεί όπου απαιτείται βάσει Διατάγματος.

Σύμφωνα με την απόφαση του Υπουργικού Συμβουλίου, στις κινητές μονάδες rapid test του Υπουργείου Υγείας δύνανται να εξημετρώνται μόνο οι πιο κάτω κατηγορίες πολιτών:

- Τα ανήλικα πρόσωπα μέχρι 17 ετών. Τα παιδιά ηλικίας 12-17 ετών, τα οποία δεν έχουν τη συγκατάθεση των γονέων/νόμιμων κηδεμόνων τους για εμβολιασμό, δύνανται να διενεργούν δωρεάν rapid test μέσω του προγράμματος του Υπουργείου Υγείας.
- Τα άτομα ηλικίας 18 ετών και άνω, τα οποία δεν μπορούν να υποβληθούν σε εμβολιασμό με τεκμηρίωση με ιατρικό πιστοποιητικό, και αφού το αίτημά τους εξετάσει από τον Παιδιατρικό Ιατρικό Σύλλογο και το Υπουργείο Υγείας και δοθεί σχετική έγκριση, η οποία πρέπει να παρουσιάζεται στη μονάδα δειγματοληψίας. Στην εν λόγω κατηγορία εμπίπτουν και οι εγκυμονούσες, νοσημένοι ότι έχουν εξασφαλίσει έγκριση από την ad hoc επιτροπή του ΠΙΣ.
- Τα άτομα που εμβολιάζονται με την 1^η δόση του εμβολίου, για την περίοδο που απαιτείται μεταξύ της 1^{ης} και της 2^{ης} δόσης του εμβολίου. Για σκοπούς επαλήθευσης θα πρέπει να παρουσιάζεται η Κάρτα Εμβολιασμού.
- Τα άτομα που έχουν ολοκληρώσει το εμβολιαστικό τους σχήμα και έχει παρέλθει η περίοδος 7 μηνών χωρίς να λάβουν ενισχυτική δόση, μέχρι νεωτέρας. Για σκοπούς επαλήθευσης θα πρέπει να παρουσιάζεται η Κάρτα Εμβολιασμού.
- Τα άτομα που έχουν λάβει ενισχυτική δόση του εμβολίου, αλλά επιθυμούν να υποβληθούν σε test για σκοπούς παρακολούθησης της υγείας τους ή για άλλους λόγους. Νοείται ότι θα πρέπει απαραίτητα να προσκομίζεται η Κάρτα Εμβολιασμού.
- Τα άτομα που κατέχουν πιστοποιητικό ανάρρωσης από τη νόσο COVID-19 εντός των τελευταίων 90 ημερών, αλλά επιθυμούν να υποβληθούν σε test για σκοπούς παρακολούθησης της υγείας τους ή για άλλους λόγους. Νοείται ότι θα πρέπει απαραίτητα να παρουσιάζεται το πιστοποιητικό ανάρρωσης.

Επιπλέον των πιο πάνω αποδεικτικών, υπενθυμίζεται ότι όλοι οι πολίτες οφείλουν απαραίτητα να προσκομίζουν αποδεικτικό ταυτότητας (π.χ. Ταυτότητα, Άδεια Οδήγησης, Διαβατήριο, κτλ), για σκοπούς ταυτοποίησης.

Επαρχία	Σημεία δειγματοληψίας	Ωράριο	Τηλέφωνο
Λευκωσία (15 σημεία)	Nicosia Mall	9 π.μ. - 6 μ.μ.	94041843
	Ιερός Ναός Αγίου Γεωργίου, Λατσιά	9 π.μ. - 6 μ.μ.	94056785
	Κοινωνικό Ιατρείο Ιδαίου	9 π.μ. - 6 μ.μ.	96812424
	Παλιό Κοινωνικό Ιατρείο Ανδόπολης (δίπλα από Κοινωνικό Συμβούλιο)	9 π.μ. - 6 μ.μ.	99969931
	Αίθουσα Ιερού Ναού Αγίου Χαραλάμους αυτοπεργασίας Γερσιού	9 π.μ. - 6 μ.μ.	96625766
	Ιερός Ναός Αγίου Δημητρίου, Ακρόπολη	9 π.μ. - 6 μ.μ.	99146623
	Ενοκλήρια Αγίου Σπυρίδωνα (πίσω από Αστυνομικό Σταθμό Λυκαβήτσου)	9 π.μ. - 6 μ.μ.	96625766
	Εκκλησία Αγίου Πολυδώρου, προάστια, Καίμακλι	9 π.μ. - 6 μ.μ.	99365616
	Προαύριο Αμφιθέατρου Λακατάμους, πίσω από Πινακοθήκη	9 π.μ. - 6 μ.μ.	99965920

Σχήμα 3.2 Παράδειγμα πληροφοριών για τους δικαιούχους και σημεία δειγματοληψίας

Συνεχίζοντας, χρησιμοποιώ το API που προσφέρει η Google για τα Google Maps. Κατ' αρχήν, χρησιμοποιώ το Geocoding API με σκοπό να ανακτήσω τις συντεταγμένες των σημείων δειγματοληψίας, και στη συνέχεια τοποθετώ τις συντεταγμένες στο χάρτη των οποίου δημιούργησα εξ αρχής. Επιπρόσθετα, χρησιμοποιώ και το Directions API για να βρω δρομολόγια από την τοποθεσία του χρήστη τα οποία αξιοποιώ με σκοπό να βρω το ποιο κοντινό στο χρήστη σημείο δειγματοληψίας (χρησιμοποιώ ως κριτήριο κοντινότερου την διάρκεια (duration)). Τέλος, μετά την ολοκλήρωση των διαδικασιών,

τα σημεία δειγματοληψίας μαζί με κάποιες πληροφορίες τους αποθηκεύονται στην τοπική βάση δεδομένων που δημιουργείται στην κινητή συσκευή του χρήστη με τη χρήση της SQLite[12] με σκοπό την επόμενη φορά που ο χρήστης θα αναζητήσει τα σημεία δειγματοληψίας, να ανακτώνται από την βάση χωρίς να καλούνται ξανά τα προηγούμενα APIs.

3.2 Android Studio – Google Maps

Για τη βάση της εφαρμογής μου χρησιμοποιώ το λογισμικό Android Studio για ανάπτυξη εφαρμογής για κινητά τηλέφωνα με λογισμικό Android[2] σε συνδυασμό με Google Maps.

Επέλεξα την εφαρμογή Android Studio καθώς μου δίνει την δυνατότητα να χρησιμοποιήσω την γλώσσα προγραμματισμού Java που είναι η κύρια γλώσσα που έχω διδαχθεί κατά την περίοδο της ακαδημαϊκής μου εκπαίδευσης. Επίσης, είναι πλήρης συμβατή με το Google Maps και με τα Google API's , και αυτό μου πρόσφερε πάρα πολλές ευκολίες και βοήθειες.

3.3 Twitter API

Ένα από τα APIs που θα χρησιμοποιηθούν είναι το API της γνωστής πλατφόρμας κοινωνικής δικτύωσης, Twitter[13]. Το Twitter API μου επιτρέπει την προγραμματιστική πρόσβαση στη πλατφόρμα Twitter με δυνατότητες όπως ανάκτησης Tweets, Direct Messages, Spaces, Lists, Users και άλλα πολλά.

Ο λόγος που θα χρησιμοποιήσω την πλατφόρμα Twitter είναι ότι καθημερινά, το Γραφείο Τύπου και Πληροφοριών, δημοσιεύει ένα tweet το οποίο περιέχει ένα URL στην ιστοσελίδα του Γραφείου Τύπου και Πληροφοριών, στο οποίο υπάρχουν οι πληροφορίες σχετικά με τα σημεία δειγματοληψίας της επόμενης μέρας. Επίσης, είναι πολύ εύκολο να φιλτράρουμε τις δημοσιεύσεις και επομένως, αποφάσισα ότι είναι ένα πολύ καλό σημείο για να αρχίσω καθώς είναι πολύ εύκολο να χρησιμοποιήσω την δημοσίευση για να ανακτήσω το URL που παραπέμπει στη ιστοσελίδα του Γραφείου Τύπου και

Πληροφοριών, το οποίο θα αργότερα θα χρησιμοποιήσω για να ανακτήσω τις πληροφορίες που χρειάζομαι.

3.4 Google Maps API

Το δεύτερο API που θα χρησιμοποιηθεί είναι το Google Maps API, το οποίο χωρίζεται σε διάφορα κομμάτια όπως το Directions API, Geocoding API, τα οποία και χρησιμοποίησα[7]. Επιπρόσθετα, χρησιμοποίησα το Google Maps sdk για την βάση της Android εφαρμογής μου στο Android Studio. Με το Google Maps API θα έχω τη δυνατότητα να μετατρέψω τα σημεία δειγματοληψίας που υπάρχουν μόνο σαν ονόματα, σε συντεταγμένες, να βρω μέσω των συντεταγμένων και του Directions API δρομολόγια από το σημείο που βρίσκεται ο χρήστης, καθώς και το κοντινότερο σημείο δειγματοληψίας στον χρήστη με κριτήριο την διάρκεια του δρομολογίου (duration).

Ο λόγος που χρησιμοποίησα τη συγκεκριμένη πλατφόρμα της Google, είναι οι πάρα πολλές δυνατότητες που μου παρέχει, καθώς και η μεγάλη της αξιοπιστία σε θέματα όπως της δρομολόγησης, και ανάκτησης συντεταγμένων. Επίσης, με τη χρήση της συγκεκριμένης πλατφόρμας δίνονται μεγάλες ευκολίες στο χρήστη, όπως η μεταφορά του από την εφαρμογή μας στην εφαρμογή Google Maps (η οποία υπάρχει σε όλα τα τηλέφωνα Android), απευθείας στο σημείο δειγματοληψίας το οποίο επέλεξε ο χρήστης, και χρήση της εφαρμογής Google Maps με σκοπό την παρουσίαση ζωντανών οδηγιών στο χρήστη.

3.5 HTML Website Parsing

Για την ανάλυση της ιστοσελίδας του Γραφείου Τύπου και Πληροφοριών χρησιμοποίησα τη μέθοδο HTML Website Parsing με τη χρήση της βιβλιοθήκης για Java HTML Parsing Jsoup[10].

Αυτή η μέθοδος μου επιτρέπει να αναλύσω μια ιστοσελίδα χρησιμοποιώντας τα HTML στοιχεία της για ανάκτηση των πληροφοριών που με ενδιέφεραν. Με αυτό το τρόπο κατάφερα να αναλύσω και να δημιουργήσω αντικείμενα από το πίνακα με τα σημεία δειγματοληψίας που υπάρχει στη ιστοσελίδα.

Η χρήση αυτής της μεθόδου είναι η βάση της εφαρμογής μας προς την χρησιμοποίηση προγραμματισμού σε πραγματικό χρόνο, με σκοπό να ανακτά τις πληροφορίες οι οποίες αλλάζουν καθημερινά χωρίς να υπάρχει κάποιο πρόβλημα και χωρίς την συμμετοχή του χρήστη σε αυτή τη διαδικασία. Να είναι δηλαδή μια αυτοματοποιημένη σειρά ενεργειών η οποία θα εκτελείται χωρίς πρόβλημα όταν οι πληροφορίες αλλάζουν.

3.6 SQLite

Για την αποθήκευση πληροφοριών με σκοπό την αποφυγή των επαναλαμβανόμενων κλήσεων στα διάφορα APIs που χρησιμοποιούνται, χρησιμοποίησα την βιβλιοθήκη SQLite.

Η συγκεκριμένη βιβλιοθήκη είναι μια απλή, γρήγορη και αξιόπιστη εκδοχή μιας SQL βάσης δεδομένων. Είναι πολύ εύκολο να υλοποιηθεί σε κινητές συσκευές, αφού με την υλοποίηση της δημιουργείται ένα τοπικό αρχείο κειμένου στο κινητή συσκευή του χρήστη. Επίσης, η SQLite δεν χρησιμοποιεί κάποια σύνδεση με διακομιστή αφού διαβάζει και γράφει σε τοπικό αρχείο κειμένου στην συσκευή του χρήστη όπως ανάφερα και πιο πάνω.

Κεφάλαιο 4

Υλοποίηση

4.1 Δημιουργία βάσης εφαρμογής	17
4.1.1 Δημιουργία χάρτη	17
4.1.2 Προβλήματα που αντιμετωπίστηκαν	18
4.2 Εύρεση συγκεκριμένου tweet	19
4.2.1 Φιλτράρισμα	20
4.2.2 Εύρεση URL από tweet	20
4.2.3 Προβλήματα που αντιμετωπίστηκαν	21

4.3 Εύρεση πληροφοριών από URL	23
4.3.1 HTML Parsing	23
4.3.2 Προβλήματα που αντιμετωπίστηκαν	25
4.4 Εύρεση τοποθεσιών	26
4.4.1 Εύρεση τοποθεσιών χρησιμοποιώντας Google API's	26
4.4.2 Προβλήματα που αντιμετωπίστηκαν	28
4.5 Εύρεση κοντινότερου σημείου δειγματοληψίας	29
4.5.1 Εύρεση δρομολόγιων	29
4.5.2 Διαδικασία εύρεσης κοντινότερου σημείου δειγματοληψίας	30
4.5.3 Διαχώριση κοντινότερου σημείου δειγματοληψίας	31
4.6 Αποθήκευση πληροφοριών σε βάση δεδομένων	32
4.6.1 Δομή βάσης δεδομένων	33
4.6.2 Διαγραφή αχρείαστων σημείων δειγματοληψίας	34

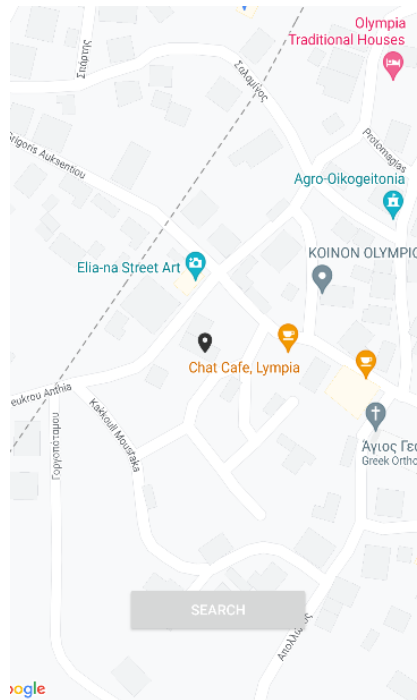
4.1 Δημιουργία βάσης εφαρμογής

Το αρχικό και κυριότερο βήμα της υλοποίησης μου ήταν η βάση της εφαρμογής. Αποφάσισα όπως η αρχική οθόνη της εφαρμογής να είναι ο χάρτης που δημιουργώ με το Google Maps sdk. Επίσης, δημιούργησα και διάφορες άλλες οθόνες για τις λειτουργίες που προσφέρει η εφαρμογή(παρουσίαση όλων των σημείων δειγματοληψίας, σχετικά με εμάς, πληροφορίες). Προσπάθησα ακόμα να κάνω όσο το δυνατόν περισσότερο εύκολη και καλύτερη την εμπειρία του χρήστη με μηνύματα λάθους και κατάλληλες ενδείξεις για τη σωστή λειτουργία της εφαρμογής.

4.1.1 Δημιουργία χάρτη

Με το άνοιγμα της εφαρμογής, η πρώτη οθόνη που εμφανίζεται είναι ο χάρτης. Δημιούργησα το χάρτη χρησιμοποιώντας το Google Maps sdk που προσφέρεται στο Android Studio. Με το άνοιγμα του χάρτη, θέλω να βρω την τοποθεσία του χρήστη. Για να γίνει αυτό εφικτό ο χρήστης πρέπει να έχει ενεργοποιημένο το GPS , σε περίπτωση που το GPS δεν είναι ενεργοποιημένο δίνω τη δυνατότητα στο χρήστη να το ενεργοποιήσει μέσω της εφαρμογής.

Σε κάποιες περιπτώσεις το GPS δεν προλαβαίνει να ανακτήσει την τοποθεσία του χρήστη έγκαιρα, μερίμνησα σε αυτές τις περιπτώσεις να εμφανίζεται μήνυμα λάθους το οποίο να προτρέπει τον χρήστη να επανεκκινήσει την εφαρμογή για να λυθεί το σχετικό πρόβλημα. Αν όμως η τοποθεσία έχει ανακτηθεί με επιτυχία, τότε σημαδεύουμε στο χάρτη μας τη τοποθεσία του χρήστη με το χαρακτηριστικό μαύρο location marker και έπειτα κεντράρουμε το χάρτη στο σημείο αυτό.



Φωτογραφία 4.1 Εύρεση τοποθεσία χρήστη και τοποθέτηση του location marker

4.1.2 Προβλήματα που αντιμετωπίστηκαν

Στην προσπάθεια μου για καλύτερη εμπειρία χρήσης, αποφάσισα να δημιουργήσω διάφορες ενδείξεις που να δείχνουν στο χρήστη ότι η εφαρμογή είναι σε κατάσταση αναμονής σε περιπτώσεις όπου είναι αναγκαίο κάποιο χρονικό διάστημα για ανάκτηση δεδομένων. Λόγω της ασύγχρονης μορφής που έχει το λογισμικό Android, δεν μπόρεσα να εμφανίσω τη συγκεκριμένη ένδειξη καθώς έτρεχε ασύγχρονα. Δηλαδή, η ένδειξη δούλευε κανονικά αλλά δεν εμφανιζόταν όπως έπρεπε (στην αρχή του κώδικα, στο τέλος του κώδικα). Ουσιαστικά, ξεκινούσε και τελειωνε αμέσως μετά το τέλος του κώδικα που ήταν ανάμεσα στις εντολές για εκκίνηση και τερματισμού της ένδειξης.

```

loadingDialog.startLoadingAnimation();
Intent intent=new Intent( packageContext: MapsActivity.this,SecondActivity.class);
if (todaysInformation.isEmpty()||todaysInformation.equals("")){
    try {
        findTodaysInformation();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
intent.putExtra( name: "information",todaysInformation);
loadingDialog.dismissLoadingAnimation();

```

Φωτογραφία 4.2 Παράδειγμα ενδεικτικής χρησιμοποίησης ένδειξης αναμονής

Για να λειτουργήσει αυτή η ένδειξη σωστά, έπρεπε να αλλάξω την δομή της υλοποίησης σε μεγάλο βαθμό. Δεν είχα πλέον το απαιτούμενο χρονικό διάστημα να προβώ σε αυτές τις αλλαγές, γι' αυτό και αποφάσισα να μην προχωρήσω στην αλλαγή της υλοποίησης μου.

```

public class LoadingDialog {

    private Context context;
    private Dialog dialog;

    LoadingDialog(Context myContext) { context = myContext; }

    void startLoadingAnimation() {

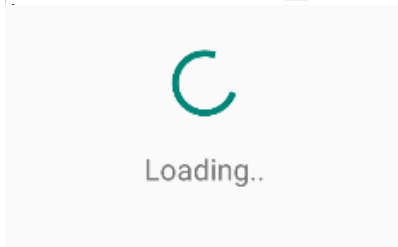
        dialog=new Dialog(context);
        dialog.setContentView(R.layout.custom_loading_dialog);
        dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
        dialog.setCancelable(false);
        dialog.show();

    }

    // dismiss method
    void dismissLoadingAnimation() { dialog.dismiss(); }

}

```



Σχήμα 4.3 Αριστερά η υλοποίησης της ένδειξης αναμονής, δεξιά η αναμενόμενη ένδειξη

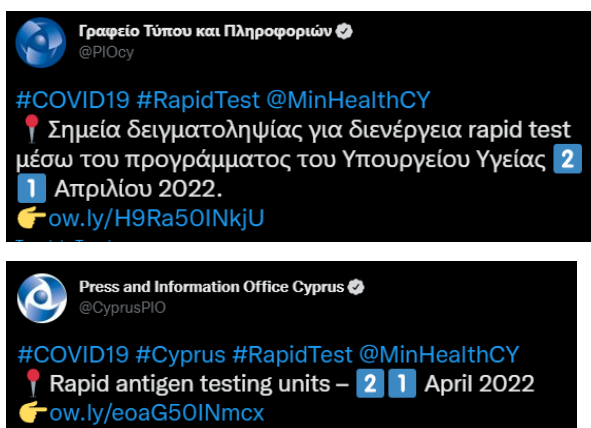
4.2 Εύρεση συγκεκριμένου tweet

Το επόμενο βήμα στην υλοποίηση ήταν να εξευρεθεί το συγκεκριμένο tweet, το οποίο θα χρησιμοποιήσω για να βρω το URL που με παραπέμπει στη ιστοσελίδα του Γραφείου Τύπου και Πληροφοριών, για να προχωρήσουμε στη συνέχεια στην ανάλυση των πληροφοριών για τα σημεία δειγματοληψίας. Για να βρούμε το tweet χρησιμοποιούμε το

Twitter API. Αρχικά αναλόγως της γλώσσας επιλογής της παρούσας στιγμής, χρησιμοποιούμε το Twitter API για να μας επιστρέψει το μοναδικό Twitter ID του Γραφείου Τύπου και Πληροφοριών (Αγγλικά-CyprusPIO, Ελληνικά-PIOcy) το οποίο χρειαζόμαστε για να ανακτήσουμε τα tweets. Μετά από την ανάκτηση του ID, το χρησιμοποιούμε καλώντας ξανά το Twitter API, αυτή τη φορά για να ανακτήσουμε τα tweets του συγκεκριμένου ID. Επειδή το Γραφείο Τύπου και Πληροφοριών καθημερινά δημοσιεύει tweet με τα σημεία δειγματοληψίας της επόμενης μέρας, και εγώ θέλω τις πληροφορίες για τα σημερινά, τότε όταν καλούμε το API χρησιμοποιώ και σαν παράμετρο την ημερομηνία (00:00:00-23:59:59) της προηγούμενης μέρας για να περιορίσω τις αχρείαστες δημοσιεύσεις. Στη συνέχεια, έχω όλα τα tweets της προηγούμενης μέρας και τα φιλτράρω βρίσκοντας το μοναδικό tweet που με ενδιαφέρει και εξάγω από αυτό τον URL σύνδεσμο για την ιστοσελίδα του Γραφείου Τύπου και Πληροφοριών.

4.2.1 Φιλτράρισμα

Για να βρούμε το συγκεκριμένο tweet που μας ενδιαφέρει, ανάλυσα και βρήκα ένα μοτίβο το οποίο χρησιμοποιεί καθημερινά το Γραφείο Τύπου και Πληροφοριών, με σκοπό να απομονώσω το tweet που με ενδιαφέρει. Συγκεκριμένα καθημερινά το tweet που αφορά τα σημεία δειγματοληψίας ξεκινά με την πρόταση «Σημεία δειγματοληψίας για διενέργεια rapid test μέσω του προγράμματος του Υπουργείου Υγείας» (“Rapid antigen testing units” στην Αγγλική γλώσσα) και ακολουθείται από την ημερομηνία. Επειδή έχω φιλτράρει ήδη την ημερομηνία, δεν υπάρχει λόγος να ελεγχτεί, επομένως το tweet το οποίο περιέχει αυτή τη πρόταση είναι και το συγκεκριμένο που αναζητώ.



Σχήμα 4.4 Παραδείγματα από tweet

4.2.2 Εύρεση URL από tweet

Το Twitter API μας επιστρέφει τα tweets σε μορφή Json, και εγώ τα εξάγω και τα τοποθετώ σε μια λίστα σε μορφή String για εύκολη περιήγηση και αναζήτηση. Επομένως, αναζητούμε από την λίστα το tweet που μας ενδιαφέρει με τον τρόπο που εξηγήσαμε στο σημείο 4.1.1, και στη συνέχεια αναζητούμε στο υπόλοιπο tweet το σημείο το οποίο ξεκινά με “https” το οποίο και υποδηλώνει ότι το υπόλοιπο του σημείου είναι ένας σύνδεσμος URL. Βρίσκω το σημείο, ανακτώ το σύνδεσμο URL και το επιστρέφω πίσω σε μορφή String.

```
Integer urlIndex=0;
if (language.equals("Greek")){
    for (int i=0; i<tweets.size();i++){
        if (tweets.get(i).contains("Σημεία Δειγματοληψίας για Διενέργεια")&&(tweets.get(i).contains("Rapid")
            ||tweets.get(i).contains("rapid"))&&(tweets.get(i).contains("test")||tweets.get(i).contains("Test")))) {
            urlIndex = tweets.get(i).indexOf("https");
            String url = tweets.get(i).substring(urlIndex);
            return url;
        }
    }
}
else if (language.equals("English")){
    for (int i=0; i<tweets.size();i++){
        if (tweets.get(i).contains("Rapid antigen testing units")||tweets.get(i).contains("Rapid Antigen Testing Units")&&(tweets.get(i).contains("Rapid")
            ||tweets.get(i).contains("rapid"))&&(tweets.get(i).contains("test")||tweets.get(i).contains("Test")))) {
            urlIndex = tweets.get(i).indexOf("https");
            String url = tweets.get(i).substring(urlIndex);
            return url;
        }
    }
}
```

Φωτογραφία 4.5 Κώδικας για εξεύρεση του συνδέσμου URL

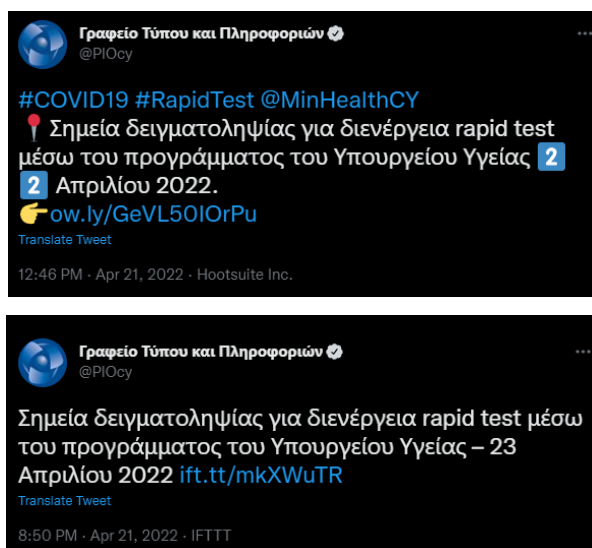
4.2.3 Προβλήματα που αντιμετωπίστηκαν

Τα URL κάποιες φορές σμικρύνονταν σε μορφές όπως “bit.ly” και “ow.ly”. Όταν δοκίμασα να συνδεθώ στο σύνδεσμο με την μικρή μορφή, σε κάποια σημεία στην ανάλυση της ιστοσελίδας είχα κάποια λάθη τα οποία δεν είχα ευθύνη, και αυτά επιλύονταν όταν χρησιμοποιούσα ολόκληρο τον σύνδεσμο στην κανονική του μορφή. Επομένως αποφάσισα ότι θα χρησιμοποιώ πάντα ολόκληρο τον σύνδεσμο για αποφυγή αυτών των προβλημάτων. Όμως, στη περίπτωση που σμικρύνονταν με την μορφή “ow.ly”, δεν μπορούσα να συνδεθώ πάνω στον σύνδεσμο για να προχωρήσω με την ανάλυση της ιστοσελίδας για ανάκτηση των πληροφοριών. Δοκίμασα διάφορα πράγματα, και ο τρόπος με τον οποίο επίλυσα αυτό το πρόβλημα ήταν η αντικατάσταση του “https:ow.ly...” με “http:ow.ly...”.

```
String fullUrl=getFullURL(url);
if (fullUrl.contains("ow.ly")){
    fullUrl=fullUrl.replace( target: "https", replacement: "http");
    fullUrl=getFullURL(fullUrl);
}
```

Φωτογραφία 4.6 Ανάκτηση ολόκληρου συνδέσμου και η περίπτωση όπου υπήρχε η μορφή “ow.ly”

Ακόμα ένα πρόβλημα που αντιμετώπισα ήταν η περίπτωση στην οποία το Γραφείο Τύπου και Πληροφοριών ανακοινώσει την ίδια μέρα τα σημεία δειγματοληψίας για δύο ημέρες. Αυτό μπορεί να γίνει σε περίπτωση αργιών ή από λάθος. Προσπάθησα να ελέγξω αυτό το πρόβλημα με το να συμπεριλάβω στο φιλτράρισμα των tweets και την ημερομηνία, αλλά δεν ήταν σε όλες τις περιπτώσεις εφικτό καθώς σε κάποιες δημοσιεύσεις χρησιμοποιούνται emojis στη θέση της ημερομηνίας, τα οποία δεν μεταφράζονται σωστά και δεν μπορώ επομένως να ελέγξω την ημερομηνία. Επιπρόσθετα, αυτό μπορεί να οδηγήσει στην εισχώρηση στη βάση τα σημεία δειγματοληψίας της επόμενης μέρας και όχι της σημερινής, αφού μπορεί η πρώτη δημοσίευση την οποία επιλέγουμε ως την επιθυμητή να περιέχει τα σημεία δειγματοληψίας της επόμενης μέρας. Αυτό συνεπάγεται σε διάφορα σφάλματα που χρησιμοποιείται η βάση δεδομένων, όπως η ανάκτηση δεδομένων αφού θα γίνεται αναζήτηση με παράμετρο την σημερινή ημερομηνία, όπως και επίσης σφάλματα στην παρουσίαση των σημείων δειγματοληψίας καθώς υπάρχει η περίπτωση να μην είναι τα ίδια, και αυτό να οδηγήσει το χρήστη σε λανθασμένη επιλογή.



Φωτογραφία 4.7 Παράδειγμα του ποιο πάνω σφάλματος

Για αποφυγή αυτών των σφαλμάτων θα ήταν ιδιαίτερα σημαντικό να τηρηθεί κάποια συνέπεια από το Γραφείο Τύπου και Πληροφοριών, ως προς την δομή των δημοσιεύσεων για καλύτερη διαχείριση τους από την υλοποίησή μου, αλλά και ως προς την μέρα ανακοίνωσης των σημείων δειγματοληψίας.

4.3 Εύρεση πληροφοριών από URL

Το πιο σημαντικό κομμάτι της υλοποίησης ήταν η εξεύρεση των πληροφοριών από την ιστοσελίδα του Γραφείου Τύπου και Πληροφοριών. Εδώ, ήταν το σημείο όπου ήθελα να χρησιμοποιήσω προγραμματισμό σε πραγματικό χρόνο όσο το περισσότερο για μια αυτοματοποιημένη διαδικασία. Δυστυχώς αυτό δεν ήταν εφικτό σε όλα τα μέρη της εξαγωγής των πληροφοριών που οφείλεται κυρίως στον ανθρώπινο παράγοντα. Στόχος ήταν η εξαγωγή όλων των πληροφοριών που υπάρχουν στην ιστοσελίδα (σημεία δειγματοληψίας, πληροφορίες σχετικά με δικαιούχους). Τα σημεία δειγματοληψίας είναι συγκεντρωμένα σε μορφή table, χωρισμένα ανά επαρχία. Οι πληροφορίες σχετικά με δικαιούχους βρίσκονται πάνω από τα σημεία δειγματοληψίας.

4.3.1 HTML Parsing

Με τη χρήση της μεθοδολογίας HTML parsing, ήταν εφικτό να αναλύσω και να βρω τα HTML elements τα οποία περιείχαν τις πληροφορίες που αναζητούσα. Για να βρω τις πληροφορίες σχετικά με τους δικαιούχους, ανέλυσα την ιστοσελίδα και βρήκα ότι οι πληροφορίες υπάρχουν στο HTML element με όνομα “press_release_main_column”. Επομένως, ανάκτησα όλα τα στοιχεία του συγκεκριμένου element και τα χρησιμοποίησα με σκοπό να κτίσω ένα κείμενο το οποίο περιείχε όλες τις πληροφορίες που δίνονταν.

```

String information = "";
Element firstDiv = document.select( cssQuery: "div.press_release_main_column").first();
try {
    Element informationDiv = firstDiv.getElementsByTagName( tagName: "div").get(9);
    for (int i = 0; i < informationDiv.childrenSize() - 3; i++) {
        Element element = informationDiv.child(i);
        if (!element.select( cssQuery: "ul").isEmpty()) {
            Elements item = element.select( cssQuery: "ul").first().children();
            for (Element e : item) {
                information = information + "\u2022" + e.text() + "\n";
            }
        } else if (!element.select( cssQuery: "ol").isEmpty()) {
            Elements item = element.select( cssQuery: "ol").first().children();
            int j = 1;
            for (Element e : item) {
                information = information + "\t" + j + ". " + e.text() + "\n";
                j++;
            }
        } else {
            information = information + element.text() + "\n";
        }
    }
}

```

Φωτογραφία 4.8 Εύρεση και εξαγωγή πληροφοριών για δικαιούχους

Στη συνέχεια, για να ανακτήσω όλες τις πληροφορίες σχετικά με τα σημεία δειγματοληψίας έπρεπε να βρω το HTML element όπου υπάρχει το table, και στη συνέχεια να περιηγηθώ στο table εξάγοντας τις πληροφορίες που χρειάζομαι.

Αρχικά, παρατήρησα ότι η πρώτη γραμμή έπρεπε να αγνοηθεί καθώς ήταν οι τίτλοι των στηλών που ακολουθούσαν. Συνεχίζοντας, παρατήρησα ότι η γραμμή στην οποία άρχιζε μια επαρχία, αποτελείτο από 4 στήλες και όχι από 3, επομένως κάθε φορά που η γραμμή είχε 4 στήλες, τότε μια νέα επαρχία ξεκινούσε. Όλες οι υπόλοιπες γραμμές μέχρι να ξανά βρούμε γραμμή με 4 στήλες, ήταν σημεία δειγματοληψίας της συγκεκριμένης επαρχίας. Κράτησα αυτό το μοτίβο μέχρι το τέλος, και δημιούργησα για κάθε επαρχία ένα αντικείμενο, το οποίο περιείχε το όνομα της επαρχίας και μια λίστα από αντικείμενο τύπου σημείου δειγματοληψίας. Τα αντικείμενα τύπου σημείο δειγματοληψίας αποτελούνταν από το όνομα του σημείου, επιπλέον πληροφορίες, ώρες λειτουργίας, τηλέφωνο, τοποθεσία (που χρησιμοποιείται αργότερα), απόσταση(που χρησιμοποιείται αργότερα), διάρκεια (που χρησιμοποιείται αργότερα), και τέλος μια λίστα από τα δρομολόγια (που χρησιμοποιείται αργότερα).

```

for (Element table : document.select( cssQuery: "table")) {
    Elements rows = table.select( cssQuery: "tr");
    int i=1;
    while (i<rows.size()){
        Elements tds = rows.get(i).select( cssQuery: "td");
        TestingUnits testingUnits1=new TestingUnits();
        int j=0;
        if (tds.size()==4) {
            String telephone=tds.get(3).text();
            String district=tds.get(0).text();
            district=district.substring(0,district.indexOf("(")-1);
            testingUnits1.setDistrict(district);
            Locations locTemp2=new Locations();
            int start=0,end=0;
            if (tds.get(1).text().indexOf("(")!=-1){
                start=tds.get(1).text().indexOf("(");
            }
            if (tds.get(1).text().indexOf(")")!=-1){
                end=tds.get(1).text().indexOf(")");
            }
            String additionalInfo="";
            if (start!=0 && end!=0){
                additionalInfo=tds.get(1).text().substring(start,end);
                String name=tds.get(1).text().replace(additionalInfo, replacement: "");
                additionalInfo=additionalInfo.replace( target: "(", replacement: "");
                additionalInfo=additionalInfo.replace( target: ")", replacement: "");
                locTemp2.setAdditionalInfo(additionalInfo);
                name=name.replace( target: "(", replacement: "");
                name=name.replace( target: ")", replacement: "");
                locTemp2.setName(name);
            }else{
                locTemp2.setName(tds.get(1).text());
            }
            locTemp2.setTelephone(telephone);
            locTemp2.setOperatingHours(tds.get(2).text());
            testingUnits1.addLocation(locTemp2);

            j = j + 1;
            if (j!=rows.size()) {
                boolean flag = true;
            }
        }
        i=j;
    }
}

boolean flag = true;
ArrayList<Location> locations = new ArrayList<>();

while (flag&&j!=rows.size()) {
    Elements tds2 = rows.get(j).select( cssQuery: "td");
    if (tds2.size() == 4) {
        flag = false;
    } else {
        Locations locTemp = new Locations();
        int start2=0,end2=0;
        if (tds2.get(0).text().indexOf("(")!=-1){
            start2=tds2.get(0).text().indexOf("(");
        }
        if (tds2.get(0).text().indexOf(")")!=-1) {
            end2 = tds2.get(0).text().indexOf(")"+1;
        }
        String additionalInfo2="";
        if (start2!=0 && end2!=0){
            additionalInfo2=tds2.get(0).text().substring(start2,end2);
            String name=tds2.get(0).text().replace(additionalInfo2, replacement: "");
            additionalInfo2=additionalInfo2.replace( target: "(", replacement: "");
            additionalInfo2=additionalInfo2.replace( target: ")", replacement: "");
            locTemp.setAdditionalInfo(additionalInfo2);
            name=name.replace( target: "(", replacement: "");
            name=name.replace( target: ")", replacement: "");
            locTemp.setName(name);
        }else{
            locTemp.setName(tds2.get(0).text());
        }
        String telephone2=tds2.get(2).text();
        locTemp.setTelephone(telephone2);
        locTemp.setOperatingHours(tds2.get(1).text());
        testingUnits1.addLocation(locTemp);
        j++;
    }
}
testingUnits.add(testingUnits1);
}
}

```

Σχήμα 4.9 Εξέυρεση και εξαγωγή πληροφοριών για σημεία δειγματοληψίας

4.3.2 Προβλήματα που αντιμετώπιστηκαν

Ο σκοπός στην υλοποίηση της εφαρμογής ήταν η πλήρης χρησιμοποίηση προγραμματισμού σε πραγματικό χρόνο για πλήρης αυτοματοποιημένη διαδικασία. Δυστυχώς στη περίπτωση μου, αυτό δεν ήταν ρεαλιστικά εφικτό και στηρίζεται στον ανθρώπινο παράγοντα.

Για παράδειγμα, ο εργαζόμενος του Γραφείου Τύπου και Πληροφοριών, σε περίπτωση που κάνει ένα λάθος και προσθέσει μια περιττή γραμμή, αλλάζει σε κάποιες περιπτώσεις η δομή της ιστοσελίδας και επομένως των HTML elements. Συγκεκριμένα, στην περίπτωση όπου εξάγω της πληροφορίες για τους δικαιούχους, δεν είναι σταθερό καθώς αν προστεθεί μια γραμμή, μπορεί στην υλοποίηση μου να αγνοηθεί ένα κομμάτι από της πληροφορίες. Επίσης, ένα άλλο σενάριο που έτυχε, ήταν η πρόσθεση μιας κενής γραμμής στο τέλος του table που αφορά τα σημεία δειγματοληψίας. Αυτό είχε ως αποτέλεσμα το κλείσιμο της εφαρμογής καθώς με την υλοποίηση μας, ελέγχουμε και τη συγκεκριμένη γραμμή για στοιχεία τα οποία δεν υπάρχουν και αυτό οδηγεί σε ένα Null pointer exception.

Σε αυτές τις περιπτώσεις, η αντιμετώπιση όλων αυτών των πιθανοτήτων είναι απίθανη. Υπάρχουν πάρα πολλά σενάρια τα οποία δεν είναι εφικτό να καλυφθούν όλα. Επομένως, για να μην υπάρχουν τέτοιου είδους προβλήματα θα πρέπει να κτιστεί ένα σταθερό template το οποίο κάθε εργαζόμενος θα χρησιμοποιεί, για την σωστή λειτουργία της υλοποίησης μου.

4.4 Εύρεση τοποθεσιών

Το επόμενο κομμάτι της υλοποίησης μου είναι η εύρεση των τοποθεσιών των σημείων δειγματοληψίας χρησιμοποιήσα το Geocoding API της Google. Με το Geocoding API μπορούμε να μετατρέψουμε μια διεύθυνση σε συντεταγμένες τις οποίες θα χρησιμοποιήσουμε για να αναπαραστήσουμε το σημείο δειγματοληψίας στο χάρτη μας. Αναπαριστούμε το σημείο δειγματοληψίας στο χάρτη μας με ένα απλό location marker με τις πληροφορίες που ανακτήσαμε από προηγούμενα βήματα, έτσι ώστε να δώσουμε τη δυνατότητα στο χρήστη να κάνει την καλύτερη επιλογή.

4.4.1 Εύρεση τοποθεσιών χρησιμοποιώντας Google API's

Για την εύρεση των τοποθεσιών χρησιμοποιούμε το Geocoding API της Google. Έχουμε πλέον όλα τα σημεία δειγματοληψίας μας σε μια λίστα, στην οποία θα περιηγηθούμε και θα χρησιμοποιήσουμε το όνομα του σημείο μαζί με την επαρχία που βρίσκεται (για εξάλειψη πιθανών λαθών με ίδιες διευθύνσεις σε διαφορετικές επαρχίες). Λόγω των διαφορετικών ονομασιών του Γραφείου Τύπου και Πληροφοριών και την βάση δεδομένων της Google, δεν είναι εφικτό να επιστραφούν όλες οι συντεταγμένες για όλα τα σημεία χωρίς να υπάρξουν λάθη, και σε κάποιες περιπτώσεις δεν υπάρχει ακρίβεια. Στις περιπτώσεις όπου δεν υπάρχει ακρίβεια, η απάντηση από το API συμπεριλαμβάνει το location_type το οποίο υποδεικνύει με πόση ακρίβεια μας επέστρεψε πίσω συντεταγμένες. Το location_type μπορεί να πάρει τις πιο κάτω τιμές:

- ROOFTOP: υποδεικνύει ότι το αποτέλεσμα είναι μια ακριβής τοποθεσία.
- RANGE_INTERPOLATED: υποδεικνύει ότι το αποτέλεσμα είναι μια ανακριβής τοποθεσία, συνήθως σε δρόμο, με παρεμβολή μεταξύ δύο σημείων.
- GEOMETRIC_CENTER: υποδεικνύει ότι το αποτέλεσμα είναι ένα γεωμετρικό κέντρο μιας αναζήτησης όπως ενός δρόμου, ή μιας περιοχής.

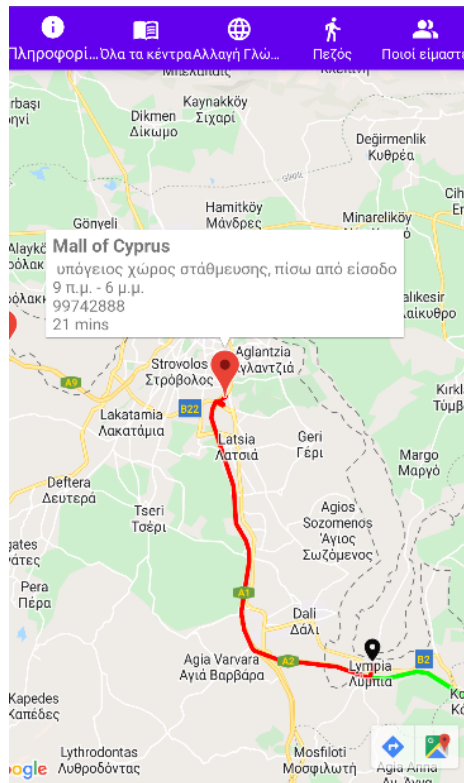
- APPROXIMATE υποδεικνύει ότι το αποτέλεσμα είναι μια ανακριβής τοποθεσία.

Επομένως, για αποφυγή λαθών, αποφάσισα ότι απαντήσεις από το Geocode API οι οποίες περιλαμβάνουν το location_type με τιμές όπως RANGE INTERPOLATED και GEOMETRIC CENTER, APPROXIMATE θα αγνοούνται για αποφυγή λαθών και ανακριβών τοποθεσιών που μπορεί να οδηγήσουν τον χρήστη σε λανθασμένη επιλογή.

```
Response response = client.newCall(request2).execute();
String answer=response.body().toString();
JSONObject json_data = new JSONObject(answer);
client.dispatcher().executorService().shutdown();
JSONArray results = json_data.getJSONArray( name: "results");
for(int i=0; i < results.length(); i++)
{
    JSONObject object = results.getJSONObject(i);
    JSONObject geometry= object.getJSONObject("geometry");
    String location_type=null;
    try {
        location_type = geometry.getString( name: "location_type");
    }catch (Exception e){
        e.printStackTrace();
    }
    if(location_type!=null && !location_type.toString().equals("APPROXIMATE") && !location_type.toString().equals("RANGE_INTERPOLATED")
        && !location_type.toString().equals("GEOMETRIC_CENTER")) {
        JSONObject loc = geometry.getJSONObject("location");
        Double Lat = loc.getDouble( name: "lat");
        Double Lng = loc.getDouble( name: "lng");
        LatLng point = new LatLng(Lat, Lng);
        return point;
    }
}
```

Φωτογραφία 4.10 Φιλτράρισμα απάντησης του Geocoding API

Όταν τελειώσω από όλα τα σημεία δειγματοληψίας, τους αναθέτω στο αντικείμενο τους την τιμή των συντεταγμένων που μου επέστρεψε το Geocoding API και στη συνέχεια τα τοποθετώ στο χάρτη μαζί με τις πληροφορίες τους.



Φωτογραφία 4.11 Ανάθεση location marker με τις συντεταγμένες και παρουσίαση πληροφοριών του σημείου δειγματοληψίας

4.4.2 Προβλήματα που αντιμετωπίστηκαν

Το κυριότερο πρόβλημα που αντιμετωπίζει η υλοποίηση μου είναι ότι με το ποιο πάνω φιλτράρισμα, δεν είναι δυνατόν να παρουσιαστούν πολλά από τα σημεία δειγματοληψίας στο χάρτη. Περιορίζουμε για λόγους αποφυγής λαθών και λανθασμένης επιλογής που μπορεί να οδηγήσει τον χρήστη σε κάποια άκυρη τοποθεσία, και αυτό μας αφήνει με πολύ λίγα σημεία δειγματοληψίας τα οποία φαίνονται στο χάρτη. Δεν είναι ένα πρόβλημα το οποίο δεν έχει λύση, καθώς για κάποια σημεία δειγματοληψίας είναι απλά διαφορετική η ονομασία. Για παράδειγμα, το Γραφείο Τύπου και Πληροφοριών μας δίνει το σημείο δειγματοληψίας «Κοινοτικό Ιατρείο Ιδαίου» το οποίο όταν σταλεί στο Geocoding API δεν μας δίνει κάποια απάντηση επειδή είναι καταχωρημένο με διαφορετική ονομασία, ή υπάρχει μόνο καταχωρημένο στην Αγγλική γλώσσα.

Για επίλυση αυτού του προβλήματος θα μπορούσαν οι εργαζόμενοι του Γραφείου Τύπου και Πληροφοριών να χρησιμοποιούν ονομασίες οι οποίες υπάρχουν ήδη στη

βάση δεδομένων της Google, ή να υπάρξει κάποιου είδους συνεργασία έτσι ώστε να γίνει αλλαγή των ονομάτων των τοποθεσιών στη βάση της Google με τα ονόματα των σημείων δειγματοληψίας που παρουσιάζει το Γραφείο Τύπου και Πληροφοριών.

4.5 Εύρεση κοντινότερου σημείου δειγματοληψίας

Για το τελευταίο κομμάτι της υλοποίησης μου περιλαμβάνει τη παρουσίαση δρομολογίων και εύρεση του κοντινότερου σημείου δειγματοληψίας χρησιμοποιώ το Directions API της Google. Το Directions API μας επιτρέπει να βρούμε δρομολόγια μεταξύ δύο συντεταγμένων, της τοποθεσίας του χρήστη με την τοποθεσία του σημείου δειγματοληψίας. Επομένως, χρησιμοποιώ το Directions API στέλνοντας την τοποθεσία του χρήστη και την τοποθεσία όλων των σημείων δειγματοληψίας μαζί με το μέσο μεταφοράς το οποίο επέλεξε ο χρήστης, μου επιστρέφει πίσω δρομολόγια από τη τοποθεσία του χρήστη, προς το κάθε σημείο δειγματοληψίας.

Η αρχική ιδέα για εύρεση του κοντινότερου σημείου δειγματοληψίας ήταν η χρήση της ευκλείδειας απόστασης μεταξύ των δύο συντεταγμένων, αλλά στη συνέχεια σκεπτόμενος ότι αυτό ίσως να μην είναι πρακτικό εξαιτίας των δρόμων που μπορεί ο χρήστης να χρησιμοποιήσει αποφάσισα ότι η καλύτερη μονάδα μέτρησης για την εύρεση του κοντινότερου σημείου δειγματοληψίας είναι η διάρκεια του ταξιδιού με τον τρόπο μεταφοράς που επέλεξε ο χρήστης. Τη διάρκεια μεταφοράς (duration) μας την επιστρέφει το Directions API μαζί με τα δρομολόγια, έτσι για κάθε σημείο δειγματοληψίας έχουμε μια μονάδα που υποδεικνύει την διάρκεια του ταξιδιού η οποία μετατρέπεται σε λεπτά, και μετά συγκρίνεται με όλα τα κέντρα δειγματοληψίας για να βρω το ποιο κοντινό στο χρήστη (δηλαδή τη μικρότερη τιμή διάρκειας σε λεπτά).

4.5.1 Εύρεση δρομολογίων

Για την εύρεση των δρομολογίων χρησιμοποιείται η διαδικασία που εξηγήθηκε πιο πάνω. Στέλνουμε στο Directions API την τοποθεσία του χρήστη σε συνδυασμό με τον τοποθεσία του σημείου δειγματοληψίας μαζί με το μέσο μεταφοράς που είναι επιλεγμένο, και μας επιστρέφει τα δρομολόγια σε μορφή κειμένου το οποίο είναι encoded. Αναθέτω αυτή τη τιμή για το κάθε σημείο δειγματοληψίας. Ουσιαστικά, για την δημιουργία ενός

δρομολογίου χρειάζομαι μια λίστα από συντεταγμένες, την οποία θα χρησιμοποιήσω για να δημιουργήσω μια πολυγραμμή (polyline) η οποία θα υποδεικνύει το δρομολόγιο. Λόγω του ότι το δρομολόγιο επιστρέφεται σε encoded μορφή κειμένου, πρέπει να γίνει μετατροπή του σε λίστα από σημεία. Αυτή η μετατροπή γίνεται με τη χρήση μιας βοηθητικής συνάρτησης και δημιουργώ μια νέα πολυγραμμή κάθε φορά που ο χρήστης πατάει σε κάποιο σημείο δειγματοληψίας, το οποίο του υποδεικνύει αμέσως ποιο είναι το δρομολόγιο προς το συγκεκριμένο σημείο δειγματοληψίας.

```
String markerTitle = marker.getTitle();
outer:
for (int i = 0; i < testingUnits.size(); i++) {
    for (int j = 0; j < testingUnits.get(i).getLocations().size(); j++) {
        if (testingUnits.get(i).getLocations().get(j).getName().equals(markerTitle)) {
            for (int k = 0; k < testingUnits.get(i).getLocations().get(j).getPolyLines().size(); k++) {
                List<LatLng> list = decodePoly(testingUnits.get(i).getLocations().get(j).getPolyLines().get(k));
                CenterRoute.add(mMap.addPolyline(new PolylineOptions()
                    .clickable(true)
                    .addAll(list)
                    .color(Color.RED)
                ));
            }
            break outer;
        }
    }
}
return false;

private List<LatLng> decodePoly(String encoded) {
    List<LatLng> poly = new ArrayList<LatLng>();
    int index = 0, len = encoded.length();
    int lat = 0, lng = 0;

    while (index < len) {
        int b, shift = 0, result = 0;
        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lat += dlat;

        shift = 0;
        result = 0;
        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lng += dlng;

        LatLng p = new LatLng((((double) lat / 1E5)),
            (((double) lng / 1E5)));
        poly.add(p);
    }

    return poly;
}
```

Σχήμα 4.12 Στα αριστερά η διαδικασία τοποθέτησης του δρομολογίου στο χάρτη, στα δεξιά η διαδικασία decode του encoded δρομολογίου σε λίστα από συντεταγμένες

4.5.2 Διαδικασία εύρεσης κοντινότερου σημείου δειγματοληψίας

Για την εύρεση του κοντινότερου σημείου δειγματοληψίας μετατρέπω τη διάρκεια του δρομολογίου, την οποία ανακτώ από το Directions API σε λεπτά (είναι της μορφής «17 mins») και στη συνέχεια συγκρίνω όλα τα σημεία μεταξύ τους για να βρω το σημείο με την μικρότερη τιμή στη διάρκεια το οποίο σηματοδοτεί το κοντινότερο σημείο δειγματοληψίας.

Υπάρχουν κάποιες περιπτώσεις όταν το μέσο μεταφοράς είναι τα πόδια, όπου η διάρκεια επιστρέφεται και είναι της μορφής («1 day 3 minutes»). Αυτή τη μορφή την μετατρέπω σε λεπτά($1*24*60+3$) και την αναθέτω στο σημείο δειγματοληψίας της. Έπειτα ακολουθώ την ίδια διαδικασία βρίσκοντας το μικρότερο αριθμό διάρκειας το οποίο σηματοδοτεί το κοντινότερο σημείο δειγματοληψίας.

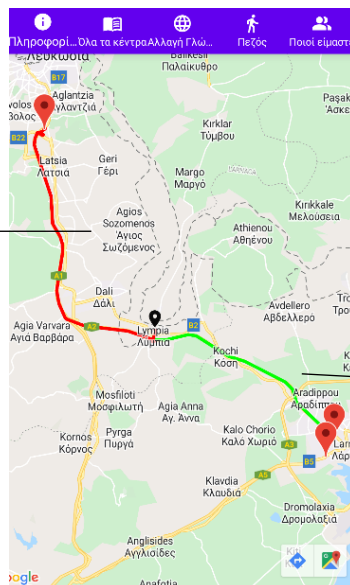
```
int min=Integer.MAX_VALUE;
Locations minHelper=null;
for (int i=0;i<testingUnits.size();i++){
    for (int j=0; j<testingUnits.get(i).getLocations().size(); j++){
        String duration=testingUnits.get(i).getLocations().get(j).getDuration();
        int hours=0,minutes=0,days=0;
        if (duration==null){
            continue;
        }
        if (duration.contains("day")||duration.contains("days")){
            days=Integer.parseInt(duration.split( regex: " ")[0]);
            hours=Integer.parseInt(duration.split( regex: " ")[2]);
        }
        else if (duration.contains("hours")||duration.contains("hour")){
            hours=Integer.parseInt(duration.split( regex: " ")[0]);
            minutes=Integer.parseInt(duration.split( regex: " ")[2]);
        }else{
            minutes=Integer.parseInt(duration.split( regex: " ")[0]);
        }
        int duration_value=days*24*60+hours*60+minutes*1;
        if (min>duration_value){
            min=duration_value;
            minHelper=testingUnits.get(i).getLocations().get(j);
        }
    }
}
return minHelper;
```

Φωτογραφία 4.13 Διαδικασία μετατροπής διάρκειας και εύρεση κοντινότερου σημείου δειγματοληψίας

4.5.3 Διαχώριση κοντινότερου σημείου δειγματοληψίας

Για τη διαχώριση του κοντινότερου σημείου δειγματοληψίας από όλα τα υπόλοιπα σημεία δειγματοληψίας, απλά αναθέσαμε το χρώμα πράσινο για την παρουσίαση του δρομολογίου του, το οποίο δεν φεύγει ποτέ από τον χάρτη.

Διαφορετικό σημείο δειγματοληψίας ←



→ Κοντινότερο σημείο δειγματοληψίας

Φωτογραφία 4.14 Διαφορά κοντινότερου σημείου δειγματοληψίας με τα υπόλοιπα

4.6 Αποθήκευση πληροφοριών σε βάση δεδομένων

Για την αποφυγή συνεχών και συστηματικών κλήσεων των διάφορων APIs τα οποία χρησιμοποιούνται για την υλοποίηση αυτής της εφαρμογής, αποφάσισα να δημιουργήσω μια βάση δεδομένων με σκοπό να καλώ τα απαραίτητα APIs μόνο μια φορά για κάθε λειτουργία που χρειάζονταν. Τα σημεία δειγματοληψίας αποθηκεύονται στη βάση δεδομένων μετά από όλες τις πιο πάνω διαδικασίες.

Η συνεχής κλήση των APIs μπορεί να επιφέρει τόσο επιβάρυνση οικονομική στο πάροχο της εφαρμογής, τόσο και αποδοτική επιβάρυνση καθώς η κλήση των APIs είναι χρονοβόρα και μπορεί να επιφέρει διάφορα προβλήματα που σχετίζονται με την απόδοση της εφαρμογής. Ακόμα, η ανάκτηση δεδομένων από την βάση δεδομένων γίνεται πολύ γρήγορα, και αυτό θα παρέχει στο χρήστη καλύτερη εμπειρία με την εφαρμογή. Τέλος, χρησιμοποιώ τη ανάκτηση δεδομένων από τη βάση δεδομένων στη λειτουργία παρουσίασης όλων των σημείων δειγματοληψίας, αφού δεν είναι εύκολη η μεταφορά τόσο μεγάλου αριθμού πληροφοριών από μια οθόνη στη άλλη.

Επέλεξα την SQLite για αυτή τη διαδικασία, καθώς η υλοποίηση της ήταν ιδιαίτερα βοηθητική και εύκολη (δημιουργία τοπικού αρχείου στη συσκευή του χρήστη), αλλά ο πιο σημαντικός λόγος ήταν ότι δεν είχα κάποιο διακομιστή να χρησιμοποιήσω. Στη

συνέχεια, λόγω του ότι η βάση δεδομένων μπορεί να μεγαλώσει αρκετά μετά από κάποιο χρονικό διάστημα, αποφάσισα όπως τα σημεία δειγματοληψίας που υπάρχουν στη βάση και είναι μεγαλύτερα σε ηλικία των δύο ημερών, θα διαγράφονται για να μην γίνεται σπατάλη αποθηκευτικού χώρου στη συσκευή του χρήστη.

4.6.1 Δομή βάσης δεδομένων

Η βάση δεδομένων μου αποτελείται από μόνο ένα πίνακα, στο οποίο υπάρχουν τα παρακάτω πεδία.

1. Date: υποδεικνύει την ημερομηνία του σημείου δειγματοληψίας
2. District: υποδεικνύει την επαρχία του σημείου δειγματοληψίας
3. Name: υποδεικνύει το όνομα του σημείου δειγματοληψίας
4. Additional_info: υποδεικνύει τις επιπλέον πληροφορίες του σημείου δειγματοληψίας
5. Operating_hours: υποδεικνύει τις ώρες λειτουργίας του σημείου δειγματοληψίας
6. Lat: υποδεικνύει το πρώτο σημείων των συντεταγμένων του σημείου δειγματοληψίας
7. Lon: υποδεικνύει το δεύτερο σημείων των συντεταγμένων του σημείου δειγματοληψίας
8. Telephone: υποδεικνύει το τηλέφωνο επικοινωνίας του σημείου δειγματοληψίας
9. Language: υποδεικνύει τη γλώσσα που είναι καταχωρημένο το σημείο δειγματοληψίας
10. Duration: υποδεικνύει τη διάρκεια ταξιδιού προς το σημείο δειγματοληψίας
11. Mode: υποδεικνύει τη μέθοδο μεταφοράς του χρήστη προς το σημείο δειγματοληψίας

date ++	district	name	filter	additional_info	operating_hours	lat	lon	telephone	language	duration	mode
filter	filter		filter		filter	filter		filter	filter	filter	filter
1	21-04-2022	Λευκωσία	Nicosia Mall	filter	9 π.μ. - 6 μ.μ.			70001866	Greek	filter	driving
2	21-04-2022	Λευκωσία	Ιερός Ναός Αγίου Γεωργίου, Ακρόα	filter	9 π.μ. - 6 μ.μ.			94050785	Greek	filter	driving
3	21-04-2022	Λευκωσία	Κανατέα Σαρτζι Βολίου	filter	9 π.μ. - 6 μ.μ.			99969931	Greek	filter	driving
4	21-04-2022	Λευκωσία	Πολικό Κανατέα Σαρτζι Αιθούσιος	filter	9 π.μ. - 6 μ.μ.			99154344	Greek	filter	driving
5	21-04-2022	Λευκωσία	Αθυσία Ιεροί Ναοί Αγίου Ιωαννίου ...	filter	9 π.μ. - 6 μ.μ.			96959317	Greek	filter	driving
6	21-04-2022	Λευκωσία	Ιερός Ναός Αγίου Δημητρίου, Αρσινόη	filter	9 π.μ. - 6 μ.μ.			96022766	Greek	filter	driving
7	21-04-2022	Λευκωσία	Εκκλησία Αγίου Σπυρίδων	filter	9 π.μ. - 6 μ.μ.			94041843	Greek	filter	driving
8	21-04-2022	Λευκωσία	Εκκλησία Αγίου Πολυδώρου, Προκόπια, Κορυμ	filter	9 π.μ. - 6 μ.μ.			96812424	Greek	filter	driving
9	21-04-2022	Λευκωσία	Προκόπια Αρρετήστρου Ακακίμης, πίσω από ...	filter	9 π.μ. - 6 μ.μ.			99969920	Greek	filter	driving
10	21-04-2022	Λευκωσία	Mall of Cyprus	filter	9 π.μ. - 6 μ.μ.	35.136263	33.370967	99742888	Greek	21 mins	driving
11	21-04-2022	Λευκωσία	Μέγας Αγίου Δημητρίου	filter	9 π.μ. - 6 μ.μ.			99395018	Greek	filter	driving
12	21-04-2022	Λευκωσία	Εμπορικό και Ψυχαγωγικό Κέντρο New Plaza, ...	filter	2 π.μ. - 9 μ.μ.	35.1305135	33.2363158	96959317	Greek	31 mins	driving
13	21-04-2022	Λευκωσία	Εμπορικό Πασατζήσιο Κέντρο	filter	9 π.μ. - 6 μ.μ.			96625766	Greek	filter	driving
14	21-04-2022	Λευκωσία	Κέντρο Υγείας Αιθούσιος	filter	9 π.μ. - 6 μ.μ.			96659317	Greek	filter	driving
15	21-04-2022	Λευκωσία	Κανατέα Σαρτζι Πολυκαταστήμα	filter	9 π.μ. - 5 μ.μ.			77774400	Greek	filter	driving
16	21-04-2022	Λευκωσία	Κανατέα Σαρτζι Πολυκαταστήμα Τ.Α.Α.Α.Α.Α.	filter	1:30 π.μ. - 4 μ.μ.			77774400	Greek	filter	driving
17	21-04-2022	Λευκωσία	Ιερός Ναός Αγίου Κωνσταντίνου Αποστολίου, Κέντρο	filter	12 π.μ. - 2 μ.μ.			77774400	Greek	filter	driving
18	21-04-2022	Λευκωσία	Εμπορικό Κέντρο κέντρο Ορφέως, Πολυκαταστήμα, ...	filter	9 π.μ. - 6 μ.μ.	34.681296	32.836988	99790687	Greek	31 mins	driving
19	21-04-2022	Λευκωσία	Αθυσία Πολυκαταστήμα Ενδύσεων Πίσω ...	filter	9 π.μ. - 6 μ.μ.	34.7046059	32.8916242	99790687	Greek	32 mins	driving
20	21-04-2022	Λευκωσία	My Mall Limassol	filter	9 π.μ. - 6 μ.μ.			94041843	Greek	filter	driving
21	21-04-2022	Λευκωσία	Εμπορικό Κέντρο Κινηματο	filter	9 π.μ. - 6 μ.μ.			77774400	Greek	filter	driving
22	21-04-2022	Λευκωσία	Stadion Atlantica Garden, Γεωμετρικό	filter	9 π.μ. - 6 μ.μ.			94041843	Greek	filter	driving
23	21-04-2022	Λευκωσία	Πασατζήσιο ΤΕΤΑ, κέντρο Τίσεως Παπαδόπουλος	filter	9 π.μ. - 6 μ.μ.	34.6730861	32.8455063	99942219	Greek	52 mins	driving
24	21-04-2022	Λευκωσία	Μεταρρύθμιση Mall	filter	9 π.μ. - 6 μ.μ.	34.9244735	32.80943	94041843	Greek	17 mins	driving
25	21-04-2022	Λευκωσία	Κανατέα Σαρτζι Πολυκαταστήμα Μουσείο, ...	filter	9 π.μ. - 6 μ.μ.	34.9123268817392	33.5988346961567	99146623	Greek	18 mins	driving
26	21-04-2022	Λευκωσία	Νέος Σαρτζι Πολυκαταστήμα Τράπεζα	filter	9 π.μ. - 6 μ.μ.			94041843	Greek	filter	driving
27	21-04-2022	Λευκωσία	Κανατέα Σαρτζι Αθυσία	filter	9 π.μ. - 6 μ.μ.			94050785	Greek	filter	driving
28	21-04-2022	Λευκωσία	Κέντρο Ακρόα Mall, Πόρος	filter	9 π.μ. - 6 μ.μ.	34.7055689	32.4137285	94057623	Greek	1 hour 31 mins	driving
29	21-04-2022	Λευκωσία	Οικόπεδο Χ.Χ.Χ.Χ.Χ.	filter	9 π.μ. - 6 μ.μ.			94057623	Greek	filter	driving

Φωτογραφία 4.15 Παράδειγμα πληροφοριών μέσα στη βάση δεδομένων

4.6.2 Διαγραφή αχρείαστων σημείων δειγματοληψίας

Για αποφυγή της επιβάρυνσης της συσκευής του χρήστη καθώς η βάση δεδομένων που χρησιμοποιώ δημιουργεί τοπικό αρχείο στη συσκευή του χρήστη, αποφάσισα όπως σημεία δειγματοληψίας με ημερομηνία παλαιότερη των δύο ημερών από την σημερινή, θα διαγράφονται από τη βάση για εξοικονόμηση χώρου.

Η απόφαση είναι για τα σημεία δειγματοληψίας παλαιότερα των δύο ημερών και όχι μίας ημέρας, καθώς σε περίπτωση που δεν βρω για την σημερινή μέρα τα σημεία δειγματοληψίας, παρουσιάζω στο χρήστη μήνυμα λάθους και τον ρωτώ αν επιθυμεί να δει τα χθεσινά σημεία δειγματοληψίας, επομένως τα χθεσινά σημεία δεν πρέπει να διαγράφονται από την βάση δεδομένων.

```

LocalDate today = LocalDate.now();
String twoDaysBeforeForDeletion = (today.minusDays(2)).format(DateTimeFormatter.ofPattern("dd-MM-yyyy"));

try {
    Cursor res= DB.getDates();
    ArrayList<Date> datesInDb=new ArrayList<Date>();

    ArrayList<Date> datesInDbForDeletion=new ArrayList<Date>();
    while (res.moveToNext()) {
        datesInDb.add(new SimpleDateFormat( patterns "dd-MM-yyyy").parse(res.getString( 0)));
    }
    Date twoDaysBeforeForDeletion2 = new SimpleDateFormat( patterns "dd-MM-yyyy").parse(twoDaysBeforeForDeletion);

    for (int i=0; i<datesInDb.size(); i++){
        if (datesInDb.get(i).compareTo(twoDaysBeforeForDeletion2)<0){
            datesInDbForDeletion.add(datesInDb.get(i));
        }
    }
    SimpleDateFormat dt1 = new SimpleDateFormat( patterns "dd-MM-yyyy");

    for (int i=0; i<datesInDbForDeletion.size(); i++){
        DB.deleteData2(dt1.format(datesInDbForDeletion.get(i)));
    }
}

```

Φωτογραφία 4.16 Διαδικασία διαγραφής σημείων δειγματοληψίας μεγαλύτερα των δύο ημερών

Κεφάλαιο 5

Εκτίμηση και αποτελέσματα

5.1 Ερωτηματολόγιο	35
5.1.1 Παρουσίαση ερωτηματολογίου	35
5.1.2 Ανάλυση ερωτηματολογίου	37
5.2 Αποτελέσματα Ερωτηματολογίου	41
5.2.1 Ανάλυση αποτελεσμάτων	41
5.2.2 Συμπέρασμα αποτελεσμάτων	47

5.1 Ερωτηματολόγιο

Το ερωτηματολόγιο δημιουργήθηκε με σκοπό την αξιολόγηση της εφαρμογής από τους τελικούς χρήστες καθώς οι ανάγκες και οι απαιτήσεις της εφαρμογής είχαν εξαχθεί από την αρχή της διπλωματικής εργασίας. Θα αναλύσω και θα εξηγήσω το λόγο που επιλέχθηκε η κάθε ερώτηση ξεχωριστά και τι αποσκοπεί η επιλογή της συγκεκριμένης ερώτησης στο ερωτηματολόγιο.

5.1.1 Παρουσίαση ερωτηματολογίου

Πόσο συχνά χρησιμοποιείτε την εφαρμογή; *

- ☐ Καθημερινά
- ☐ Τουλάχιστον μια φορά την εβδομάδα
- ☐ Λίγες φορές κατά την διάρκεια της εβδομάδας
- ☐ Λίγες φορές κατά την διάρκεια του μήνα
- ☐ Άλλο: _____

Πόσο εύκολη είναι η χρήση της εφαρμογής; *

- | | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| ☐ | ☐ | ☐ | ☐ | ☐ |

Πόσο σας αρέσει ο σχεδιασμός της εφαρμογής; *

- | | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| ☐ | ☐ | ☐ | ☐ | ☐ |

Επιτύχατε τον σκοπό σας με τη χρήση της εφαρμογής; *

- ☐ Ναι
- ☐ Όχι

Πόσο ευχαριστημένοι είστε από την ταχύτητα της εφαρμογής; *

- | | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| ☐ | ☐ | ☐ | ☐ | ☐ |

Ποια λειτουργικότητα της εφαρμογής θεωρείτε πιο σημαντική; *

- ☐ Αλλαγή γλώσσας
- ☐ Εμφάνιση όλων των σημείων δειγματοληψίας
- ☐ Αλλαγή μέσου μεταφοράς
- ☐ Πληροφορίες
- ☐ Σχετικά με εμάς

Ποια λειτουργικότητα της εφαρμογής θεωρείτε λιγότερο σημαντική; *

☐ Αλλαγή γλώσσας

☐ Εμφάνιση όλων των σημείων δειγματοληψίας

☐ Αλλαγή μέσου μεταφοράς

☐ Πληροφορίες

☐ Σχετικά με εμάς

Πόσο εύκολη είναι η εξεύρεση των κουμπιών για τη χρήση των λειτουργιών; *

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

Υπάρχει κάποια λειτουργία που θα θέλατε να προστεθεί στην εφαρμογή;

Η απάντησή σας _____

Θα προτείνατε αυτή την εφαρμογή σε κάποιο γνωστό σας;

☐ Ναι

☐ Όχι

Αν στη προηγούμενη ερώτηση απαντήσατε ναι, για ποιο λόγο, αν απαντήσατε όχι γιατί δεν θα την πρότεινες;

Η απάντησή σας _____

5.1.2 Ανάλυση ερωτηματολογίου

Το ερωτηματολόγιο που δημιουργήθηκε για τη φάση της αξιολόγησης της εφαρμογής μου αποτελείται από 11 ερωτήσεις. Θα προχωρήσω στη ανάλυση των 11 ερωτήσεων και των λόγων που επιλέχθηκαν για αυτό το ερωτηματολόγιο.

1. Πόσο συχνά χρησιμοποιείτε την εφαρμογή;

Πιθανές απαντήσεις:

- Καθημερινά
- Τουλάχιστο μία φορά την εβδομάδα
- Λίγες φορές κατά την διάρκεια της εβδομάδας
- Λίγες φορές κατά την διάρκεια του μήνα

- Άλλο

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω την ανάγκη των χρηστών στη συνέπεια χρήσης της εφαρμογής με σκοπό να εξακριβώσω πόσο χρησιμοποιείται από τους χρήστες και να σκεφθώ πιθανές βελτιώσεις.

2. Πόσο εύκολη είναι η χρήση της εφαρμογής;

Πιθανές απαντήσεις:

- Κλίμακα αριθμών από 1-5 με το 5 να έχει την μεγαλύτερη βαρύτητα

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω τη ευκολία χρήσης της συσκευής από τους τελικούς χρήστες.

3. Πόσο σας αρέσει ο σχεδιασμός της εφαρμογής;

Πιθανές απαντήσεις:

- Κλίμακα αριθμών από 1-5 με το 5 να έχει την μεγαλύτερη βαρύτητα

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω πόσο αρεστός είναι ο σχεδιασμός της εφαρμογής στους χρήστες για να προχωρήσω σε ανάλογες αλλαγές στο σχεδιασμό της εφαρμογής.

4. Επιτύχετε τον σκοπό σας με τη χρήση της εφαρμογής;

Πιθανές απαντήσεις:

- Ναι
- Όχι

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω αν ο χρήστης με το πέρας της εμπειρίας του με την εφαρμογή, καταφέρνει να επιτύχει τον στόχο του. Αν δεν επιτυγχάνει το στόχο του, τότε είναι καταστροφικό.

5. Πόσο ευχαριστημένοι είστε από την ταχύτητα της εφαρμογής;

Πιθανές απαντήσεις:

- Κλίμακα αριθμών από 1-5 με το 5 να έχει την μεγαλύτερη βαρύτητα

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω την εμπειρία των χρηστών σχετικά με τη ταχύτητα της εφαρμογής καθώς υπάρχουν περιπτώσεις όπου η εφαρμογή παρουσιάζει κάποιες καθυστερήσεις, και ήθελα να δω κατά πόσο αυτές οι καθυστερήσεις επηρεάζουν την εμπειρία των χρηστών με σκοπό να κάνω τις ανάλογες αλλαγές για καλύτερη εμπειρία.

6. Ποια λειτουργικότητα της εφαρμογής θεωρείται πιο σημαντική;

Πιθανές απαντήσεις:

- Αλλαγή γλώσσας
- Εμφάνιση όλων των σημείων δειγματοληψίας
- Αλλαγή μέσου μεταφοράς
- Πληροφορίες
- Σχετικά με εμάς

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω την χρησιμότητα και σημαντικότητα κάποιων κύριων λειτουργιών της εφαρμογής έτσι ώστε να της βελτιώσω ακόμη περισσότερο για καλύτερη εμπειρία του χρήστη.

7. Ποια λειτουργικότητα της εφαρμογής θεωρείται λιγότερο σημαντική;

Πιθανές απαντήσεις:

- Αλλαγή γλώσσας
- Εμφάνιση όλων των σημείων δειγματοληψίας
- Αλλαγή μέσου μεταφοράς
- Πληροφορίες

- Σχετικά με εμάς

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω την σημαντικότητα κάποιων κύριων λειτουργιών της εφαρμογής έτσι ώστε να προβώ σε βελτιώσεις ή σε κατάργηση κάποιων λειτουργιών που δεν είναι χρήσιμες.

8. Πόσο εύκολη είναι η εξεύρεση των κουμπιών για τη χρήση των λειτουργιών;

Πιθανές απαντήσεις:

- Κλίμακα αριθμών από 1-5 με το 5 να έχει την μεγαλύτερη βαρύτητα

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω τη ευκολία χρήσης της εφαρμογής από τους χρήστες. Αν παρουσιάζετε κάποια δυσκολία εξεύρεσης των κουμπιών για τις λειτουργίες θα πρέπει να προβώ στις ανάλογες αλλαγές έτσι ώστε να είναι όσο το δυνατό πιο εύκολη η χρήση της εφαρμογής από το χρήστη.

9. Υπάρχει κάποια λειτουργία που θα θέλατε να προστεθεί στη εφαρμογή;

Πιθανές απαντήσεις:

- Ελεύθερο

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να έχω την άποψη των τελικών χρηστών με λειτουργίες που θα ήθελαν να προστεθούν στην εφαρμογή με σκοπό την ανάπτυξη τους.

10. Θα προτείνατε αυτή την εφαρμογή σε κάποιο γνωστό σας;

Πιθανές απαντήσεις:

- Ναι
- Όχι

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι για να εξακριβώσω την πρόθεση των χρηστών να μοιραστούν την εφαρμογή με κάποιο γνωστό, που μου δείχνει ότι είχαν καλή εμπειρία χρήσης και πέτυχαν το σκοπό τους.

11. Αν στη προηγούμενη ερώτηση απαντήσατε ναι, για ποιο λόγο, αν απαντήσατε όχι γιατί δεν θα την προτείνετε;

Πιθανές απαντήσεις:

- Ελεύθερο

Ο λόγος που επιλέχθηκε η συγκεκριμένη ερώτηση είναι η άμεση σύνδεση της με την προηγούμενη ερώτηση. Ο σκοπός είναι να ακούσω την άποψη του χρήστη για το λόγο που θα πρότεινε την εφαρμογή μου σε κάποιο γνωστό έτσι ώστε να καταλάβω τι κίνησε το χρήστη στο να την προτείνει, και αν όχι να λάβω υπόψη το λόγο που δεν θα την πρότεινε για τυχόν βελτιώσεις.

5.2 Αποτελέσματα Ερωτηματολογίου

Θα παρουσιάσω τα αποτελέσματα του ερωτηματολογίου και στη συνέχεια θα τα αναλύσω βγάζοντας χρήσιμα συμπεράσματα σχετικά με την υλοποίηση της εφαρμογής μου. Αρχικά θα αναλύσω τα αποτελέσματα του ερωτηματολογίου και στη συνέχεια θα εξάγω τα συμπεράσματα.

5.2.1 Ανάλυση αποτελεσμάτων

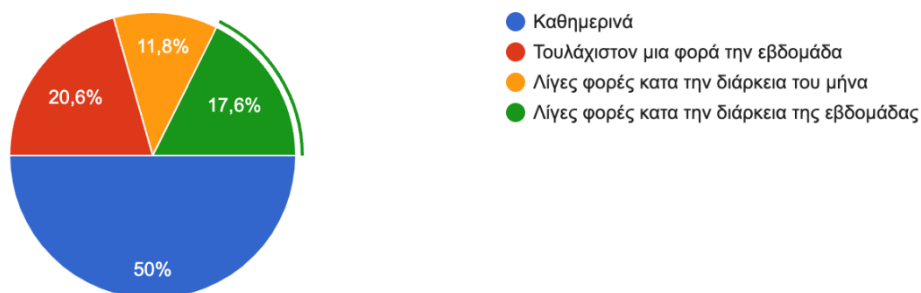
Το ερωτηματολόγιο που δημιουργήθηκε για τη φάση της αξιολόγησης της εφαρμογής απαντήθηκε από 34 άτομα.

Από τα 34 άτομα το 50% χρησιμοποιεί την εφαρμογή μου σε καθημερινή βάση, ενώ το υπόλοιπο 50% χωρίζεται στις υπόλοιπες επιλογές με την επιλογή «Τουλάχιστον μια φορά την εβδομάδα» να υπερσχύει ελάχιστα με την επιλογή «Λίγες φορές κατά την διάρκεια της εβδομάδας».

Πόσο συχνά χρησιμοποιείτε την εφαρμογή;

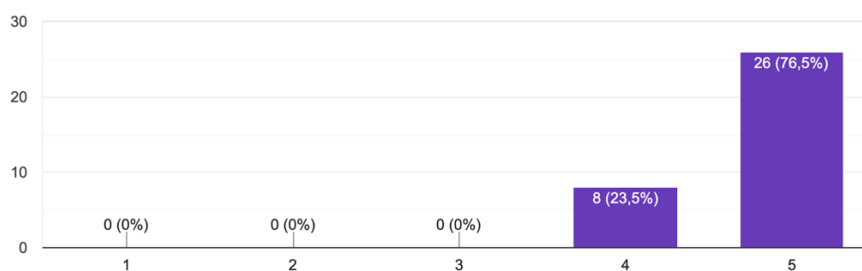
Αντιγραφή

34 απαντήσεις



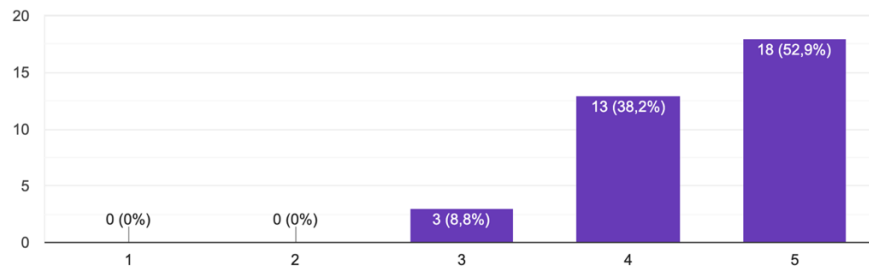
Στην ερώτηση για το πόσο εύκολη είναι η χρήση της εφαρμογής η συντριπτική πλειοψηφία με 76,5 % επέλεξε την επιλογή 5 ενώ το υπόλοιπο 23,5% επέλεξε την επιλογή 4.

Πόσο εύκολη είναι η χρήση της εφαρμογής;
34 απαντήσεις



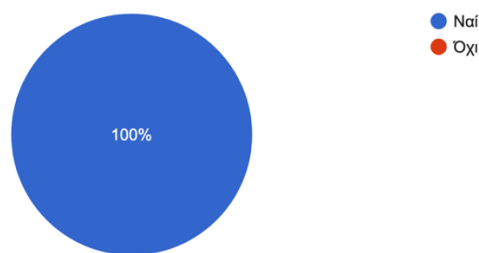
Στην επόμενη ερώτηση σχετικά με την αρέσκεια του χρήστη ως προς τον σχεδιασμό της εφαρμογής περισσότερο από το 50% επέλεξε την επιλογή 5 ενώ το υπόλοιπο 50% μοιράστηκε στις επιλογές 4 και 3 με την επιλογή 4 να παίρνει το μεγαλύτερο μερίδιο επιλογής.

Πόσο σας αρέσει ο σχεδιασμός της εφαρμογής;
34 απαντήσεις



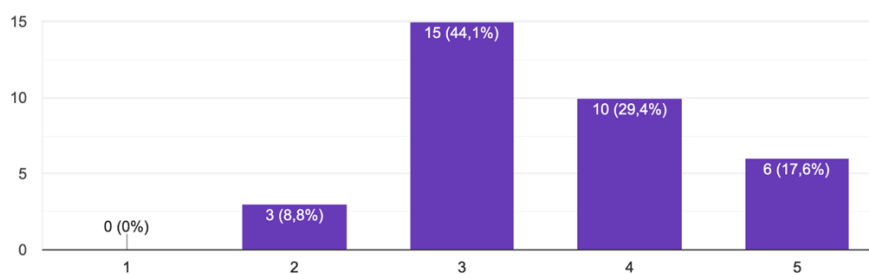
Στην ερώτηση σχετικά με το αν κατάφερε ο χρήστης να επιτύχει τον σκοπό του με τη χρήση της εφαρμογής, όλοι οι χρήστες επέλεξαν την επιλογή «Ναι».

Επιτύχατε τον σκοπό σας με τη χρήση της εφαρμογής;
34 απαντήσεις



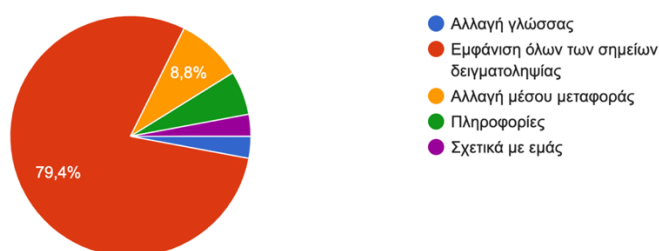
Η ερώτηση σχετικά με την ταχύτητα της εφαρμογής είχε τις πιο πολλές διαφορετικές επιλογές, καθώς οι περισσότεροι χρήστες επέλεξαν την επιλογή 3 και 4 με την επιλογή 2 να παίρνει το μικρότερο ποσοστό επιλογής και έπειτα να ακολουθεί η επιλογή 5.

Πόσο ευχαριστημένοι είστε από την ταχύτητα της εφαρμογής;
34 απαντήσεις



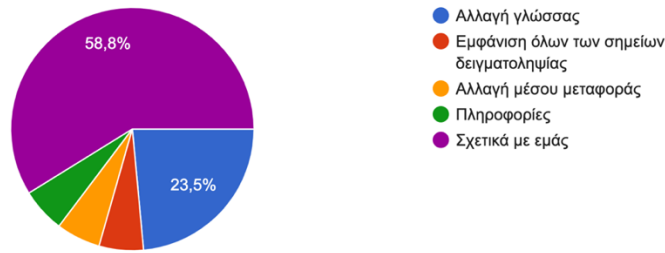
Η ερώτηση σχετικά με τη σημαντικότερη λειτουργία της εφαρμογής είχε το μεγαλύτερο ποσοστό στη επιλογή «Εμφάνιση όλων των σημείων δειγματοληψίας» με σχεδόν 80% των χρηστών να την επιλέγουν, ενώ οι υπόλοιποι χρήστες μοίρασαν σχεδόν ισάξια τις ψήφους τους στις υπόλοιπες επιλογές.

Ποια λειτουργικότητα της εφαρμογής θεωρείτε πιο σημαντική;
34 απαντήσεις



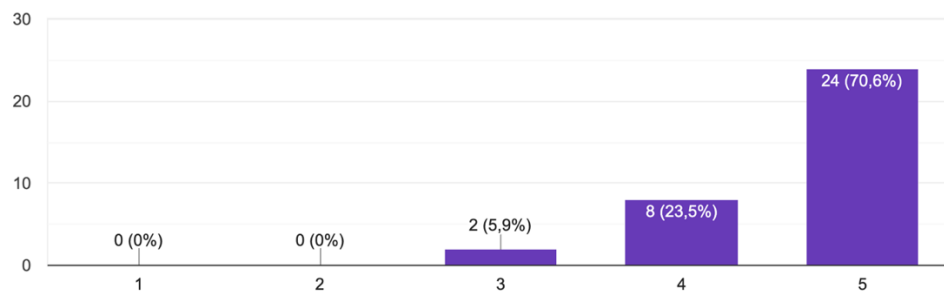
Η ερώτηση σχετικά με την λιγότερο σημαντική λειτουργία της εφαρμογής προσέελκυσε το 58.5% των χρηστών στην επιλογή «Σχετικά με εμάς», ενώ η επιλογή «Αλλαγή γλώσσας» ήταν η επόμενη πιο επιλεγμένη λειτουργία.

Ποια λειτουργικότητα της εφαρμογής θεωρείτε λιγότερο σημαντική;
34 απαντήσεις



Στην ερώτηση που αναφερόταν σχετικά με την ευκολία εξεύρεσης των κουμπιών για τη χρήση των διαφόρων λειτουργιών, το μεγαλύτερο ποσοστό των χρηστών επέλεξε την επιλογή 5 με την επιλογή 3 να έχει μόνο 2 ψήφους.

Πόσο εύκολη είναι η εξεύρεση των κουμπιών για τη χρήση των λειτουργιών;
34 απαντήσεις



Στην ανοικτή ερώτηση με σκοπό την συμμετοχή των χρηστών στη υλοποίηση της εφαρμογής με λειτουργίες που θα ήθελαν να προστεθούν στην εφαρμογή είχα διάφορες απαντήσεις, όπως:

- Όχι
- Να καταχωρηθούν όλα τα κέντρα εμβολιασμού
- Εμφάνιση τοποθεσιών
- Εάν υπάρχουν διαδρομές με λεωφορείο προς τα σημεία δειγματοληψίας

Υπάρχει κάποια λειτουργία που θα θέλατε να προστεθεί στην εφαρμογή;

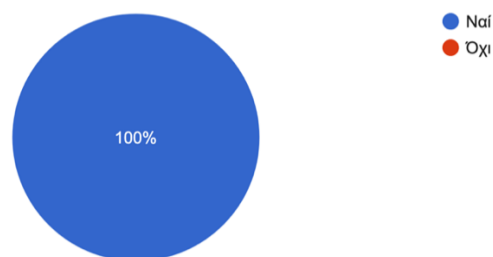
8 απαντήσεις

- Όχι
- Όχι
- να καταχωρηθούν όλα τα κέντρα εμβολιασμού
- Εμφάνιση τοποθεσιών
- ΟΧΙ
- Εάν υπάρχουν διαδρομές με λεωφορείο προς τα σημεία δειγματοληψίας

Στη επόμενη ερώτηση σχετικά με το αν θα πρότειναν την εφαρμογή σε κάποιο γνωστό τους, όλοι οι χρήστες επέλεξαν την επιλογή «Ναι».

Θα προτείνατε αυτή την εφαρμογή σε κάποιο γνωστό σας;

33 απαντήσεις



Στη τελευταία ερώτηση που συνδέεται με την προηγούμενη ερώτηση, υπήρξαν διάφορες απαντήσεις για τους λόγους για τους οποίους οι χρήστες θα πρότειναν την εφαρμογή σε κάποιο γνωστό τους, όπως:

- Είναι πολύ χρήσιμη
- Μπορεί να χρειαστεί ανά πάσα στιγμή
- Επειδή λύνει το πρόβλημα του να ψάχνεις για σημεία δειγματοληψίας
- Διευκόλυνση στην καθημερινότητα για εύρεση κέντρου δειγματοληψίας
- Γρήγορη και χρήσιμη
- Θεωρώ ότι είναι χρήσιμη
- Βρίσκει γρήγορα Rapid Test

- Εύκολη χρήση και γρήγορη εφαρμογή
- Εύκολη και πρακτική χρήση, δεν χρειάζεται να ψάχνεις εφημερίδες και ανακοινωθέντα τύπου για να βρεις όλες αυτές τις πληροφορίες.

Αν στη προηγούμενη ερώτηση απαντήσατε ναι, για ποιο λόγο, αν απαντήσατε όχι γιατί δεν θα την πρότεινες;

13 απαντήσεις

είναι πολύ χρυσή
μπορεί να χρειαστεί ανά πάσα στιγμή
Επειδή λύνει το πρόβλημα του να ψάχνεις για σημεία δειγματοληψίας
Διευκόλυνση στην καθημερινότητα για εύρεση κέντρου δειγματοληψίας
γρήγορη και χρήσιμη
Θεωρώ ότι είναι χρήσιμη
na vriskei grigora rapid test
Ευκολή χρήση και γρήγορη εφαρμογή
Εύκολη και πρακτική χρήση, δεν χρειάζεται να ψάχνεις εφημερίδες και ανακοινωθέντα τύπου για να βρεις όλες αυτές τις πληροφορίες

5.2.2 Συμπεράσματα αποτελεσμάτων

Έχουμε αναλύσει τα αποτελέσματα του ερωτηματολογίου στα πλαίσια της αξιολόγησης της εφαρμογής από τους τελικούς χρήστες ως:

- Ερωτήσεις που λάμβαναν ποσοστό πέραν του 70% σε απάντηση “Ναι” τότε η αξιολόγηση που συσχετίζεται με αυτή την ερώτηση καθορίζεται ως άκρως σημαντική και πρέπει να ληφθεί υπόψη.
- Ερωτήσεις με απαντήσεις που λάμβαναν ποσοστό πάνω από 50% τότε η επιλογή που συσχετίζεται με αυτή την ερώτηση καθορίζεται ως άκρως σημαντική και πρέπει να ληφθεί υπόψη.
- Ερωτήσεις με απαντήσεις που λάμβαναν ποσοστό πάνω από 40% τότε η επιλογή που συσχετίζεται με αυτή την ερώτηση καθορίζεται ως σημαντική και πρέπει να ληφθεί υπόψη.
- Ερωτήσεις όπου λάμβαναν απαντήσεις ελάχιστα δημοφιλέστερες σε ερωτήματα αξιολόγησης τις θεώρησα ως αξιολογήσεις που δεν πρέπει να συμπεριληφθούν.
- Ερωτήσεις όπου η απάντηση των χρηστών ήταν ελεύθερη, όλες οι επιλογές θα ληφθούν υπόψη και θα θεωρηθούν άκρως σημαντικές.

Άκρως σημαντικές αξιολογήσεις σε ερωτήσεις επιλογών Ναι/Όχι:

- Οι χρήστες επέτυχαν το σκοπό τους με τη χρήση της εφαρμογής
- Οι χρήστες θα πρότειναν την εφαρμογή σε κάποιο γνωστό τους

Επομένως, από αυτές τις ερωτήσεις μπορούμε να εξάγουμε το συμπέρασμα ότι οι χρήστες πέτυχαν το σκοπό τους με τη χρήση της εφαρμογής και είχαν καλή εμπειρία χρήσης αφού θα πρότειναν την εφαρμογή και σε κάποιο γνωστό τους.

Άκρως σημαντικές αξιολογήσεις σε ερωτήσεις πολλαπλών επιλογών:

- Οι χρήστες χρησιμοποιούν την εφαρμογή σε καθημερινή βάση
- Η χρήση της εφαρμογής είναι πολύ εύκολη
- Ο σχεδιασμός της εφαρμογής αρέσει στους χρήστες
- Η πιο σημαντική λειτουργία είναι η εμφάνιση όλων των σημείων δειγματοληψίας
- Η λιγότερο σημαντική λειτουργία είναι η σχετικά με εμάς
- Η εξεύρεση των κουμπιών για τις λειτουργίες είναι πολύ εύκολη

Επομένως από αυτές τις ερωτήσεις μπορούμε να εξάγουμε το συμπέρασμα ότι η λειτουργία σχετικά με εμάς είναι λίγο σημαντική και δεν προτιμάται από τους χρήστες. Επίσης η χρήση της εφαρμογής, ο σχεδιασμός, η καθημερινή χρήση και η εύκολη χρήση της εφαρμογής είναι σημαντικά θετικά στοιχεία.

Σημαντικές αξιολογήσεις σε ερωτήσεις πολλαπλών επιλογών:

- Οι χρήστες δεν είναι απόλυτα ευχαριστημένοι με την ταχύτητα της εφαρμογής

Επομένως, από αυτή τη ερώτηση μπορούμε να εξάγουμε το συμπέρασμα ότι η ταχύτητα της εφαρμογής μας θα πρέπει να βελτιωθεί για καλύτερη εμπειρία των χρηστών.

Άκρως σημαντικές αξιολογήσεις σε ερωτήσεις ελεύθερης απάντησης:

- Να καταχωρηθούν όλα τα κέντρα εμβολιασμού
- Εμφάνιση τοποθεσιών
- Εάν υπάρχουν διαδρομές με λεωφορείο προς τα σημεία δειγματοληψίας
- Είναι πολύ χρήσιμη
- Μπορεί να χρειαστεί ανά πάσα στιγμή
- Επειδή λύνει το πρόβλημα του να ψάχνεις για σημεία δειγματοληψίας

- Διευκόλυνση στην καθημερινότητα για εύρεση κέντρου δειγματοληψίας
- Γρήγορη και χρήσιμη
- Θεωρώ ότι είναι χρήσιμη
- Βρίσκει γρήγορα Rapid Test
- Εύκολη χρήση και γρήγορη εφαρμογή
- Εύκολη και πρακτική χρήση, δεν χρειάζεται να ψάχνεις εφημερίδες και ανακοινωθέντα τύπου για να βρεις όλες αυτές τις πληροφορίες.

Επομένως, από αυτές τις ερωτήσεις μπορούμε να εξάγουμε το συμπέρασμα ότι το να καταχωρηθούν όλα τα κέντρα εμβολιασμού, να γίνεται εμφάνιση των τοποθεσιών, να μπορεί ο χρήστης να δει αν υπάρχουν διαδρομές με λεωφορείο προς τα σημεία δειγματοληψίας είναι μερικές λειτουργίες οι οποίες θα ήταν σημαντικό να προστεθούν στη εφαρμογή.

Ακόμα, σχετικά με τους λόγους που θα πρότειναν την εφαρμογή σε γνωστούς, όλες οι απαντήσεις είναι ενθαρρυντικές και θετικές.

Κεφάλαιο 6

Συμπεράσματα

6.1 Σχολιασμός αποτελεσμάτων	50
6.1.1 Αποτελέσματα από Twitter API	50
6.1.2 Αποτελέσματα από Google API	51
6.1.3 Αποτελέσματα από HTML Web Parsing	52
6.1.4 Αποτελέσματα από Android Studio	53
6.1.5 Αποτελέσματα από βάση δεδομένων	54
6.1.6 Αποτελέσματα εφαρμογής σε σχέση με ταχύτητα	54
6.2 Συμπεράσματα	54
6.3 Μελλοντική εργασία	55

6.1 Σχολιασμός αποτελεσμάτων

Με το τέλος της υλοποίησης της εφαρμογής ακολούθησε αξιολόγηση της εφαρμογής από τους τελικούς χρήστες για να βρεθούν πιθανά προβλήματα όπως και τρόποι βελτίωσης της εφαρμογής. Η εφαρμογή λειτουργεί σε τηλέφωνα με λογισμικό Android, επομένως ήταν σχετικά εύκολο να αποστείλω την εφαρμογή στους χρήστες για δοκιμή. Τα αποτελέσματα που θα αναλυθούν σε αυτό το κεφάλαιο αφορούν την απόδοση της εφαρμογής, περιορισμούς που είχα στην υλοποίηση της εφαρμογής.

6.1.1 Αποτελέσματα από Twitter API

Σχετικά με το Twitter API το οποίο χρησιμοποιήθηκε με σκοπό την ανάκτηση της καθημερινής δημοσίευσης του Γραφείου Τύπου και Πληροφοριών, η ανάκτηση των πληροφοριών ήταν πάντα γρήγορη, αξιόπιστη και χωρίς προβλήματα. Η κλήση της μεθόδου ήταν απλή και η σωστή δομή του API μου επέτρεψε να φιλτράρω χωρίς κανένα πρόβλημα τις δημοσιεύσεις ωσότου να βρω την συγκεκριμένη που έψαχνα. Όμως, αντιμετώπισα κάποια θέματα σχετικά με το περιεχόμενο των δημοσιεύσεων που επέστρεφε. Κάποιες δημοσιεύσεις οι οποίες περιείχαν κάποιου είδους “emojis” δεν ήταν δυνατόν να της χειριστώ κατάλληλα αφού τα “emojis” είχαν κάποια συγκεκριμένη encoded μορφή. Ακόμα, κάποιες φορές το URL που περιείχε η δημοσίευση δεν ήταν πάντα στη ίδια μορφή, καθώς σμικρυνόταν με μια μορφή την οποία χρησιμοποιεί το Twitter, με κάποιες από αυτές τις μορφές να παρουσιάζουν κάποια προβλήματα στη προσπάθεια μου να ανακτήσω τις πληροφορίες από την ιστοσελίδα για τα σημεία δειγματοληψίας. Στη πρώτη φωτογραφία, φαίνεται ένα παράδειγμα δημοσίευσης η οποία περιέχει “emojis” στη θέση των αριθμών που υποδεικνύουν την ημερομηνία, κάτι που δεν μου δίνει τη δυνατότητα να ανακτήσω αυτό τον αριθμό που μπορούσε να βοηθήσει στο φιλτράρισμα πολλών δημοσιεύσεων που μπορεί να γίνουν σε μία μέρα με σκοπό να επιλέξω τη μέρα που θέλω. Στη δεύτερη και τρίτη φωτογραφία παρατηρείται η διαφορά μεταξύ της σμίκρυνσης του URL. Στη δεύτερη φωτογραφία η συγκεκριμένη δομή είχε κάποια προβλήματα όταν προσπαθούσα να ανακτήσω ολόκληρο τον σύνδεσμο από το URL σε σμίκρυνση, καθώς είχε τη μορφή “https”, κάτι που δεν με άφηνε να ενωθώ στη σελίδα, ενώ με την αλλαγή του “https” σε “http” το θέμα αυτό λύθηκε. Στη τρίτη φωτογραφία η συγκεκριμένη μορφή ήταν που παρουσίασε τα αρχικά προβλήματα, καθώς όταν προσπάθησα να ενωθώ με τη συγκεκριμένη μορφή σμίκρυνσης παρουσιάζονταν διάφορα προβλήματα στην ανάκτηση των πληροφοριών από τη σελίδα το οποίο αντιμετώπισα ανακτώντας ολόκληρο το URL στη κανονική του μορφή.



Σχήμα 6.1 Φωτογραφίες από tweets

6.1.2 Αποτελέσματα από Google API

Η βάση της υλοποίησης μου ήταν το Google API από το οποίο χρησιμοποίησα διάφορα κομμάτια, όπως το Maps API, Directions API, Geocoding API που συνέβαλαν στο μέγιστο βαθμό για την ολοκλήρωση αυτής της υλοποίησης. Η ευκολία χρήσης των πιο πάνω API's όπως και η ευκολία της δομής των πληροφοριών ήταν κομβικά και ιδιαίτερα βοηθητικά. Αρχικά, με το Maps API δημιούργησα τη βάση της εφαρμογής η οποία ήταν ένας χάρτης. Με την υλοποίηση του χάρτη, το επόμενο βήμα ήταν η εύρεση της τοποθεσίας του χρήστη και τοποθέτηση της στο χάρτη. Δεν παρουσιάστηκε κανένα πρόβλημα ούτε περιορισμός με τη χρήση του Maps API. Στη συνέχεια, χρησιμοποιώντας το Geocoding API έπρεπε να βρω συντεταγμένες για τα σημεία δειγματοληψίας χρησιμοποιώντας το όνομα τους και την επαρχία τους. Αυτό το κομμάτι ήταν το κομμάτι

που είχα περιορισμούς καθώς κάποιες ονομασίες δεν ταίριαζαν με τις ονομασίες που είχε η Google με αποτέλεσμα να μην έχουμε αποτελέσματα. Σε πολλές περιπτώσεις τις οποίες αποφάσισα να αποκλείσω, είχα ανακριβείς αποτελέσματα που θα μπορούσαν να οδηγήσουν το χρήστη σε λάθος επιλογή, το οποίο θα ήταν καταστροφικό με το χρήστη να έχει μια πολύ κακή εμπειρία χρήσης. Τέλος, το τελευταίο API που χρησιμοποίησα από την Google ήταν το Directions API με σκοπό να βρω δρομολόγια από την τοποθεσία του χρήστη προς το επιλεγμένο σημείο δειγματοληψίας. Δεν παρουσιάστηκε κανένα πρόβλημα ούτε περιορισμός με τη χρήση του Directions API, παρά μόνο ένας μικρός βαθμός δυσκολίας για την μετατροπή της encoded μορφής του δρομολογίου σε μορφή λίστας από συντεταγμένες.

6.1.3 Αποτελέσματα από HTML Web Parsing

Ένα από τα σημαντικότερα βήματα της υλοποίησης μου ήταν το HTML Web Parsing το οποίο μου επέτρεψε να ανακτήσω της πληροφορίες από την ιστοσελίδα του Γραφείου Τύπου και Πληροφοριών χρησιμοποιώντας στοιχεία HTML. Αυτός ο τρόπος ήταν ο μοναδικός που μπορούσα να χρησιμοποιήσω για να ανακτήσω της πληροφορίες σχετικά με τα σημεία δειγματοληψίας, καθώς δεν υπήρχε δυνατότητα άμεσης επικοινωνίας με το Γραφείο Τύπου και Πληροφοριών. Με το συγκεκριμένο τρόπο είχα κάποιους περιορισμούς στην ανάκτηση πληροφοριών καθώς ήταν ιδιαίτερα ευαίσθητος στο ανθρώπινο παράγοντα. Δεν υπήρχε η δυνατότητα για εντελώς αυτοματοποιημένη διαδικασία, καθώς η δομή HTML της ιστοσελίδας δεν ήταν πάντα σταθερή, και ένα απλό λάθος όπως μια κενή γραμμή στο σημείο όπου παρουσιάζονταν τα σημεία δειγματοληψίας, μπορούσε να ήταν καταστροφικό. Οι περιπτώσεις που μπορούσαν να συμβούν ήταν πάρα πολλές για αυτό και δεν ήταν εφικτό για έλεγχο όλων των περιπτώσεων μια προς μια. Στην φωτογραφία παρουσιάζεται μια περίπτωση η οποία προέκυψε με καταστροφικό αποτέλεσμα για την εφαρμογή.

Αμμόχωστος (3 σημεία)	Δημοτικό Μέγαρο Παραλιμνίου (στεγασμένος χώρος στάθμευσης)	9 π.μ. - 6 μ.μ.	94041843
	Εθνικόφρονα Σωματεία Λιοπετρίου	9 π.μ. - 6 μ.μ.	94056785
	Εθνικόφρονα Σωματεία Λιοπετρίου	9 π.μ. - 6 μ.μ.	94056785

Φωτογραφία 6.2 Παράδειγμα που οδήγησε την υλοποίηση σε σφάλμα

6.1.4 Αποτελέσματα από Android Studio

Το λογισμικό Android Studio ήταν η επιλογή μου σχετικά με τη δημιουργία της εφαρμογής. Επέλεξα το συγκεκριμένο λογισμικό καθώς υποστήριζε τη γλώσσα προγραμματισμού Java την οποία διδάχθηκα σε μεγάλο βαθμό κατά την διάρκεια των ακαδημαϊκών σπουδών μου. Το μοναδικό θέμα που προέκυψε με τη χρήση του Android Studio ήταν ο ασύγχρονος τρόπος λειτουργίας των εφαρμογών Android που με περιόρισε σε κάποια σημεία. Τα σημεία που με περιόρισε ήταν κυρίως η εντελώς αυτοματοποιημένη λειτουργία της εφαρμογής χωρίς τη χρήση κουμπιού για έναρξη αναζήτησης για τα σημεία δειγματοληψίας, καθώς σε πολλές περιπτώσεις δεν είναι δυνατόν να ανακτηθεί η τοποθεσία του χρήστη με το άνοιγμα της εφαρμογής παίρνοντας ένα ελάχιστο χρονικό διάστημα. Ακόμα, στη προσπάθειά μου για καλύτερη εμπειρία χρήσης χρησιμοποιώντας διάφορα μηνύματα προς ενημέρωση του χρήστη για την κατάσταση της εφαρμογής, όταν προσπάθησα να δημιουργήσω ένα loading animation που θα έδειχνε στο χρήστη ότι η εφαρμογή είναι σε κατάσταση αναμονής λόγω της φόρτωσης και αναζήτησης δεδομένων, παρατήρησα ότι δεν μπορούσε να δουλέψει σωστά λόγω του ασυγχρονισμού της εφαρμογής (πρώτα εκτελούσε το κώδικα ανάμεσα στις εντολές για έναρξη και τερματισμό του loading animation, και μετά εμφάνιζε το loading animation). Τέλος, με την χρήση του Android Studio είχα την δυνατότητα να

δημιουργήσω ξεχωριστές οθόνες για κάθε λειτουργία που ήθελα που ήταν ιδιαίτερα βοηθητικό.

6.1.5 Αποτελέσματα από βάση δεδομένων

Η βάση δεδομένων ήταν ένα πολύ σημαντικό κομμάτι στην υλοποίηση μου καθώς χρησιμοποιείται σε πολλές λειτουργίες όπως σε περιπτώσεις που αλλάζω οθόνη και υπάρχει η ανάγκη για μεταφορά πληροφοριών, όπως και επίσης σε περιπτώσεις όπου ανακτώ τα δεδομένα τα οποία ήδη έχω για αποφυγή περιττών κλήσεων στα API που μπορεί να επιφέρουν χρεώσεις αλλά και σπατάλη χρόνου. Ο μοναδικός περιορισμός που είχα με την βάση δεδομένων είναι ο τρόπος λειτουργίας της συγκεκριμένης βάσης δεδομένων, καθώς δημιουργώ ένα τοπικό αρχείο στη συσκευή του χρήστη. Χωρίς τις ανάλογες ενέργειες, αυτό το αρχείο μπορεί να γίνει τεράστιο και να χρησιμοποιεί ένα μεγάλο μερίδιο μνήμης της συσκευής του χρήστη.

6.1.6 Αποτελέσματα εφαρμογής σε σχέση με ταχύτητα

Για την σύγκριση της εφαρμογής σε σχέση με την ταχύτητα, έτρεξα την εφαρμογή σε 3 διαφορετικές ώρες, 10 φορές σε κάθε ώρα και κατέγραψα το χρόνο τον οποίο χρειάζεται για την αναζήτηση και εύρεση των πληροφοριών σχετικά με τα σημεία δειγματοληψίας. Ο μέσος όρος ανάκτησης των πληροφοριών ήταν 15 (15.3) δευτερόλεπτα, ένας αριθμός τον οποίο θεώρησα σχετικά μεγάλο αλλά όχι καταστροφικό. Επομένως, αποφάσισα όπως αποθηκεύω στη βάση δεδομένων τις πληροφορίες για τα σημεία δειγματοληψίας, έτσι ώστε να εξοικονομείται χρόνος κάθε φορά όπου θα χρειαστεί ανάκτηση αυτών των δεδομένων. Με αυτή τη κίνηση, ο χρόνος ανάκτησης δεδομένων μειώθηκε δραματικά από τα 15 δευτερόλεπτα, στα μόλις 4 (3.9) δευτερόλεπτα. Χρόνος ο οποίος θεώρησα ότι είναι εξαιρετικός και θα προσφέρει στο χρήστη την κατάλληλη εμπειρία χρήσης.

6.2 Συμπεράσματα

Είναι σημαντικό να αναφερθεί ότι πέραν της θεωρητικής και πρακτικής γνώσης που απόκτησα κατά την έρευνα και την ανάπτυξη της εφαρμογής, δημιουργήθηκε μια εφαρμογή που θα μπορεί να βοηθήσει άτομα τα οποία θέλουν μια αυτοματοποιημένη

διαδικασία για τη συγκεκριμένη διαδικασία εξεύρεσης σημείου δειγματοληψίας για διενέργεια Rapid Test μέσω του προγράμματος του Γραφείου Τύπου και Πληροφοριών. Επίσης, τέθηκαν οι βάσεις για την ανάπτυξη και επιπρόσθετων εφαρμογών παρόμοιας λειτουργικότητας, οι οποίες θα αυτοματοποιούν διαδικασίες για τους πολίτες.

Αρχικά, είναι σημαντικό να αναφερθεί ότι λόγω της έλλειψης επικοινωνίας με το Γραφείο Τύπου και Πληροφοριών η υλοποίηση της εφαρμογής είναι ευαίσθητη στον ανθρώπινο παράγοντα. Σε περίπτωση που υπάρξει μια συνεργασία μεταξύ του Γραφείου Τύπου και Πληροφοριών, χρησιμοποιώντας ένα κοινό πλαίσιο για την παρουσίαση των σημείων δειγματοληψίας και τήρηση κάποιων σημαντικών συνέπειων στη δημοσίευση αλλά και στη ονομασία των σημείων δειγματοληψίας, η υλοποίηση της εφαρμογής θα λειτουργήσει εντελώς αυτοματοποιημένα παρουσιάζοντας στο χρήστη όλες τις δυνατές πληροφορίες χωρίς περιθώριο λάθους.

Μπορεί η υλοποίηση τη δεδομένη στιγμή να μην μπορεί να παρουσιάσει όλες τις δυνατές πληροφορίες στο χρήστη, έχουν όμως μπει τα θεμέλια για μια ενδεικτική λύση που συνδυάζει τη ανάγκη των πολιτών για αυτοματοποιημένη διαδικασία εξεύρεσης πληροφοριών με την χρήση των κινητών τηλεφώνων με το πάτημα ενός κουμπιού.

6.3 Μελλοντική εργασία

Με τον τρόπο υλοποίησης της εφαρμογής είναι εφικτό να προστεθούν και διάφορες άλλες λειτουργίες, όπως και επίσης βελτίωση ορισμένων λειτουργιών. Αρχικά, υπάρχουν λειτουργίες που θα ήταν καλό να προστεθούν στην εφαρμογή, όπως η παρουσίαση των κέντρων εμβολιασμού με τον ίδιο τρόπο που παρουσιάζονται και τα σημεία δειγματοληψίας, και με τις ίδιες λειτουργίες. Επίσης, καλό θα ήταν να προστεθούν και διάφορες άλλες πληροφορίες σχετικά με τα μέτρα που ισχύουν τη δεδομένη στιγμή στη χώρα μας όπως και τις διάφορες τοποθεσίες όπου διενεργούνται δειγματοληψίας στα πλαίσια του “Test to Stay”. Επιπρόσθετα, μια σημαντική παράμετρος που συνδέεται άμεσα και με το σκοπό του ερευνητικού κέντρου RISE και ιδιαίτερα του τομέα Smart City, είναι η εμφάνιση δρομολογίων με τη χρήση λεωφορείων. Με αυτό το τρόπο θα παρακινήσουμε τους χρήστες για χρήση των λεωφορείων με σκοπό την περιβαλλοντική προστασία.

Τέλος, σε θέματα βελτίωσης των υφιστάμενων λειτουργιών, θα μπορούσε να δοθεί η επιλογή στο χρήστη για να επιλέξει παραμέτρους αναζήτησης των σημείων δειγματοληψίας, το οποίο θα επέτρεπε και την εμφάνιση σημείων δειγματοληψίας των οποίων η τοποθεσία μπορεί να μην ήταν ακριβής (Geocoding API). Ακόμα ένα σημείο το οποίο επιβάλλεται να διορθωθεί είναι η παρουσίαση του loading animation το οποίο είναι ήδη υλοποιημένο αλλά εξαιτίας της ασύγχρονης φύσης των Android εφαρμογών δεν δουλεύει σωστά. Κλείνοντας, με τη χρήση μιας κανονικής βάσης δεδομένων σε διακομιστή μπορούμε να αποφύγουμε τη δέσμευση χώρου στη συσκευή του χρήστη, αλλά κυρίως να μπορούμε να ανακτούμε κάποια δεδομένα τα οποία χρειάζονται κάποιου είδους ασφάλεια (encoded μορφή), όπως τα API keys που χρησιμοποιούνται για την κλήση του Google και Twitter API.

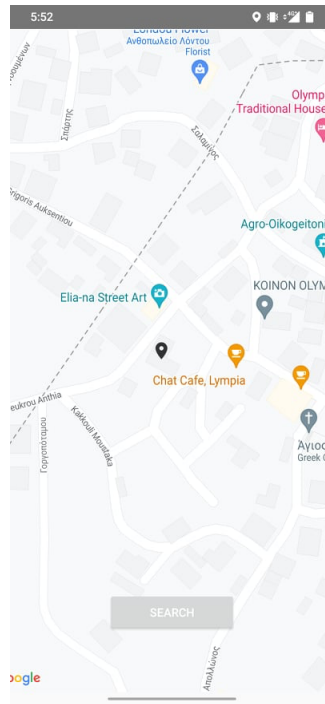
Βιβλιογραφία

- [1] Anders Møller, Michael I. Schwartzbach, "Static Program Analysis", 2022.
- [2] Android Studio tool: <https://developer.android.com/studio> .
- [3] Martin, James, Programming Real-time Computer Systems, Englewood Cliffs, NJ: Prentice-Hall Inc, page 4.
- [4] Fielding, Roy T.; Gettys, James; Mogul, Jeffrey C.; Nielsen, Henrik Frystyk; Masinter, Larry; Leach, Paul J.; Berners-Lee, Tim "Hypertext Transfer Protocol – HTTP/1.1", 1999 .
- [5] "Find the route to your nearest Covid-19 vaccination center in Germany — new App by HeiGIT based on OpenStreetMap and openrouteservice" <http://klz.blog.uni-heidelberg.de/2021/01/19/find-the-route-to-your-nearest-covid-19-vaccination-center-in-germany-new-app-by-heigit-based-on-openstreetmap-and-openrouteservice/>
- [6] "Getting Started with HTML", https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started, 2022
- [7] Google Maps API's manuals: <https://developers.google.com/maps>
- [8] "How do APIs work?", https://www.redhat.com/cms/managed-files/styles/wysiwyg_full_width/s3/API-page-graphic.png?itok=5zMemph9
- [9] "How HTTP works", https://cdn.ttgtmedia.com/rms/onlineimages/whatis-how_http_works.png
- [10] Jsoup HTML Parser tool: <https://jsoup.org/apidocs>
- [11] Lane Kin , "Intro to APIs: History of APIs", <https://blog.postman.com/intro-to-apis-history-of-apis/> , 2019
- [12] SQLite extension: <https://www.sqlite.org/index.html>, 2022
- [13] Twitter API manuals: <https://developer.twitter.com/en/docs/twitter-api>

Παράρτημα Α

Σε αυτό το παράρτημα επισυνάπτονται οι οθόνες των λειτουργιών της εφαρμογής μου, μαζί με τον κώδικα που χρησιμοποίησα για την δημιουργία της οθόνης.

Αρχική οθόνη εφαρμογής:



Κώδικας δημιουργίας οθόνης:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MapsActivity">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        app:itemBackground="@color/purple_500"
        app:itemIconTint="@drawable/selector"
        app:itemTextColor="@drawable/selector"
```

```

        app:labelVisibilityMode="labeled"
        android:visibility="invisible"
        app:menu="@menu/navigation_menu" />

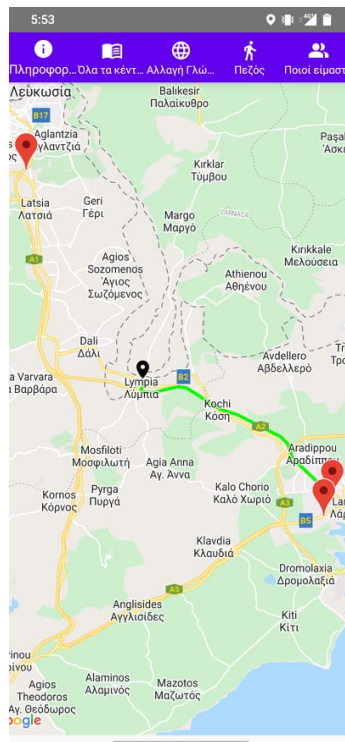
<fragment
    android:id="@+id/location_map"
    class="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="-25dp"
    android:layout_marginTop="1dp" />

<Button
    android:id="@+id/findCenters"
    style="@style/Widget.AppCompat.Button.Small"
    android:layout_width="179dp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="116dp"
    android:layout_marginEnd="116dp"
    android:layout_marginBottom="71dp"
    android:text="Search"
    android:textColor="@color/white" />

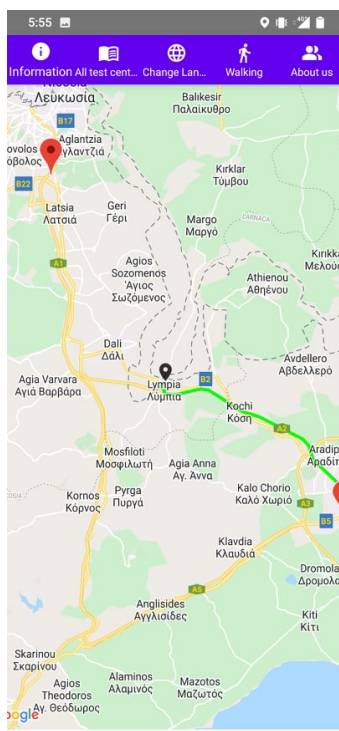
</RelativeLayout>

```

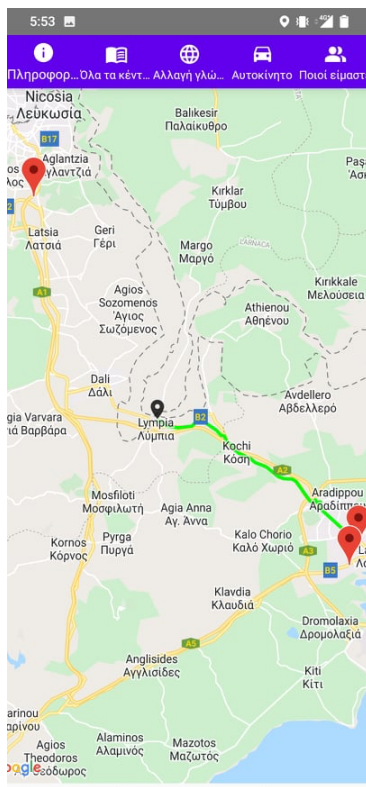
Οθόνη μετά την αναζήτηση για τα σημεία δειγματοληψίας (Ελληνικά):



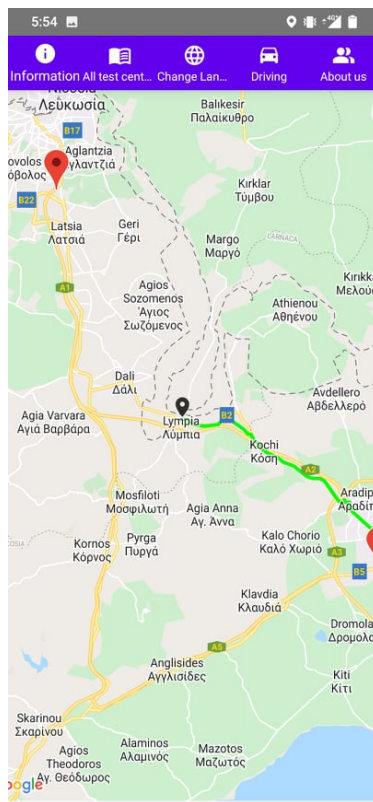
Οθόνη μετά την αναζήτηση για τα σημεία δειγματοληψίας (Αγγλικά):



Οθόνη μετά την αναζήτηση με αλλαγή μέσου μεταφοράς (Ελληνικά):



Οθόνη μετά την αναζήτηση με αλλαγή μέσου μεταφοράς (Αγγλικά):



Οθόνη λειτουργίας «Όλα τα κέντρα» (Ελληνικά):

Όνομα	Ώρες λειτουργίας	Τηλέφωνο	Διάρκεια	Διαθεσιμότητα
Nicosia Mall	9 π.μ. - 6 μ.μ.	94041843		Μη διαθέσιμο
Ιερός Ναός Αγίου Γεωργίου, Λατσιά	9 π.μ. - 6 μ.μ.	99146623		Μη διαθέσιμο
Κοινοτικό Ιατρείο Ιδαίου	9 π.μ. - 6 μ.μ.	94041843		Μη διαθέσιμο
Ιερός Ναός Αγίου Δημητρίου, Ακρόπολη	9 π.μ. - 6 μ.μ.	94041843		Μη διαθέσιμο
Εκκλησία Αγίου Σπυρίδωνα	9 π.μ. - 6 μ.μ.	99146623		Μη διαθέσιμο
Εκκλησία Αγίου Πολυδώρου, προαύλιο, Καϊμακλί	9 π.μ. - 6 μ.μ.	99146623		Μη διαθέσιμο
Προαύλιο Αμφιθεάτρου Λακατάμις, πίσω από Πινακοθήκη	9 π.μ. - 6 μ.μ.	94041843	27 mins	Διαθέσιμο
Εμπορικό και Ψυχαγωγικό Κέντρο Neo Plaza, Κοκκινότριμιθια	2 μ.μ. - 6 μ.μ.	99146623	30 mins	Διαθέσιμο
Κέντρο Υγείας Αγλαντζιάς	9 π.μ. - 6 μ.μ.	99146623		Μη διαθέσιμο
Προαύλιο Δημαρχείου Αγίου Δομετίου	9 π.μ. - 6 μ.μ.	94041843		Μη διαθέσιμο
Ευρωπαϊκό Πανεπιστήμιο Κύπρου	9 π.μ. - 6 μ.μ.	99146623		Μη διαθέσιμο

Οθόνη λειτουργίας «Όλα τα κέντρα» (Αγγλικά):

Name	Operating Hours	Telephone	Trip Duration	Availability
Nicosia Mall	9 am - 6 pm	94041843		Unavailable
Agios Georgios Church, Latsia	9 am - 6 pm	99146623		Unavailable
Dali Community Clinic	9 am - 6 pm	94041843		Unavailable
Agios Demetrios Church, Akropolis	9 am - 6 pm	94041843		Unavailable
Agios Spyridonas Church, Lefkosa	9 am - 6 pm	99146623		Unavailable
Agios Polydorus Church, courtyard, Kaimakli	9 am - 6 pm	99146623		Unavailable
Lakatamia Amphitheatre courtyard, behind Municipal Gallery	9 am - 6 pm	94041843		Unavailable
Neo Plaza Retail and Fun Park, Kokkinotrimithia	2 pm - 6 pm	99146623	30 mins	Available
Aglantzia Healthcare Centre	9 am - 6 pm	99146623		Unavailable
Agios Dometios Municipality	9 am - 6 pm	94041843		Unavailable
European University Cyprus	9 am - 6 pm	99146623		Unavailable

Κώδικας δημιουργίας οθόνης:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation_details"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        app:itemBackground="@color/purple_500"
        app:itemIconTint="@drawable/selector"
        app:itemTextColor="@drawable/selector"
        app:labelVisibilityMode="labeled"
        android:visibility="visible"
        app:menu="@menu/details_menu" />

    <ScrollView
        android:id="@+id/scrollViewDetails"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="0dp">

        <HorizontalScrollView
            android:id="@+id/horizScrollView"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">

            <RelativeLayout
                android:id="@+id/relativeLayout_details"
                android:layout_width="wrap_content"
                android:layout_height="fill_parent"
                android:layout_gravity="center">

                <TableLayout
                    android:id="@+id/table_main"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_alignParentLeft="true"
                    android:layout_centerHorizontal="true"
                    android:layout_marginTop="60dp" />

            </RelativeLayout>
        </HorizontalScrollView>
    </ScrollView>

</RelativeLayout>
```

Οθόνη λειτουργίας «Πληροφορίες» (Ελληνικά):

5:53

Στη βάση των αποφάσεων του Υπουργικού Συμβουλίου, και με στόχο την αναχαίτιση της διασποράς του ιού SARS-CoV-2, στους χώρους όπου η πρόσβαση προϋποθέτει την κατοχή SafePass, τα άτομα ηλικίας 6 ετών και άνω πρέπει να πληρούν μια από τις πιο κάτω προϋποθέσεις:

1. Πιστοποιητικό πλήρους εμβολιασμού.
2. Αποδεικτικό ότι ένα άτομο έχει νοσήσει από την ασθένεια COVID-19 τους τελευταίους 3 μήνες.
3. Πιστοποιητικό αρνητικού rapid test με ισχύ 7 ημερών για άτομα 6-11 ετών.
4. Πιστοποιητικό αρνητικού PCR ή rapid test με ισχύ 72 ωρών για άτομα 12-17 ετών.
5. Πιστοποιητικό αρνητικού τεστ, PCR με ισχύ 72 ωρών ή rapid test με ισχύ 48 ωρών για άτομα 18 ετών και άνω, και τουλάχιστον μια δόση εμβολίου εκεί όπου απαιτείται βάσει Διατάγματος.
6. Πιστοποιητικό αρνητικού rapid test με ισχύ 24 ωρών για άτομα 12 ετών και άνω εκεί όπου απαιτείται βάσει Διατάγματος.

Σύμφωνα με την απόφαση του Υπουργικού Συμβουλίου, στις κινητές μονάδες rapid test του Υπουργείου Υγείας δύνανται να εξυπηρετούνται μόνο οι πιο κάτω κατηγορίες πολιτών:

- Τα ανήλικα πρόσωπα μέχρι 17 ετών. Τα παιδιά ηλικίας 12-17 ετών, τα οποία δεν έχουν τη συγκατάθεση των γονέων/νόμιμων κηδεμόνων τους για εμβολιασμό, δύνανται να διενεργούν δωρεάν rapid test μέσω του προγράμματος του Υπουργείου Υγείας.
- Τα άτομα ηλικίας 18 ετών και άνω, τα οποία δεν μπορούν να υποβληθούν σε εμβολιασμό με τεκμηρίωση με ιατρικό πιστοποιητικό, και αφού το αίτημά τους εξεταστεί από τον Παγκύπριο Ιατρικό Σύλλογο και το Υπουργείο Υγείας και δοθεί σχετική έγκριση, η οποία πρέπει να παρουσιάζεται στη μονάδα δειγματοληψίας. Στην εν λόγω κατηγορία εμπίπτουν και οι εγκυμονούσες, νοούμενου ότι έχουν εξασφαλίσει έγκριση από την ad hoc επιτροπή του ΠΙΣ.
- Τα άτομα που εμβολιάζονται με την 1η δόση του εμβολίου, για την περίοδο που απαιτείται μεταξύ της 1ης και της 2ης δόσης του εμβολίου. Για σκοπούς επαλήθευσης θα πρέπει να παρουσιάζεται η Κάρτα Εμβολιασμού.
- Τα άτομα που έχουν ολοκληρώσει το εμβολιαστικό τους σχήμα και έχει παρέλθει η περίοδος 7 μηνών χωρίς να λάβουν ενισχυτική δόση, μέχρι νεωτέρας. Για σκοπούς επαλήθευσης θα πρέπει να παρουσιάζεται η Κάρτα Εμβολιασμού.
- Τα άτομα που έχουν λάβει την 1η δόση του εμβολίου, αλλά επιθυμούν να υποβληθούν σε τεστ για σκοπούς παρακολούθησης της υγείας τους ή για άλλους λόγους. Νοείται ότι θα πρέπει απαραίτητα να προσκομίζεται η

Οθόνη λειτουργίας «Πληροφορίες» (Αγγλικά)

5:55

Based on the decisions of the Council of Ministers, and aiming at the suppression of the dispersion of the virus SARS-CoV-2, in places where SafePass is required, individuals aged 6 and above must hold one of the following:

1. Certificate of full vaccination.
2. Proof that they have been infected with COVID-19 during the last three months.
3. Certificate of a negative rapid antigen test, valid for 7 days, for individuals aged 6-11 years old.
4. Certificate of a negative PCR or rapid antigen test, valid for 72 hours, for individuals aged 12-17 years old.
5. Certificate of a negative PCR test, valid for 72 hours, or rapid antigen test, valid for 48 hours, for individuals aged 18 and over, and at least one dose of the vaccine where required by Decree.
6. Certificate of a negative rapid antigen test, valid for 24 hours, for individuals aged 12 and over, where required by Decree.

According to the Ministerial Decision, only the following groups of citizens may be served through the mobile rapid antigen testing units:

- Minors up to 17 years old. It is understood that children aged 12-17 who do not have the consent of their parents/legal guardians for vaccination may carry out a rapid test free of charge through the programme of the Ministry of Health.
- Persons aged 18 years or over who cannot be vaccinated, provided they possess a medical certificate, and after their request has been examined by the Cyprus Medical Association and the Ministry of Health and has been approved. The approval must be presented to the testing unit. Pregnant women fall in this category, provided they possess a relevant approval from the ad hoc committee of the Cyprus Medical Association.
- Persons who have been vaccinated with the 1st dose of the vaccine, only for the period required between the 1st and 2nd dose of the vaccine. For verification purposes the Vaccination Card must be presented.
- Persons who have completed their vaccination scheme and the period of 7 months has elapsed without receiving the booster dose, until further notice. For verification purposes the Vaccination Card must be presented.
- Persons who received the booster dose of the vaccine, but wish to take tests for health monitoring purposes or for other reasons. It is understood that the Vaccination Card must be presented.
- Persons holding a COVID-19 recovery certificate within the last 90 days but wish to take tests for health monitoring purposes or for other reasons. It is understood that the

Κώδικας δημιουργίας οθόνης:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button_back_info"
        android:layout_width="90dp"
        android:layout_height="44dp"
        android:text="Back"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.943" />

    <ScrollView
        android:id="@+id/text_info"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.492"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.219" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Οθόνη λειτουργίας «Ποιοι είμαστε»



Κώδικας δημιουργίας οθόνης:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/about_us_image"
        android:layout_width="380dp"
        android:layout_height="240dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.516"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.032"
        app:srcCompat="@drawable/ucy" />

    <TextView
        android:id="@+id/about_us_text"
        android:layout_width="246dp"
        android:layout_height="256dp"
        android:text="TextView"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.597" />

    <Button
        android:id="@+id/about_us_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Back"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.935" />
</androidx.constraintlayout.widget.ConstraintLayout>
```