ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

Έξυπνη χωροθέτηση οχημάτων σε υπόγειο χώρο στάθμευσης μέσω συστημάτων IoT: Μία πρακτική προσέγγιση

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Φώτιος Γιαννουγλίδης(ΑΜ: 3121140)

Επιβλέπων: Ξενάκης Απόστολος/Επίκουρος Καθηγητής

ΛΑΡΙΣΑ, Μάρτιος,2024

UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

Intelligent vehicle parking in underground parking through IoT systems: A practical approach

BACHELOR THESIS

Fotios Giannouglidis (RN: 3121140)

Supervisor:*Xenakis Apostolos, Assistant Professor*

LARISSA, March,2024

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

Φώτης Γιαννουγλίδης

28/8/2025

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/	**ΞΕΝΆΚΗΣ ΑΠΌΣΤΟΛΟΣ** Επίκουρος Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	**ΧΑΙΚΆΛΗΣ ΚΩΝΣΤΑΝΤΊΝΟΣ** Αναπληρωτής Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	**ΚΟΣΜΆΝΟΣ ΔΗΜΉΤΡΙΟΣ** Ακαδημαϊκός Υπότροφος, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: dd-mm-yyyy

Περίληψη

—Η αποτελεσματική διαχείριση μιας επιχείρησης μπορεί να αποδειχθεί τιτάνιος άθλος μέχρι και σε έμπειρα αφεντικά. Ο ιδιοκτήτης πρέπει είτε να διαχειρίζεται ταυτόχρονα όλα τα κομμάτια της επιχείρησής του-κάτι που κοστίζει χρόνο- είτε να προσλάβει κάποιον που εμπιστεύεται για να μοιράσει υποχρεώσεις μαζί του, κάτι που κοστίζει χρήμα και δεν είναι πάντα αξιόπιστη μέθοδος. Εάν όμως υπήρχε κάποιος ο οποίος θα μπορούσε να τα κάνει όλα αυτά σχεδόν δωρεάν χωρίς την περεταίρω καταπόνηση του ιδιοκτήτη? Η εργασία επιδιώκει να λύσει το παραπάνω πρόβλημα την κατασκευή μιας εφαρμογής διαχείρισης κλειστού χώρου στάθμευσης και ενός συστήματος παρακολούθησης κατάστασης χώρου με χρήση αισθητήρων και μικροελεγκτών. Η εφαρμογή θα επιβλέπει τον χώρο και τις συσκευές που τον απαρτίζουν επικοινωνώντας με τους αισθητήρες του μικροελεγκτή, θα εξυπηρετεί τον πελάτη κρατώντας τα στοιχεία του σε βάση δεδομένων και θα βοηθάει τον ιδιοκτήτη προς την διαχείρισή του. Σε αυτό το έγγραφο θα ερευνηθεί το παρελθόν της πλατφόρμας Android στην οποία θα αναπτυχθεί η εφαρμογή καθώς και το πώς συνδέεται με το Διαδίκτυο των πραγμάτων. Θα επεξηγηθεί η εκτενώς η κατασκευή αυτής, της βάσης δεδομένων αλλά και του συστήματος παρακολούθησης λειτουργίας κλειστού χώρου καθώς και οι μέθοδοι που χρησιμοποιήθηκαν για να τελειοποιηθεί το έργο. Στο τέλος θα παρουσιαστεί η λειτουργία της εφαρμογής με φωτογραφίες αλλά και ζωντανά στο κοινό. Αφού αναλυθούν τα παραπάνω το έργο θα αξιολογηθεί και θα καθοριστεί αν άξιζε η υλοποίηση του.

Abstract

Περίληψη στα Αγγλικά.

Abstract—Effectively managing a business can be a titanic feat even for experienced of supervisors. The owner must either manage all the parts of his business simultaneously - something that costs time - or hire someone he trusts to share with him, something that costs money and is not always a reliable method. But what if there was someone who could do all this almost for free without further straining on the part of the owner? This work seeks to solve the above problem with the construction of an indoor parking lot management and a space status monitoring system using sensors and microcontrollers. The application that will supervise the space and the devices that make up it by communicating with the microcontroller's sensors, it will serve the customer by keeping their details in a database and will help the owner in managing it. In this paper, the background of the Android platform on which the application will be developed as well as how it is connected to the Internet of Things will be investigated. The construction of the aforementioned app, the database and the indoor operation monitoring system will be explained in detail, as well as the methods used to perfect the project. In the end, its operation will be presented with photographs and live to the public. After analyzing the above, the project will be evaluated and it will be determined whether its implementation was worth it.

Ευχαριστίες

Θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη στον Απόστολο Ξενάκη για την πολύτιμη καθοδήγηση, την υπομονή και τη συνεχή υποστήριξη καθ' όλη τη διάρκεια της εκπόνησης της παρούσας πτυχιακής εργασίας. Η συμβολή του ήταν καθοριστική για την ολοκλήρωση του έργου αυτού. Για την κατασκευή των πλακετών ήταν αναπόσπαστη η αμέριστη βοήθεια του Παναγιώτη Γιαννουγλίδη. Για τον σχεδιασμό και την υλοποίηση του οδοστρώματος της μακέτας έχει τα θερμά ευχαριστώ μου η Ελένη Γιαννουγλίδου. Τέλος στέλνω ιδιαίτερες ευχαριστίες στην οικογένεια μου για την συνεχή στήριξή και ενθάρρυνση κατά την διάρκεια των σπουδών μου.

Φώτης Γιαννουγλίδης

28/8/2025

Περιεχόμενα

1 Εισαγωγή

Η διαχείριση οποιουδήποτε κλειστού χώρου στάθμευσης απαιτεί την συλλογική συμμετοχή κατόχου και υπαλλήλων για την συνεχή λειτουργία του. Ο κάτοχος θα πρέπει να καταμετρά τα έξοδα λειτουργίας και τα έσοδα που του αποφέρει η επιχείρησή του ενώ ταυτόχρονα θα πρέπει να επιβλέπει τυχόν επισκευές σε βλάβες προσωπικά, παράλληλα με την διαχείριση του προσωπικού του.

Οι υπάλληλοι θα είχαν αρμοδιότητες όπως η φύλαξη και η καθαριότητα του χώρου, επίβλεψη οποιωνδήποτε ηλεκτρονικών συστημάτων, (π.χ μπάρες εισόδου-εξόδου, κάμερες, φωτεινοί σηματοδότες) και την εξυπηρέτηση των υποψήφιων πελατών. Όλες αυτές οι λειτουργίες αλλά και επιπλέον, όπως έλεγχος ποιότητας του αέρα, υγρασίας και θερμοκρασίας θα μπορούσαν να εκτελούνται αυτόματα ή από απόσταση με τον ιδιοκτήτη να επενέβαινε μόνο σε περίπτωση ανάγκης.

Αυτό που θα το καθιστούσε εφικτό αυτό θα ήταν ένα σύνολο αισθητήρων συνδεδεμένα με έναν μικροελεγκτή, που με την σειρά του θα επικοινωνούσε με μία εφαρμογή κινητού. Ο μικροελεγκτής θα είχε την ευθύνη της επίβλεψης των συστημάτων και αισθητήρων που θα ήταν συνδεδεμένα σε αυτόν και θα επέστρεφε μήνυμά στον διαχειριστή σε περίπτωση βλάβης.

Ο χρήστης θα έκλεινε το εισιτήριό του για συγκεκριμένη ώρα και θέση, θα πλήρωνε μέσω της εφαρμογής και κατά την είσοδό του στον χώρο, θα καθοδηγούνταν από φωτεινούς σηματοδότες (και από την εφαρμογή) ελεγχόμενους από τον μικροελεγκτή στην καθορισμένη του θέση. Αν επιθυμούσε ο ιδιοκτήτης να κάνει τον χώρο ακόμα πιο αυτόνομο, θα

έβαζε και ένα ή περισσότερα ρομπότ που θα καθάριζαν τον χώρο αυτόματα συγκεκριμένες ώρες ελεγχόμενα από την εφαρμογή κινητού.

Τέλος, ένας υπάλληλος (έναντι τριών ή παραπάνω) θα είχε την ευθύνη παρακολούθησης του χώρου με κάμερες σε περίπτωση παραβίασης.

Πρώτος στόχος της παρούσας εργασίας είναι η υλοποίηση κομματιού της παραπάνω περιγραφής και η δημιουργία μιας εφαρμογής η οποία θα δίνει την δυνατότητα- σε όποιον

έχει δικαιώματα διαχειριστή-να παρακολουθεί την κατάσταση, ασφάλεια και απόδοση του χώρου από την άνεση την οικείας του. Ως αποτέλεσμα, ο χώρος θα γίνονταν και πιο αποδοτικός

αυξάνοντας τις απολαβές, αλλά και πιο φτηνός στην συντήρησή του. Εφόσον υπάρχει μόνο ένας σταθερός υπάλληλος για την ασφάλεια του χώρου, πέρα από τον μισθό του θα χρειάζονταν να γίνεται μόνο περιστασιακή συντήρηση των συστημάτων που αναφέρθηκαν παραπάνω.

Δεύτερος στόχος της εργασίας είναι η κατασκευή μίας διαδραστικής, εύχρηστης και γρήγορης εφαρμογής για την χρήση της από τον υποψήφιο πελάτη ο οποίος θα ψάχνει χώρο στάθμευσης μέσα στην οποιαδήποτε πόλη στην οποία θα έχει ο ιδιοκτήτης τις εγκαταστάσεις του .

Τρίτος στόχος είναι ο σχεδιασμός μιας βάσης δεδομένων η οποία θα κρατάει λογαριασμούς και πληροφορίες για τους πελάτες. Ταυτόχρονα θα προσομοιώνει την ύπαρξη τραπεζικού λογαριασμού για την πραγματοποίηση συναλλαγών σε συνδυασμό με την φιλοξενία συστήματος επιβράβευσης πόντων με εκπτώσεις για επιμελής πελάτες.

Τελευταίος στόχος είναι η υλοποίηση του πρακτικού κομματιού με την χρήση μικροελεγκτών και αισθητήρων .

Πέρα από τα παραπάνω ένας ακόμα περιορισμός θα είναι η προσθήκη κινούμενων εικόνων και η προσαρμογή της εφαρμογής στα επίπεδα μιας επαγγελματικής πλατφόρμας που θα χρησιμοποιούνταν από μία επώνυμη εταιρεία

2 Ιστορική Αναδρομή

Η ενότητα αυτή θα επικεντρωθεί στην ιστορία των κύριων θεμάτων των τα οποία εξερευνά η εργασία. Αποτελούνται από το λειτουργικό σύστημα Android που θα επεκταθεί στο κεφάλαιο 2.1 και το διαδίκτυο των πραγμάτων [4] στο κεφάλαιο 2.2..

2.1 Android

Το Android ως εταιρεία δημιουργήθηκε το 2003 στο

Πάλτο Άλτο της Καλιφόρνιας από τους Andy Rubin, Rich Miner, Nick Sears και Chris White [26,27]. Ο αρχικός τους σκοπός ήταν η δημιουργία λογισμικού για χρήση του σε ψηφιακές κάμερες, ιδέα την οποία παρουσίασαν στους τότε υποψήφιους επενδυτές το 2004[16]. Αφού όμως διαπιστώθηκε ότι η αγορά ψηφιακών φωτογραφικών μηχανών δεν ήταν επαρκής για τις φιλοδοξίες της σαν εταιρεία, αποφασίστηκε η προώθηση του Android σαν ένα λειτουργικό για χρήση σε κινητές συσκευές με σκοπό τον ανταγωνισμό των τότε πιο διαδεδομένων Symbian και Windows Mobile [28,29].

Λόγω της μη επαρκής συμμετοχή επενδυτών εκείνη την περίοδο, ο Steve Perlman- θέλοντας να βοηθήσει τον φίλο του Rubin και πιστεύοντας στις επιχειρηματικές του επιδιώξεις-έστειλε σε αυτόν 10000 δολάρια σε ένα γράμμα για να ενισχύσει την εταιρεία αρκετά ώστε να αποφύγει την χρεοκοπία [30,31]. Το 2005 έγιναν συμφωνίες από μέρους της εταιρίας με την Samsung [32] και την HTC [33], με την google να την εξαγοράζει αμέσως μετά τον Ιούλιο [27,34].

Οι ιδρυτές της Android μεταβιβάστηκαν στην google μαζί με την εταιρεία τους ξεκινώντας δουλειά για την κατασκευή κινητών τηλεφώνων. Από την υπηρεσία τους δημιουργήθηκε μια κινητή συσκευή με βάση το λειτουργικό Linux Kernel. Το ίδιο, παρουσιάστηκε από την Google στους κατασκευαστές ακουστικών και παροχών κινητής τηλεφωνίας ως ένα ευέλικτο και αναβαθμίσιμο σύστημα [35] για το οποίο είχαν ήδη ετοιμάσει hardware αλλά και software με τις ανάλογες επιχειρηματικές συμφωνίες [36].

Τον Δεκέμβριο του 2006 εμφανίστηκαν πρωτότυπα του κινητού BlackBerry με πληκτρολόγιο σε κατάταξη QWERTY και δίχως οθόνη αφής [37]. Ωστόσο η apple το 2007 με την ανακοίνωση του πρώτου iphone κέρδισε τα κοινά κάνοντας την google να

επενδύσει και αυτή στις οθόνες αφής [38,39]. Μετά από μερικούς μήνες η nokia και Blackberry εμφάνισαν τα έξυπνα κινητά τους τηλέφωνα με οθόνες αφής για να ανταγωνιστούν το τότε iphone 3G. Αυτό αποτέλεσε μεγάλο βήμα στην αλλαγή της πολιτικής του android προς την χρήση οθονών αφής στις κινητές του συσκευές με το πρώτο εμπορικώς διαθέσιμο smartphone να φτάνει στα ράφια στις 23 Σεπτεμβρίου του 2008 [40,41].

Το 2007 ένα σύνολο από εταιρίες τεχνολογίας συμπεριλαμβανόμενων των google, HTC, Motorola, Samsung και άλλες όπως Qualcomm και Texas Instruments ανακοίνωσε τα σχέδια του για την ανάπτυξη μιας ανοιχτής και ευκολομεταχείριστης βάσης για το λειτουργικό android[42][43][44]. Ωστόσο σύντομα το σύνολο θα είχε και ανταγωνισμό, από την Symbian Foundation και την LiMo Foundation, η οποία μάλιστα έφτιαχνε λειτουργικό ανοιχτού κώδικα με βάση τα linux για κινητές συσκευές[45][46]. Τον Σεπτέμβριο του 2008 ανακοινώνεται το android για πρώτη φορά στον καταναλωτή σε μια συνέντευξη τύπου στο μετρό της Νέας Υόρκης από τους Andy Rubin, Larry Page, Sergey Brin, Cole Broadman, Christopher Schlaeffer και Peter Chou [47].

Από το 2008 μέχρι σήμερα, το λειτουργικό έχει λάβει αναρίθμητες ενημερώσεις οι οποίες έχουν προσθέσει καινούρια χαρακτηριστικά και έχουν διορθώσει τυχόν λάθη στον κώδικα. Κάθε μεγάλη έκδοση ονομάζεται με αλφαβητική σειρά σύμφωνα με ένα γλυκό(π.χ KitKat, Lollipop η Marshmallow). Με την ανακοίνωση του KitKat το 2013 η Google εξήγησε ότι ο λόγος για αυτές τις ονομασίες είναι ότι "κάθε συσκευή κάνει την ζωή μας λίγο πιο γλυκιά, άρα αρμόζει να τα ονομάζουμε ανάλογα" [48].

Το 2010 έφτασαν στην αγορά η σειρά συσκευών Nexus από την Google, με την συνεργασία της με διάφορους παραγωγούς. Η σειρά λέγονταν ότι έπαιξε ιδιαίτερα σημαντικό ρόλο στην ιστορία του Android εισάγοντας καινούριες εκδόσεις λειτουργικού και θέτοντας καινούρια πρότυπα στο υλικό και τις δυνατότητες των συσκευών που το έφεραν [49](Εκδόσεις Android ανά τα χρόνια).

Ο Hugo Barra εξυπηρέτησε την Google σαν εκπρόσωπος τύπου από το 2008 μέχρι την αποχώρηση του πηγαίνοντας το 2013 με την κινέζικη εταιρεία παραγωγής τηλεφώνων Xiaomi [50,51]. Ύστερα από έξι μήνες, ο διευθύνων σύμβουλος Larry Page ανακοίνωσε την μετάθεση του Andy Rubin από το τμήμα αφοσιωμένο στο Android για να ασχοληθεί με καινούρια έργα στην Google, με τον Sundar Pichai να παίρνει την θέση του [52,53]. Τον Αύγουστο ο προαναφερόμενος θα γίνονταν ο καινούριος διευθύνων σύμβουλος της

Google [54,55] με τον Hiroshi Lockheimer να αναλαμβάνει το τμήμα του Android. [56,57]

Το λειτουργικό ενεπλάκη στον εμπορικό πόλεμο μεταξύ Κίνας και Ηνωμένων Πολιτειών το Μάιο του 2019, βάζοντας την εταιρεία Huawei σε δύσκολη θέση με την εξάρτησή τους από το Android. [58,59] Το καλοκαίρι του 2019 ανακοινώθηκε από την Huawei η ανάπτυξη ενός καινούριου λειτουργικού [60] του Harmony OS σαν εναλλακτική του Android [61] υποβάλλοντας αίτημα για τα πνευματικά της δικαιώματα στην παγκόσμια αγορά. [62,63] Λόγω της πίεσης των κυρώσεων οργανώθηκε σχέδιο για την αντικατάσταση του android μέχρι το 2022 με άλλο λειτουργικό σύστημα αφού το Harmony OS προορίζονταν για συσκευές διαδικτύου των πραγμάτων παρά κινητές συσκευές [64]. Το 2021 αναφέρθηκε από ορισμένους χρήστες ότι δεν μπορούσαν να καλέσουν αριθμούς εκτάκτου ανάγκης [65,66]. Το πρόβλημα οφείλονταν σε έναν συνδυασμό λαθών στον κώδικα του Android και στην εφαρμογή Microsoft Teams. Οι δύο εταιρείες επιλύσαν το πρόβλημα με αντίστοιχες ενημερώσεις σύντομα μετά την εμφάνισή του.

Εκδόσεις Android ανά τα χρόνια

2.2 Internet Of Things

Το Διαδίκτυο των πραγμάτων (IoT) είναι μια έννοια στην οποία τα φυσικά αντικείμενα ή «πράγματα» που επικοινωνούν μέσω του Διαδικτύου, επιτρέποντας την επικοινωνία μεταξύ συσκευών και εφαρμογών που είναι συνδεδεμένες στο Διαδίκτυο.[66] Το 1982 [7] ένα πρωτότυπο δίκτυο συνδεδεμένο σε μία έξυπνη συσκευή η οποία έλεγχε με την σειρά της ένα σύνολο από αισθητήρες συνδεδεμένους στο internet δημιουργήθηκε στο Πανεπιστήμιο Επιστήμης και Υπολογιστών του Carnegie. Η

συσκευή κατέληγε σε έναν αυτόματο πωλητή Coca cola και έπαιρνε δεδομένα για την θερμοκρασία του πωλητή ενώ κατέγραφε και το εμπόρευμα που υπήρχε εντός σε πραγματικό [8,9] χρόνο κάνοντας την την πρώτη ARPANET [11,12] συνδεδεμένη συσκευή. Η έμπνευση για αυτήν την εφεύρεση ήταν ο αυτόματος πωλητής χειριζόμενος από έναν υπολογιστή στα ενδότερα του εργαστηρίου τεχνητής νοημοσύνης του Standford [10].

Η εργασία του Mark Weiser το 1991 "Ο υπολογιστής του 21ου αιώνα" σε συνδυασμό με ακαδημαϊκές συνεδρίες όπως το Ubicomp και το Percom έδωσαν ζωή στην ιδέα της διασύνδεσης συνεννοουμένων συσκευών μέσω διαδικτύου [13,14]. Το 1994 η γενική ιδέα περιγράφτηκε από τον Reza Raji ως μετακίνηση μικρών πακέτων δεδομένων σε πολλούς κόμβους με σκοπό την ενσωμάτωση και την αυτοματοποίηση των πάντων από οικιακές συσκευές μέχρι μεγάλα εργοστάσια [15].

Μεταξύ του 1993 και 1997 προτάθηκαν πολλές λύσεις στο πρόβλημα συμπεριλαμβανομένου του at Work της Microsoft και το Novel της NEST. Ο τομέας άρχισε να επιταχύνει με την άφιξη της τεχνολογίας επικοινωνίας μεταξύ συσκευών από τον Bill Joy το 1999 η οποία παρουσιάστηκε στο Παγκόσμιο Οικονομικό Φόρουμ στο Νταβός [16].

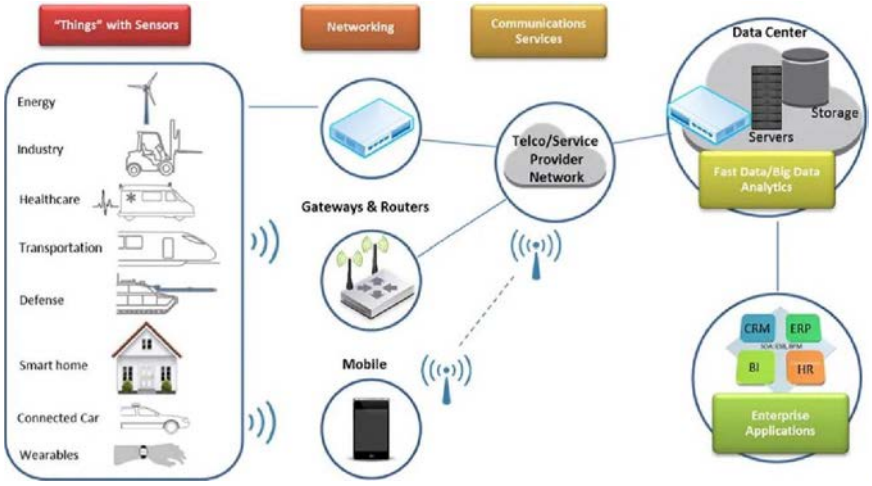
Η έννοια του Διαδικτύου των Πραγμάτων πρωτοεμφανίστηκε σε ομιλία του Peter T. Lewis στο 15ο Ετήσιο Νομοθετικό Σαββατοκύριακο του κογκρέσου Black Caurus Foundation στην Ουάσινγκτον D.C που εκδόθηκε τον Σεπτέμβρη του 1985. Σύμφωνα με τον Lewis το διαδίκτυο των πραγμάτων είναι η ενσωμάτωση διεργασιών και τεχνολογίας στην ζωή του ανθρώπου, τα οποία με αισθητήρες επιτρέπουν παρακολούθηση από απόσταση της κατάστασης τους,

και της διαχείρισής τους [17].

Ο όρος Διαδίκτυο των Πραγμάτων εφευρέθηκε από τον Kevin Ashton του κέντρου αυτόματης αναγνώρισης του MIT το 1999 [18]. Ο ίδιος πίστευε ότι η αναγνώριση ραδιοσυχνοτήτων σαν τεχνολογία ήταν αναγκαία για την υλοποίηση της ιδέας[19] επιτρέποντας σε κάθε υπολογιστή να διαχειρίζεται ξεχωριστά οποιοδήποτε αντικείμενο [20,22]. Το κύριο θέμα του διαδικτύου των πραγμάτων ήταν η εγκατάσταση αναμεταδοτών μικρής εμβέλειας σε διάφορες συσκευές καθημερινής χρήσης επιτρέποντας την καλύτερη επικοινωνία μεταξύ ανθρώπων-πραγμάτων και μεταξύ των πραγμάτων των ίδιων [23].

Το 2004 ο διευθύνων σύμβουλος της NetSilicon Cornelius Pete peterson είπε ότι η επόμενη γενιά της πληροφορικής θα ασχολείται σε μεγάλο ποσοστό με το διαδίκτυο των πραγμάτων και τις διασυνδεδεμένες συσκευές που το διέπουν. Μάλιστα προέβλεψε ότι θα γίνει σε τέτοιο βαθμό που οι συνδεόμενες συσκευές στο διαδίκτυο θα ξεπεράσουν σε αριθμό τους προσωπικούς και επαγγελματικούς υπολογιστές. Μια τέτοια εποχή ο διευθυντής πίστευε ότι θα επικροτούνταν από συσκευές στον τομέα της υγείας και της βιομηχανίας [24].

Η Cisco Systems ,παίρνοντας σαν ορισμό του Διαδικτύου των Πραγμάτων το σημείο στην ιστορία που είχε περισσότερα πράγματα συνδεδεμένα στο διαδίκτυο από ότι ανθρώπους, εκτιμήθηκε ότι η εμφάνισή του σε μεγάλη κλίμακα έγινε μέσα στο 2008 με 2009 με τον λόγο πραγμάτων- ανθρώπων να μεγαλώνει από 0.08 το 2003 σε 1.84 το 2010 [25].ι τίτλοι θα πρέπει να είναι μικροί σε μήκος (να χωράνε σε μία σειρά) και περιεκτικοί. Για τη μορφοποίηση υπάρχουν τα στυλ επικεφαλίδων. Σε εκτύπωση διπλής όψης, το πρότυπο είναι ρυθμισμένο έτσι ώστε τα κεφάλαια να αρχίζουν όλα σε δεξιά σελίδα. Μη βάζετε Αγγλική ορολογία στους τίτλους, εκτός από εξαιρετικές περιπτώσεις (ρωτήστε τον επιβλέποντα). Αν θέλετε να βάλετε την μετάφραση κάποιου όρου, κάντε το μέσα στο κείμενο που ακολουθεί, στην πρώτη ευκαιρία.



Παράδειγμα συστήματος IoT

2.3 SQL

Οι Donald D. Chamberlin και Raymond F. Boyce ήταν οι οι πατέρες της γλώσσας SQL αναπτύσοντάς την στην IBM το 1970 βασισμένοι στο σχεσιακό μοντέλο του Edgar F. Codd[12](Structured Query Language). Ονομαζόμενη SEQUEL στην πρώτη της έκδοση

προορίζονταν για την αποθήκευση και ανάκτηση δεδομένων από την βάση δεδομένων της IBM που είχε αναπτυχθεί απο μία ομάδα το 1973 στο εργαστήριό της San Jose..[13]

Οι Chamberlin και Boyce στην πρώτη προσπάθεια τους να φτιάξουν μια γλώσσα σχεσιακής βάσης δεδομένων έφτιαξαν το SQUARE. Ωστόσο αποδείχθηκε δύσκολη η χρήση της λόγω σημειογραφίας δείκτη/υπέρθετο.

Η ανάπτυξη του διάδοχου της SQUARE συνέχισε στο Σαν Χοσέ το 1973 με όνομα SEQUEL [12]το οποίο ήταν λογοπαίγνιο στην QUEL μια άλλη ευρηματική γλώσσα του Igres.

Για την αποφυγή καταπάτησης πνευματικών δικαιωμάτων το όνομα της γλώσσας άλλαξε σε SQL λόγω της συνωνυμίας με την εταιρείας Hawker Siddeley Dynamics Engineering Limited με έδρα το Ηνωμένο Βασίλειο.[15].

Μετά την αξιολόγηση της SQL σε ιστότοπους δοκιμών πελατών για να διαπιστωθεί η εφαρμοσιμότητα και η χρησιμότητα του συστήματος,

Με βάση το πρωτότυπο System R, η IBM άρχισε να δημιουργεί εμπορικά προϊόντα. Αυτά ήταν τα System/38, SQL/DS και IBM Db2, τα οποία διατέθηκαν για αγορά το 1979, το 1981 και το 1983, αντίστοιχα. [17]

Η Relational Software, Inc γνωστή τώρα ως Oracle Corporation αναγνώρισε τις δυνατότητες των προσπαθειών των Codd, Chamberlin και Boyce και ανέπτυξε το δικό της RDBMS. Βασισμένο στην SQL ο σκοπός του ήταν να πουληθεί στο ναυτικό των ΗΠΑ την CIA και άλλες κυβερνητικές υπηρεσίες. Το 1979 παρουσίασε τον Ιούνιο τις πρώτες εμπορικά διαθέσιμες υλοποιήσεις της SQL, την Oracle V2 (Version2) για υπολογιστές VAX.

Ομάδες προτύπων όπως η ANSI και η ISO υιοθέτησαν επισήμως τον όρο "SQL Language Database" μέχρι το 1986 και νέες εκδόσεις των προτύπων εκδόθηκαν το 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011,

[12] 2016 και πιο πρόσφατα, 2023.[18]

Stuctured Query Language

3 Κατασκευή και Σχεδιασμός Εφαρμογής Android

3.1 Επεξήγηση Λειτουργίας

Η εφαρμογή θα αναπτυχθεί σε περιβάλλον android studio στην γλώσσα java σε API 30 και Android version: 11 Red Velvet Cake. Θα επεξηγηθεί κυρίως το κομμάτι του κώδικα που θα εκτελεί τις λειτουργίες για τις οποίες σχεδιάστηκε(κώδικας java)[3] καθώς η επεξήγηση της προσαρμογής εμφάνισης(XML) δεν θα αναλυθεί σε αυτήν την εργασία και θα παρουσιαστεί μόνο το αποτέλεσμα. Πρώτα θα επεξηγηθεί εκτενώς ο κώδικας της κάθε κλάσης με λόγια και άμα χρειαστεί παραπάνω διευκρίνιση θα παρέχεται απαράλλαχτος σε παράρτημα στο τέλος της εργασίας.

3.1.1 Δραστηριότητα Main Activity

Αρχικά ανακτώνται οι ταυτότητες των Radiobutton rb1 και rb2 με την findViewById κάνοντας το ίδιο και για το Button με το όνομα ”Συνέχεια”. Για να μπορέσει το πρόγραμμα να καταλάβει το πότε πατήθηκε το κουμπί ή όχι χρησιμοποιείται το btn.SetOnClickListener(). Ύστερα ορίζεται η μέθοδος onClick στην xml αλλά και στην Java και ανακτείται η κατάσταση του RadioButton χρησιμοποιώντας το radiobutton.isChecked()[2]. Με μία if σε κάθε περίπτωση γίνεται ανάλογα με το ποιο έχει επιλεγεί να ξεκινήσει μια καινούρια δραστηριότητα με το Intent i =new Intent(v.getContext(),HandleGuest.class) ή με το Intent a =new Intent(v.getContext(),HandleMember.class) για τον επισκέπτη και το Μέλος ανάλογα. Επίσης, κάθε φορά που επιλέγεται ένα radiobutton εμφανίζεται και ένα μήνυμα στο κάτω μέρος της οθόνης. Αυτό επιτυγχάνεται με την χρήση ενός string output. Το string αλλάζει τιμή όταν πατηθεί ένα radiobutton και όχι όταν πατηθεί το κουμπί Συνέχεια. Επομένως, ελέγχοντας ποιο πατήθηκε με το radiobutton.isChecked() αλλάζει και η τιμή του σε ”Είστε επισκέπτης!” η ”Είστε Μέλος!”. Η εντολή Toast.makeText(this,output,Toast.LENGTH_SHORT).show είναι αυτή που εμφανίζει το μήνυμα στην οθόνη.

3.1.2 Δραστηριότητα Handle_Guest

Η κλάση αποτελείται από δύο κομμάτια. Τον κώδικα χειρισμού του κουμπιού ”Πίσω” OnClickBack και την μέθοδο OnClickMember για το κουμπί ”Συνέχεια”. Για την OnClickMember ανακτούμε τις ταυτότητες των πεδίων υποδοχής EditText t1, t2, t3, t4. Ύστερα ορίζουμε ως string αυτά που θα εισάγει ο επισκέπτης δηλαδή το name, car_type, last_name και email. Μέσα σε μία try-catch επιχειρείται η σύνδεση της εφαρμογής στην βάση δεδομένων SQL. Πρώτα καλείται η κλάση ConnectionHelper() και ορίζεται ως connect. Ύστερα εφόσον η σύνδεση είναι επιτυχής δημιουργείται καινούριο Statement με το connect.createStatement(); και εισάγεται στην βάση το query INSERT.INTO.GUESTS.VALUES('"+name+"','"+last_name+"','"+car_type+"','"+email +"','"+1000+"') με την χρήση της Boolean rs=st.execute(sqlinsert);

Ελέγχεται με την χρήση της TextUtils.isEmpty() σαν λογική μέσα σε μία if αν οποιοδήποτε από τα πεδία στα οποία θα εισάγει πληροφορίες ο χρήστης είναι κενά. Εάν έχουν συμπληρωθεί όλα ορίζεται η επόμενη δραστηριότητα στην else με το Intent b = new Intent(this,SeatPick.Class), θέτεται το string guest ως ”yes” , το name ως"GName" και μεταβαίνουν στην επόμενη δραστηριότητα SeatPick με την b.putExtra() η οποία ξεκινάει με το startActivity(b). Τα τελευταία δύο γίνονται για να μπορεί να καταλάβει η εφαρμογή σε ποια δραστηριότητα να επιστρέψει εφόσον ο χρήστης το επιθυμήσει και για να μπορεί να καταλάβει η εφαρμογή από ποιόν να αφαιρέσει χρήματα όταν γίνει η αγορά θέσης.. Για να μπορέσει να πάει πίσω όμως χρειάζεται την OnclickBack η οποία με το πάτημα του κουμπιού ”Πίσω” εκκινεί την προηγούμενη δραστηριότητα με το startActivity(k1) αφού οριστεί από το Intent k1= new Inent(this,MainActivity.Class)

3.1.3 Δραστηριότητα HandleMember

Περικυκλωμένα από μία try_catch, ορίζονται πρώτα τα string member και customer τα οποία θα πάρουν τα ονόματα του μέλους και του επισκέπτη αντιστοίχως ανάλογα με την επιλογή του χρήστη. Ανακτώνται οι ταυτότητες των editTextText editTextPassword και θέτονται πάνω στα string user kai pass . Ορίζεται καινούρια σύνδεση με το όνομα connect και αρχίζει την λειτουργία της η κλάση connectionHelper με την κλήση της από τις εντολές

```
ConnectionHelper connectionHelper = new ConnectionHelper();
connect = connectionHelper.connectionClass();
```

Για τον έλεγχο κενών εισαγωγής χρησιμοποιείται η TextUtils.isEmpty() για τα user και pass και εμφανίζεται μήνυμα "Παρακαλώ γεμίστε όλα τα πεδία." εφόσον ισχύει η συνθήκη if.

Σε άλλη περίπτωση else if, εάν τα συνθηματικά του χρήστη αντιστοιχούνε με αυτά του διαχειριστή διαδικασία η οποία επιτυγχάνεται με τις εντολές:

```
user.equals("admin") && pass.equals("1234")
```

η εφαρμογή θα καταλάβει ότι πρόκειται για τον ιδιοκτήτη θα ορίσει νέο Intent και θα ξεκινήσει την δραστηριότητα Administrator με τις εντολές:

```
Intent admin = new Intent(getApplicationContext(), Administrator.class); startActiv-
ity(admin);
```

Στην περίπτωση που δεν ισχύουν καμία από τις παραπάνω συνθήκες ξεκινάει νέο Intent k, το string member παίρνει την τιμή "yes" και περνάει το status2 με τη τιμή του member στην δραστηριότητα SeatPick. Αυτό γίνεται για να ξέρει η προαναφερθέν δραστηριότητα σε ποια να επιστρέψει σε περίπτωση που ο χρήστης πατήσει το κουμπί Πίσω.

Εφόσον η σύνδεση είναι επιτυχής ορίζεται το string query σαν

```
"SELECT usr_name, pass_word FROM MEMBERS WHERE usr_name = ? AND pass_word = ?"
```

Το παραπάνω ερώτημα ψάχνει στην βάση εάν υπάρχει εγγραφή με τα ίδια ακριβώς στοιχεία . Ύστερα τοποθετούνται στις θέσεις των ερωτηματικών τα στοιχεία user και pass που εισίγαγε ο χρήστης με τις εντολές:

```
st.setString(1, user);
st.setString(2, pass);
```

Για την μεταφορά του παραπάνω ερωτήματος στην βάση δεδομένων χρησιμοποιείται ένα PreparedStatement st και εκτελείται με την ResultSet rs με την εντολή st.executeQuery();.

Εφόσον βρεθεί λογαριασμός στην βάση

```
if(rs.next())
```

περνιέται η μεταβλητή paystatus με την τιμή της customer η οποία θα είναι η ίδια με τον user και ξεκινάει η επόμενη δραστηριότητα.

```
k.putExtra("paystatus", customer);

startActivity(k);
Toast.makeText(HandleMember.this, "Επιτυχία σύνδεσης.", Toast.LENGTH_SHORT).show();
```

Εάν δεν βρεθεί κάποια αντιστοίχιση στην βάση δεδομένων η εφαρμογή εμφανίζει μήνυμα σφάλματος "Αποτυχία σύνδεσης. Λανθασμένος κωδικός ή όνομα χρήστη."

Τέλος σε περίπτωση που η σύνδεση αποτύχει η catch βγάζει μήνυμα στην εφαρμογή και το logcat:

```
catch (Exception ex) {
    Log.e("Error", "Connection error: ", ex);
    Toast.makeText(HandleMember.this, "Αποτυχία σύνδεσης με τον διακομιστή.",
Toast.LENGTH_SHORT).show();

}
```

3.1.4 Δραστηριότητα New_Member

Πρώτα ανακτώνται οι ταυτότητες των πεδίων εισαγωγής button2 , editTextUsername , editTextPass, editTextName, editTextLastName, car, και editTextTextEmailAddress2 έξω από την OnclickNewMember με την χρήση της findViewById(R.id.ExampleText). Ύστερα , μέσα στην OnclickNewMember τους δίνονται συνθηματικά αναγνώρισης user_name, pass_word, first_name, last_name, car_type και email αντίστοιχα , ορίζεται η σύνδεση connect και καλείται η κλάση ConnectionHelper με τις εντολές

```
ConnectionHelper connectionHelper = new ConnectionHelper();
connect = connectionHelper.connectionClass();
```

Εφόσον η σύνδεση είναι επιτυχής μέσα σε μία try-catch ορίζεται το ερώτημα με όνομα query

```
"SELECT usr_name, pass_word FROM MEMBERS WHERE usr_name = ? AND pass_word = ?"
```

για την αναζήτηση συνθηματικών και ορίζεται το PreparedStatement st1. Στις θέσεις των δύο ερωτηματικών μπαίνουν οι τιμές από τα πεδία user_name και pass-word αντίστοιχα με τις :

```
st1.setString(1, user_name);
st1.setString(2, pass_word);
```

και το ερώτημα στέλνεται στην βάση με την εντολή :

```
ResultSet rs = st1.executeQuery();
```

Εφόσον βρεθούν συνθηματικά ίδια με αυτά που εισάχθηκαν η εφαρμογή εμφανίζει μήνυμα στην οθόνη με την εντολή:

```
Toast.makeText(New_Member.this, "Αυτό το όνομα και ο κωδικός υπάρχουν ήδη.",
Toast.LENGTH_SHORT).show();
```

Και κλείνει το statement και το ResultSet με τα¨:

```
rs.close();
st1.close();
```

Αν από την άλλη δεν βρεθούν ίδια συνθηματικά άλλα τα πεδία εισαγωγής είναι κενά (χρήση της TextUtils.isEmpty() μέσα στην if για κάθε πεδίο) η εφαρμογή εμφανίζει μήνυμα με την:

```
Toast.makeText(New_Member.this, "Παρακαλώ γεμίστε όλα τα πεδία.",
Toast.LENGTH_SHORT).show();
```

Ειδάλλως εάν δεν γίνει τίποτε από τα παραπάνω(else) εκτελείται το ερώτημα

```
String sqlinsert = "INSERT INTO MEMBERS VALUES('" + first_name + "','" + last_name +
"','" + user_name + "','" + pass_word + "','" + car_type + "','" + email + "','" + 1000
+ "','" + 1000 + "')";
```

που ορίστηκε λίγο πιο πάνω με το:

```
Statement st = connect .createStatement()
```

και εκτελείται με την:

```
Boolean rs2 = st.execute(sqlinsert);.
```

Ενώ το πρόγραμμα μετακινεί τον χρήστη στην δραστηριότητα HandleMember με τις εντολές :

```
Intent b = new Intent(this, HandleMember.class);
startActivity(b);
```

Σε περίπτωση αποτυχίας σύνδεσης με την βάση δεδομένων δουλεύει η catch και εμφανίζει στην οθόνη το μήνυμα

```
Toast.makeText(New_Member.this, "Αποτυχία σύνδεσης με τον διακομιστή.",
Toast.LENGTH_SHORT).show();
```

Και στο logcat

```
Log.e("Error","Connection error: ",ex);
```

Τέλος εφόσον ο χρήστης θελήσει να πάει πίσω για οποιονδήποτε λόγο η OnclickBack1 ανακτεί το intent k3 και μεταφέρει τον χρήστη στην δραστηριότητα HandleMember.

```
Intent k3 = new Intent(getApplicationContext(), HandleMember.class);
startActivity(k3);
```

3.1.5 Δραστηριότητα Seat_Pick

Η δραστηριότητα SeatPick αποτελείται από την OnClickBack1 την OnClickNavigate και την OnClickAvailabillity.

Η διεργασία OnClickBack1 για να πάει πίσω ο χρήστης είναι ιδιαίτερη στην SeatPick διότι πρέπει να ξεχωρίσει ποια δραστηριότητα να εκκινήσει ανάλογα με το αν ο χρήστης είναι μέλος ή όχι. Αυτό το επιτυγχάνει με τις μεταβλητές Status1 και Status2 οι οποίες περνιούνται από τις δραστηριότητες HandleGuest και HandleMember αντίστοιχα με τα b.getStringExtra(”status1”) και k.getStringExtra(”status2”). Ελέγχοντας με δύο if εάν κάποια απο τις δύο έχει την τιμή ”yes” εκκινεί την ανάλογη δραστηριότητα με τα Intent k2 = new Intent(getApplicationContext(),HandleGuest.class), start Activity(k2) και τα k3 = new Intent(getApplicationContext(),HandleMember.class), start Activity(k3).

Για την OnclickNavigate ανακτούμε τις ταυτότητες time1 time2 και spin με την findViewById(R.id.editTextTime); και τις βάζουμε σε μεταβλητή τύπου String με την

time.getText()toString();. Του spinner η τιμή ανακτάται με την String dir= String.ValueOf(spin.getSelectedItem());.

Για κάθε μία από τις εννέα περιπτώσεις επαναλαμβάνεται η ίδια δήλωση if(dir.equals("Θέση x")). Στην συνέχεια ορίζονται τα String output1, output2, output3, output4. Αρχικοποιείται καινούρια δραστηριότητα pi(;όπου i αριθμός δραστηριότητας NavigationActivity) με το new Intent (getApplicationContext(),NavigationActicvity.class);.

Εμφανίζεται νέο μήνυμα στην οθόνη που χρησιμοποιεί το output1 με την εντολή Toast.makeText(this, output1, Toast.LENGTH_SHORT).show(); και οι μεταβλητές τύπου String μεταφέρονται στην οριστικοποιημένη δραστηριότητα με την εντολή pi.putExtra("line",output2);.

Η OnClickAvailabillity δουλεύει ακριβώς όπως και στην

Δραστηριότητα Administrator με την μικρή διαφορά της ανάκτησης της ταυτότητας και της τιμής της κάθε θέσης με σκοπό την εμφάνιση ενός μηνύματος με το περιεχόμενο "Η " + dir + "\n δέν είναι διαθέσιμη παρακαλώ επιλέξτε κάποια άλλη."; με το dir να μπορούσε να είναι από θέση ένα μέχρι θέση εννέα.

3.1.6 Δραστηριότητα Seat_Pay

Μέσα στην κλάση Seat_Pay ορίζονται τα string len1,len2,len3,time1,time2,seat,customername και GuestName. Ανακτάται το Intent p από την προηγούμενη δραστηριότητα μαζί με τα line1, line2, line3, time1, time2, Seat, customer και Gname τα οποία παίρνουν τα ονόματα των string που αναφέρθηκαν προηγουμένως με την p.getStringExtra.

```
Intent p = getIntent();
len1 = p.getStringExtra("line1");
```

Αμέσως μετά υπάρχει η OnclickBack2 η οποία στέλνει τον χρήστη πίσω στην δραστηριότητα Seat_Pick εφόσον πατήσει το κουμπί ορίζοντας νέο Intent k3 και εκτελώντας το.

```
Intent k3 = new Intent(getApplicationContext(), SeatPick.class);
startActivity(k3);
```

Ύστερα ξεκινάει ο κύριος χειρισστής κουμπιού της δραστηριότητας ο οποίος έχει την δουλειά της αναγνώρισης του χρήστη και της ενημέρωσης της βάσης δεδομένων.

Για τον χρήστη υπάρχουν τέσσερις επιλογές πληρωμής.Με Visa , Mastercard, Paypal, και Revolut. Ανάλογα με τον ποιο τρόπο επιλέξει κερδίζει και τους αντίστοιχους πόντους(x3 Visa, x5 Mastercard x1 Paypal, x3 Revolut) με πενήντα πόντους το κατώτερο κέρδος. Αυτό γίνεται με την χρήση ενός spinner το οποίο κρατά. Εάν ο χρήστης έχει διακόσιους πενήντα πόντους ή περισσότερους κατά την ώρα της πληρωμής έχει έκπτωση πέντε ευρώ από τα δέκα. Μέσα σε ένα try-catch αρχικοποιείται η νέα σύνδεση στην βάση , καλείται η κλάση ConnectionHelper και ορίζεται ως string το ερώτημα sql το οποίο θα ψάξει για χρήστη που έχει πάνω από διακόσιους πενήντα πόντους.

```
ConnectionHelper connectionHelper = new ConnectionHelper();
connect = connectionHelper.connectionClass();
String sqlSELECT ="SELECT points FROM MEMBERS WHERE usr_name=? AND points>=250";
```

Ύστερα εφόσον η σύνδεση είναι επιτυχής εκτελείται για κάθε επιλογή πληρωμής για τον πελάτη η παρακάτω διαδικασία. Το spinner λαμβάνει την τιμή του dir για να καθοδηγήσει την εφαρμογή στις σωστές γραμμές κώδικα.

```
if ("Visa(x2)".equals(dir)) {

    PreparedStatement st = connect.prepareStatement(sqlSELECT);
    st.setString(1,customername);
    ResultSet RS = st.executeQuery();
```

Ορίζεται το PreparedStatement για την εκτέλεση του query και τοποθετούνται μέσα στο query τα στοιχεία του χρήστη που λήφθηκαν από τις προηγούμενες δραστηριότητες. Εφόσον δεν βρεθεί ο χρήστης με όνομα customername εκτελείται το query, ο χρήστης πληρώνει δέκα ευρώ κερδίζοντας τους αντίστοιχους πόντους, κλείνει το prepared statement και μεταφέρεται στην NavigationActivity.

```
PreparedStatement st2 = connect.prepareStatement(sqlinsert2);
rs = st2.executeUpdate();
Toast.makeText(Seat_Pay.this, "Κερδίζετε 100 πόντους!", Toast.LENGTH_SHORT).show();
st2.close();
startActivity(k);
```

Εάν βρεθεί όμως χρήστης με παραπάνω από διακόσιους πενήντα πόντους εκτελείται διαφορετικό query με την ίδια διαδικασία το οποίο αφαιρεί τους πόντους για την έκπτωση και ο χρήστης πληρώνει πέντε ευρώ έναντι των 10.

```
Toast.makeText(Seat_Pay.this, "Κερδίζετε 100 πόντους και 5 ευρώ έκπτωση!",
Toast.LENGTH_SHORT).show();
PreparedStatement st1 = connect.prepareStatement(sqlinsert1);
rs = st1.executeUpdate();
st1.close();
startActivity(k);
```

3.1.7 Δραστηριότητα NavigationActivity

Έξω από την onCreate ανακτείται το intent p από την SeatPay και ταυτότητα του textView8 .

```
Intent p= getIntent();
TextView FinalText=findViewById(R.id.textView8);
```

Τα string που περάστηκαν από την SeatPay line1,line2,line3,time1,time2, και Seat παίρνουν αναγνωριστικές τιμές τύπου string len1, len2 , len3, time1, time2 και Seat με την εντολή :

```
String value =p.getStringExtra("PreviousValue");
```

Μετά από αυτό μπαίνουν όλα στο FinalText.

```
FinalText.setText(len1+seat+len2+time1+len3+time2);
```

Με τον χειριστή κουμπιού OnclickClose όταν ο χρήστης πατήσει το κουμπί η εφαρμογή καθυστερεί για δύο δευτερόλεπτα χρησιμοποιώντας την Thread.sleep και κλείνει όλα τα νήματα αμέσως μετά.

```
int secondsToSleep=2;
try {
    Thread.sleep(secondsToSleep * 1000);
} catch (InterruptedException ie) {
    Thread.currentThread().interrupt();
}
```

3.1.8 Δραστηριότητα Administrator

Για την λειτουργία της καρτέλας του διαχειριστή απαιτείται σύνδεση στην πλατφόρμα του firebase για την άντληση δεδομένων από τους αισθητήρες. Άρα για αυτό χρειάζονται οι συγκεκριμένες βιβλιοθήκες.

```
com.google.firebase.database.DatabaseReference;
 com.google.firebase.database.FirebaseDatabase;
```

Μέσα στο public class Administrator ορίζεται το Database Reference δηλαδή το όνομα της βάσης δεδομένων του firebase που της δόθηκε.

```
private DatabaseReference ESPDatabase;
```

Ύστερα πρέπει να δοθεί στο firebase οδηγία από ποια πτυχή του ζητούνται δεδομένα , δηλαδή στο δένδρο με όνομα Sensor.

```
ESPDatabase = FirebaseDatabase.getInstance().getReference("Sensor");
```

Για να καταλάβει η εφαρμογή ποια TextView η Button πρέπει να αλλάξει θα χρειαστεί να ανακτηθούν ταυτότητές τους. Αυτό γίνεται με την χρήση της εντολής findViewById(R.id.example).

```
TextView CO1 = findViewById(R.id.textView36);
TextView LPG = findViewById(R.id.textView34);
TextView Temp = findViewById(R.id.textView39);
TextView Hum = findViewById(R.id.textView32);

Button b1 = findViewById(R.id.button3);
Button b2 = findViewById(R.id.button4);
Button b3 = findViewById(R.id.button5);
Button b4 = findViewById(R.id.button6);
Button b5 = findViewById(R.id.button7);
Button b6 = findViewById(R.id.button8);
Button b7 = findViewById(R.id.button9);
Button b8 = findViewById(R.id.button10);
Button b9 = findViewById(R.id.button11);
```

Αφού ανακτηθούν οι ταυτότητες, η διαδικασία άντλησης δεδομένων και εμφάνισής τους για τους αισθητήρες αερίων και θερμοκρασίας-υγρασίας είναι η ίδια.

Πρώτα καλείται να βρεθεί από ποιο κλαδί του δένδρου ή παιδί χρειάζονται δεδομένα.

```
ESPDatabase.child("CO").get().addOnCompleteListener(task -> {
```

Μετά θα ανακτηθεί το αντικείμενο value 1 .

```
Object value1 = task1.getResult().getValue();
```

Εφόσον το η αντικείμενο υπάρχει είναι νούμερο ή double και η αίτηση της εφαρμογής στην βάση δεν απορριφθεί

```
if (task1.isSuccessful() && task1.getResult().exists()) {
    if(value1 instanceof Number){
```

τότε τοποθετείται η τιμή του αντικειμένου στο αντίστοιχο κομμάτι κειμένου TextView και εμφανίζεται μήνυμα επιτυχίας ανάγνωσης πληροφορίας.

```
Toast.makeText(this, "Successful read", Toast.LENGTH_SHORT).show();

String co = String.valueOf(value1);
CO1.setText(co);
```

Εάν η αίτηση της εφαρμογής απορριφθεί ή ανάγνωση αποτύχει για οποιονδήποτε άλλον λόγο τότε εμφανίζεται μήνυμα σφάλματος στην οθόνη.

```
Toast.makeText(this, "Failed read", Toast.LENGTH_SHORT).show();
```

Για να μπορεί να βλέπει ο διαχειριστής την κατάσταση των θέσεων υπάρχει η On-ClickAvailabillity1. Ο συγκεκριμένος χειριστής του κουμπιού "Διαθεσιμότητα" λαμβάνει από την βάση δεδομένων του firebase είτε τον αριθμό 0 ή 1 και αλλάζει το χρώμα των άλλων κουμπιών σε γκρίζο για πιασμένη η πράσινο για κενή για την αναπαράσταση της κατάστασής της. Αυτό το επιτυγχάνει παρόμοια με τον παραπάνω κώδικα αντλώντας από τις διεργασίες παιδιά PIR. Όπως και στα παραπάνω γίνεται ανάκτηση της τιμής του αντικείμενου value. Εφόσον υπάρχει σαν integer και Long, η αίτηση της εφαρμογής στην βάση ΄δεν απορριφθεί για κάποιο λόγο η τιμή του value περνιέται σαν string στην μεταβλητή onoff.

```
if (task8.isSuccessful() && task8.getResult().exists()) {
    if (value8 instanceof Long || value8 instanceof Integer) {
String onoff = String.valueOf(value8);
```

Αυτή η μεταβλητή ελέγχεται από μία if η οποία αν ανιχνεύσει μηδέν κάνει το συγκεκριμένο κουμπί γκρίζο και αν ανιχνεύσει οτιδήποτε άλλο το κάνει πράσινο.

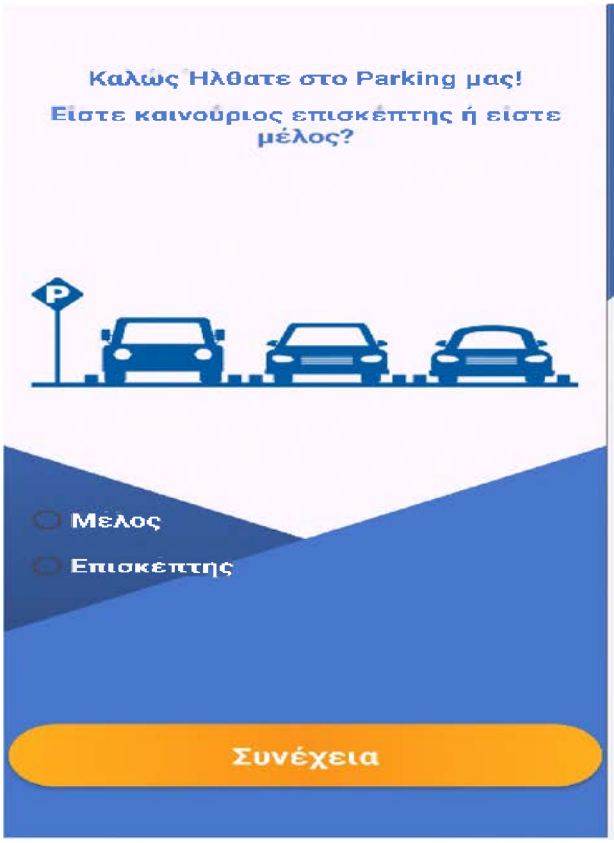
```
if (onoff.equals("0")) {
    b5.setBackgroundColor(Color.GRAY);
    b5.setTextColor(Color.WHITE);
} else {
    b5.setBackgroundColor(Color.GREEN);
    b5.setTextColor(Color.BLACK);
}
```

Τέλος σε περίπτωση αποτυχίας μεταφοράς δεδομένων εμφανίζεται μήνυμα σφάλματος στην οθόνη.

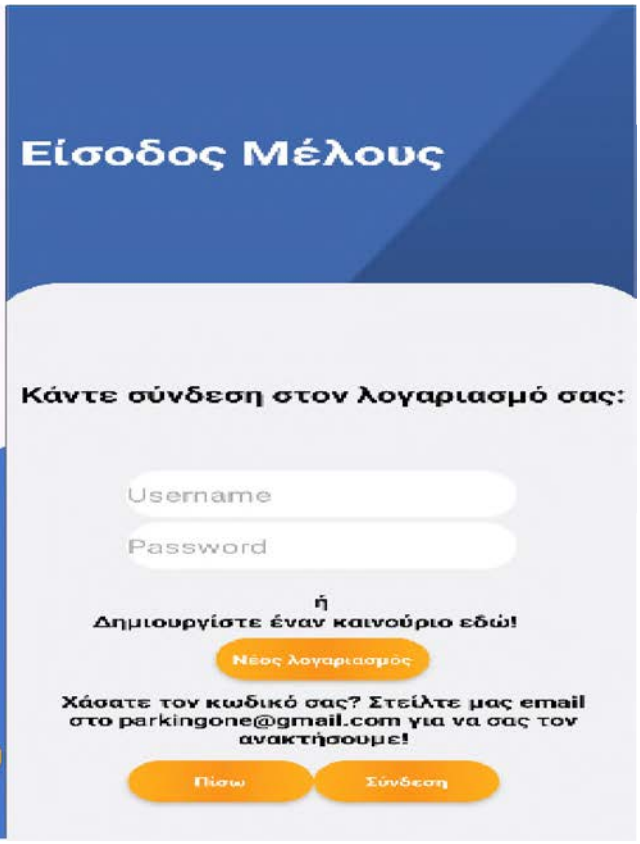
```
Toast.makeText(this, "Failed read", Toast.LENGTH_SHORT).show();
```

3.2 Παράδειγμα Λειτουργίας

Ο χρήστης καλείται να επιλέξει το αν είναι μέλος ή επισκέπτης.(MainActivity):



Καρτέλα MainActivity



Καρτέλα HandleMember

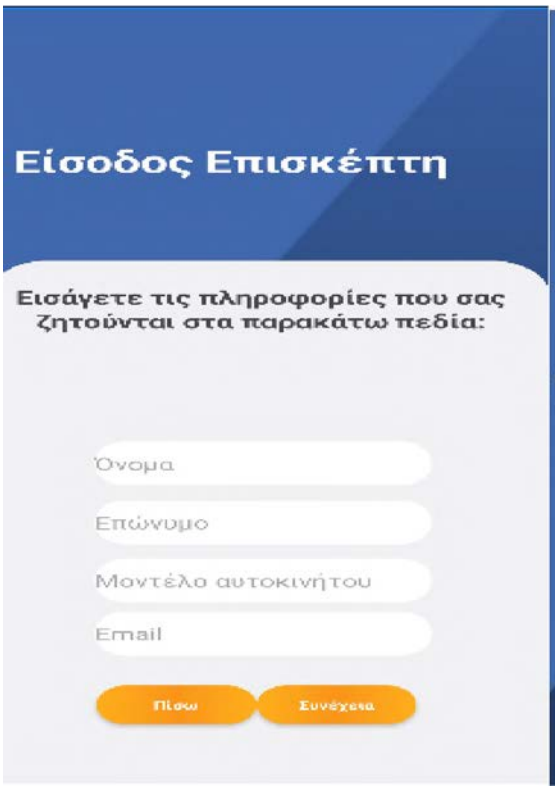
Περίπτωση 1: Μέλος

Η εφαρμογή ζητάει τον όνομα χρήστη και το συνθηματικό του ή δίνει την εναλλακτική να δημιουργήσει καινούριο.

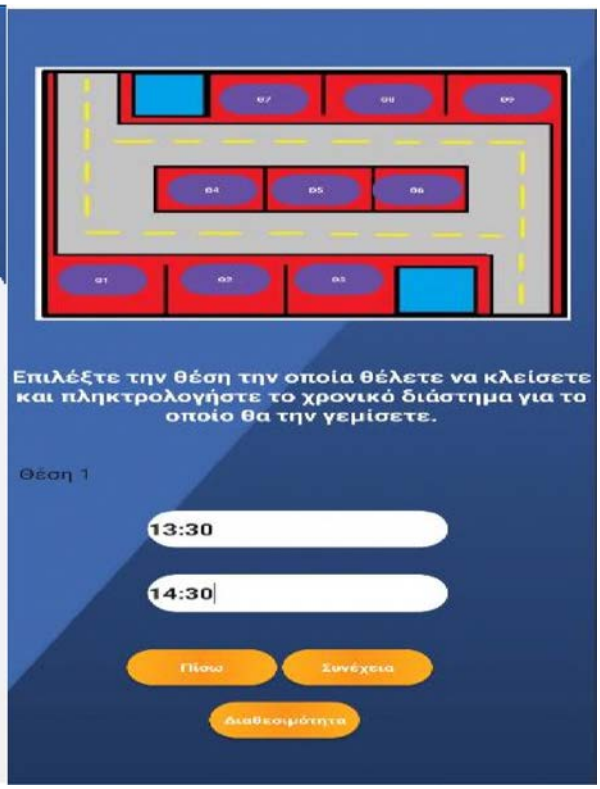
(Καρτέλα HandleMember)

Περίπτωση 2: Επισκέπτης

Ζητούνται να συμπληρωθούν τα στοιχεία, Ονοματεπώνυμο, τύπος αυτοκινήτου, κυβικά και email.(Καρτέλα HandleGuest)

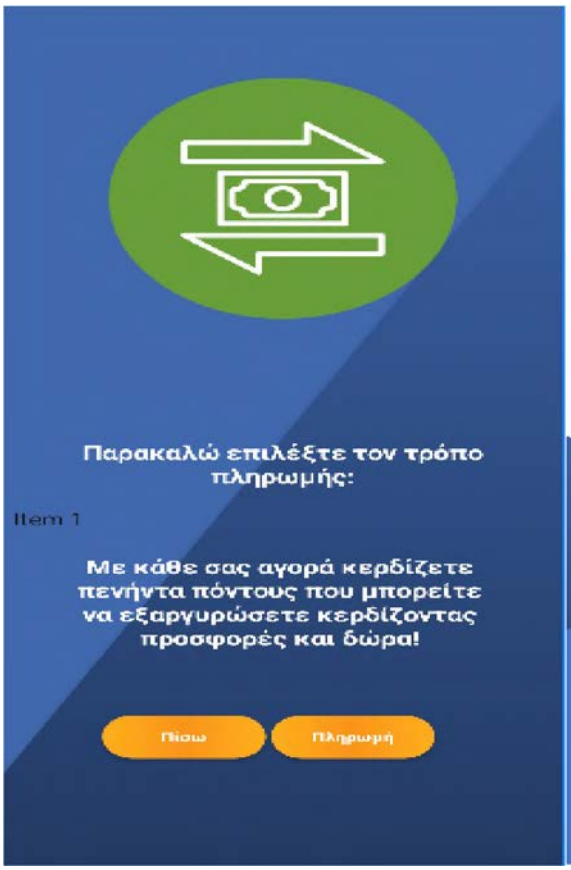


Καρτέλα HandleGuest



Καρτέλα SeatPick

Ύστερα αφού κατηγοριοποιήσει η εφαρμογή τον χρήστη , τον κατευθύνει σε άλλο παράθυρο το οποίο του ζητάει το διάστημα για το οποίο χρειάζεται την θέση στάθμευσης και ποια θέση προτιμάει(Καρτέλα SeatPick). Αφού τα συμπληρωθούν και αυτά τον κατευθύνει στην δραστηριότητα πληρωμής SeatPay στην οποία έχει τέσσερις επιλογές για να πληρώσει(Καρτέλα SeatPay). Μετά, στην επόμενη δραστηριότητα εμφανίζεται ένα μοντέλο του χώρου με τις θέσεις που είναι διαθέσιμες να τονίζονται. Η εφαρμογή καθοδηγεί τον πελάτη στην θέση που επέλεξε και εμφανίζει το τελευταίο μήνυμα εκτέλεσης(Καρτέλα NavigationActivity).



Καρτέλα SeatPay

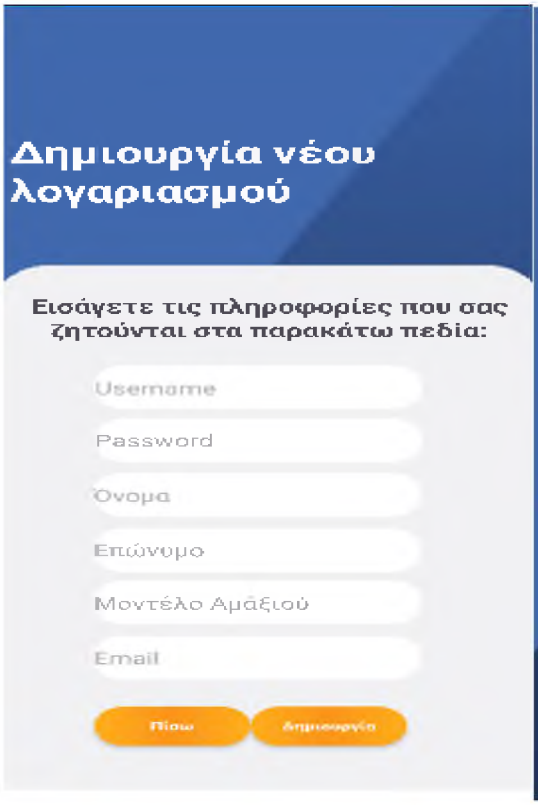
Καρτέλα Navigation Activity

Περίπτωση 3:

Για να δημιουργηθεί καινούριος χρήστης, ανακατευθύνεται ο πελάτης στην ανάλογη καρτέλα.(Καρτέλα NewMember) Εκεί του ζητούνται κάποια προσωπικά στοιχεία για αποθήκευση στην βάση δεδομένων. Με το πάτημα του κουμπιού Δημιουργία ανακατευθύνεται πάλι στην HandleMember οπού καλείται να συνδεθεί με τα στοιχεία που εισήγαγε προηγουμένως.

Στην περίπτωση του Διαχειριστή:

Εισάγονται στην καρτέλα του μέλους τα στοιχεία του διαχειριστή(Username,password). Ύστερα εμφανίζεται η καρτέλα διαχείρισης η οποία δείχνει δεδομένα για τις θέσεις, την κατάστασή τους(λειτουργική, ελεύθερη-πιασμένη), την θερμοκρασία, την υγρασία, και την περιεκτικότητα του αέρα σε βλαβερά αέρια όπως μονοξείδιο, προπάνιο η ακόμα και καπνό.(Καρτέλα Administrator)



Καρτέλα NewMember

Καρτέλα Administrator

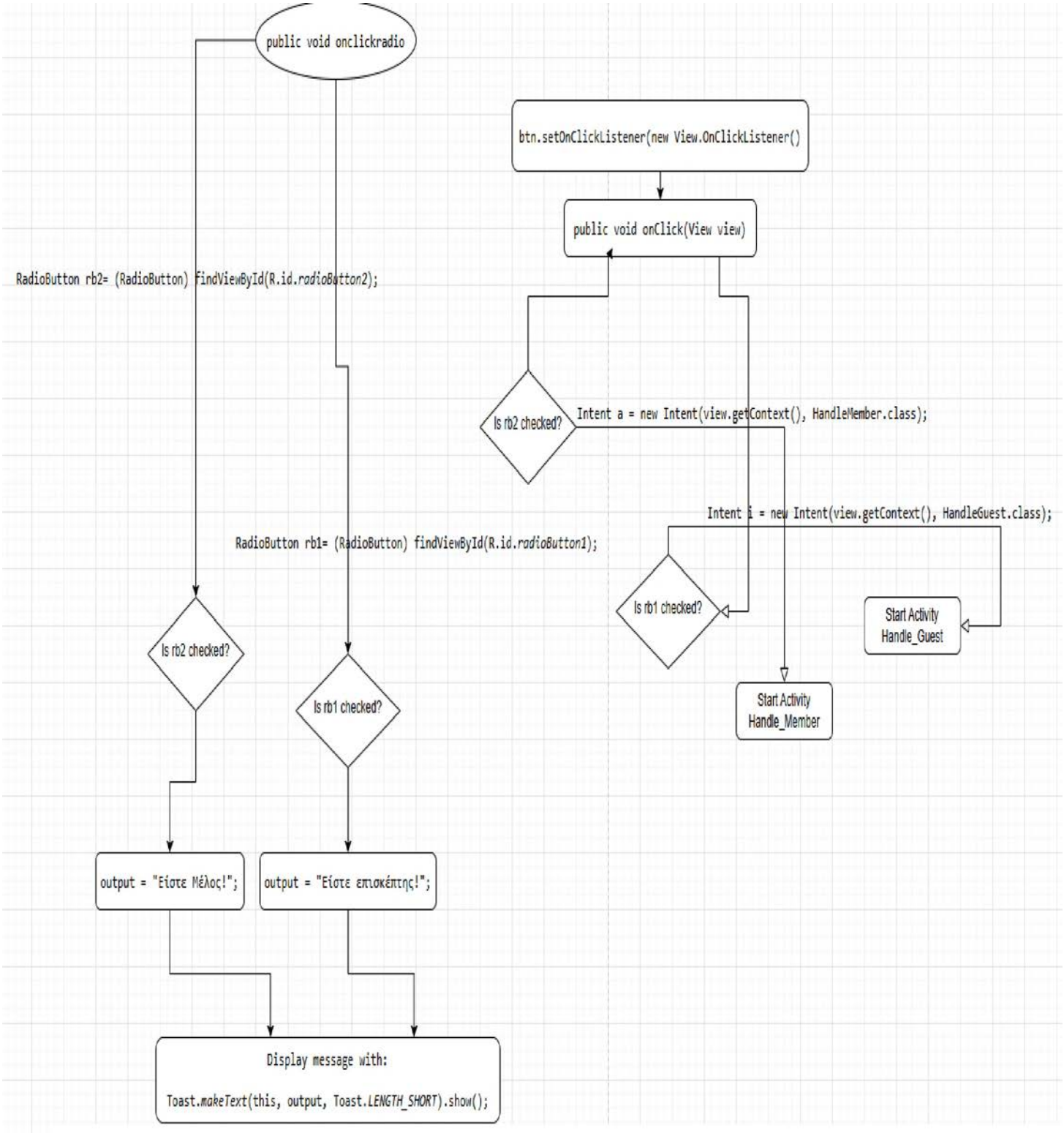
3.3 Διαγράμματα Ακολουθίας

Από εδώ θα ακολουθήσουν τα διαγράμματα για κάθε δραστηριότητα της εφαρμογής. Όλα φτιάχτηκαν στην ιστοσελίδα draw.io[72] με το στυλ flowchart με σκοπό να

εξηγήσουν πιο συνοπτικά τον ρόλο και την λειτουργία κάθε δραστηριότητας στην εφαρμογή.

3.3.1 Διάγραμμα της Main_activity

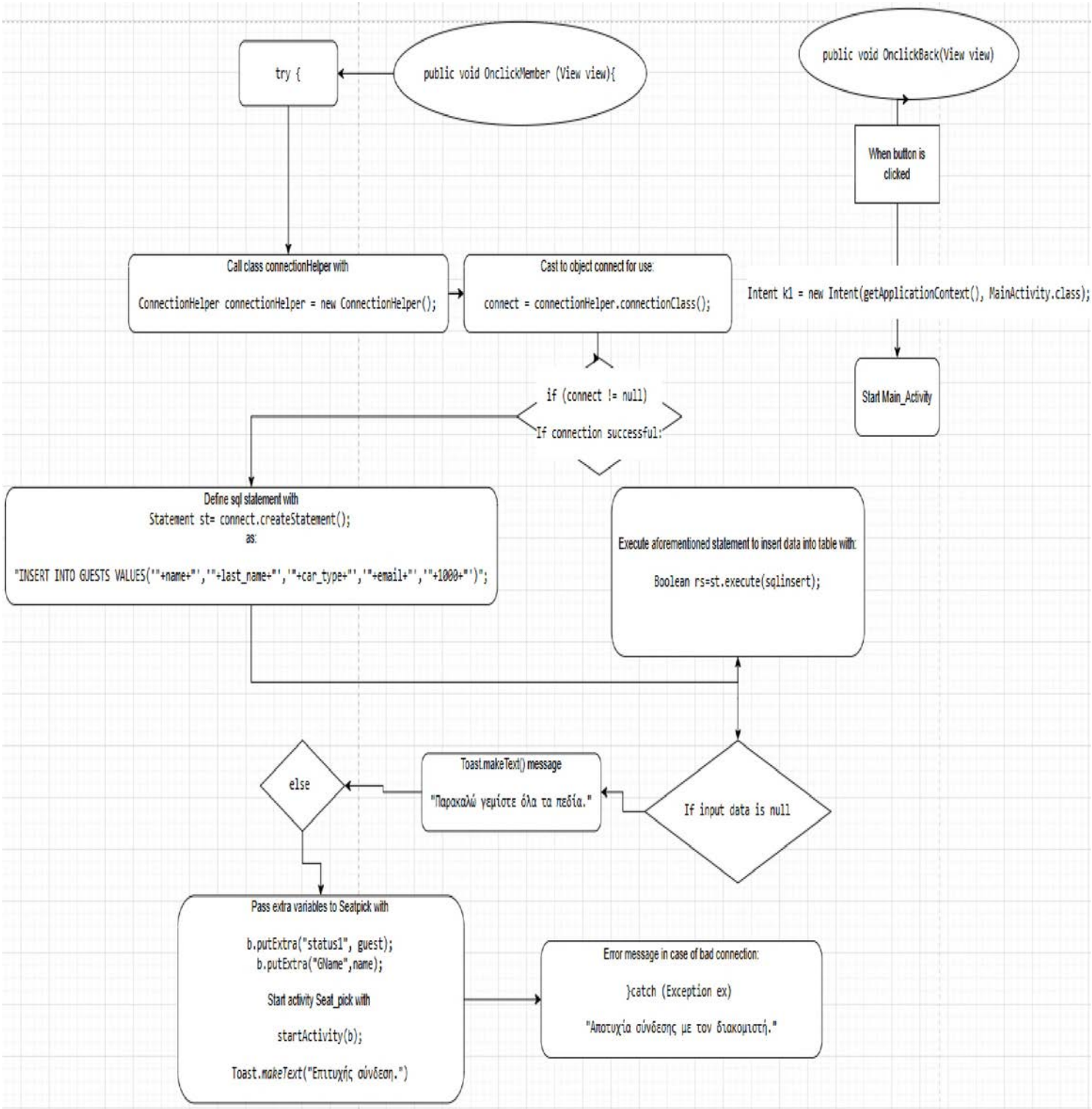
Η εισαγωγική σελίδα για τον χρήστη. Τον προτρέπει να επιλέξει μεταξύ επισκέπτη ή μέλους και τον κατευθύνει στην κατάλληλη δραστηριότητα ανάλογα με το κουμπί που πατήθηκε(Flowchart Main_Activity).



Flowchart Main_Activity

3.3.2 Διάγραμμα της HandleGuest

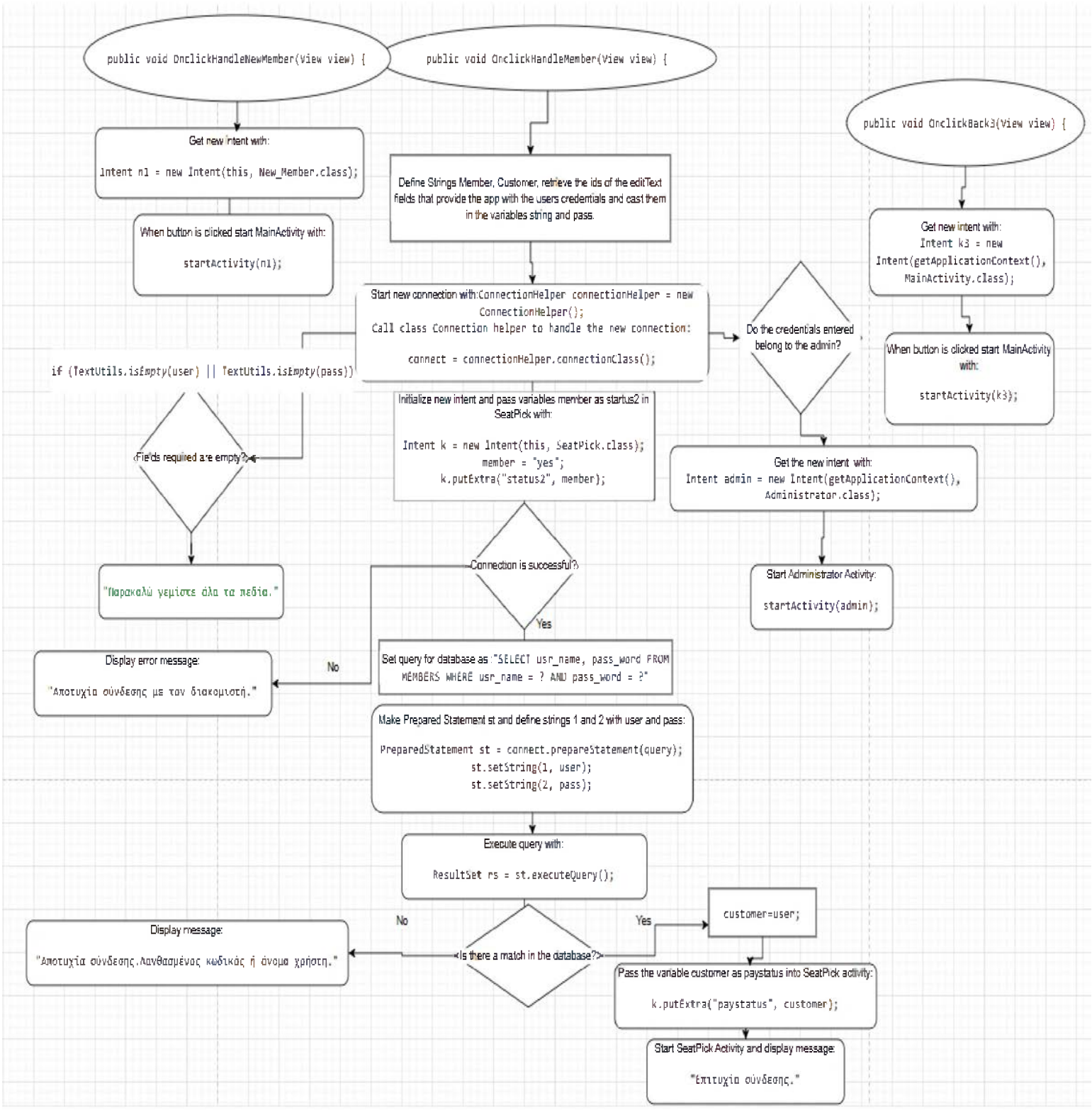
Ο ρόλος της HandleGuest κλάσης είναι να συλλέγει τα δεδομένα που εισάγει ο χρήστης με μεταβλητές και να τα εισάγει στην βάση δεδομένων σαν αρχείο καταγραφής κινήσεως(Flowchart Handle_Guest).



Flowchart Handle_Guest

3.3.3 Διάγραμμα της HandleMember

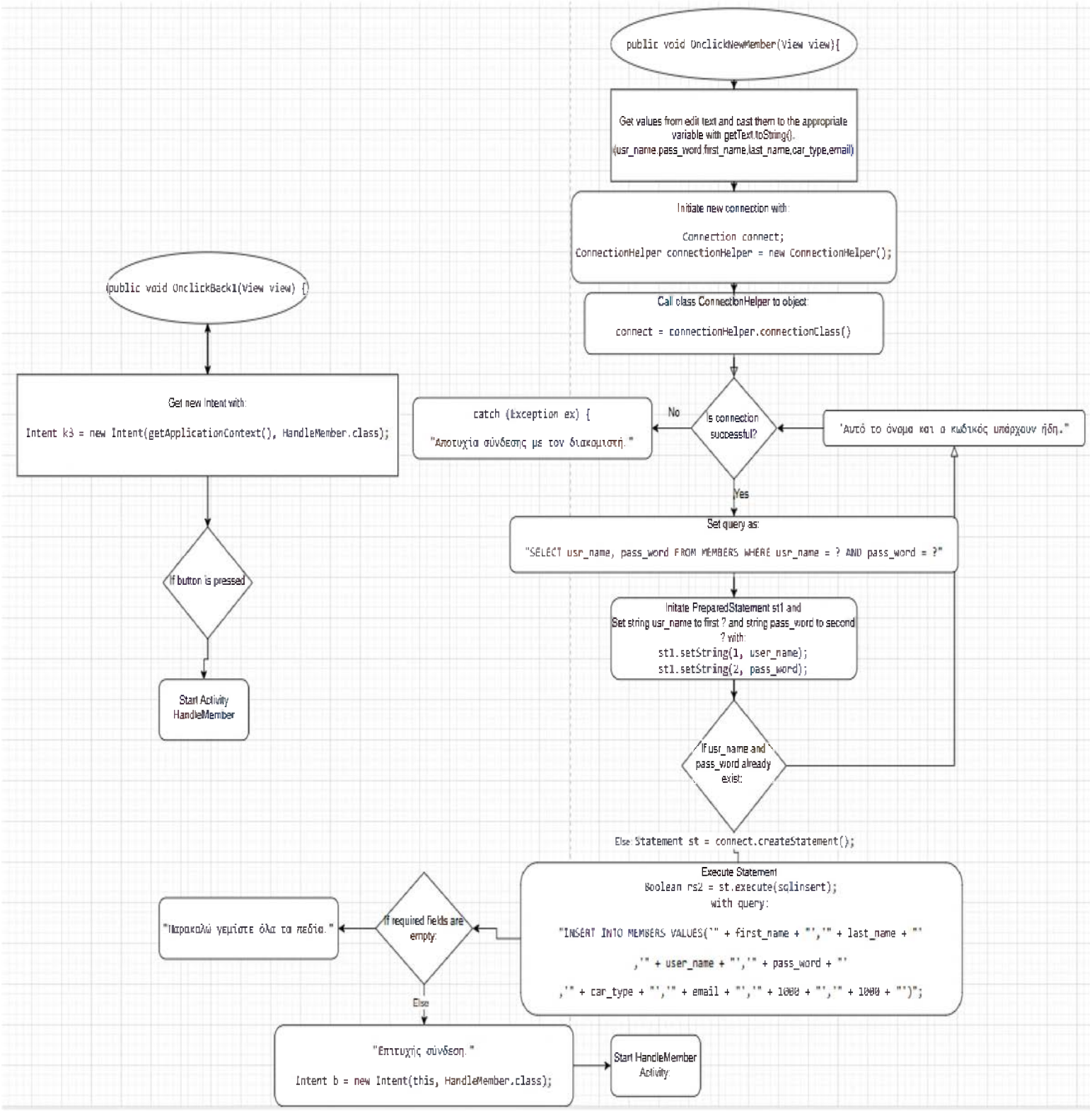
Σκοπός της είναι η ταυτοποίηση του χρήστη κάνοντας σύγκριση των εισακτέων δεδομένων με τα ήδη υπάρχουσα στην βάση SQL. Επίσης λειτουργεί και ως πύλη πρόσβασης για την καρτέλα του διαχειριστή(Flowchart Handle_Member).



Flowchart Handle_Member

3.3.4 Διάγραμμα της New Member

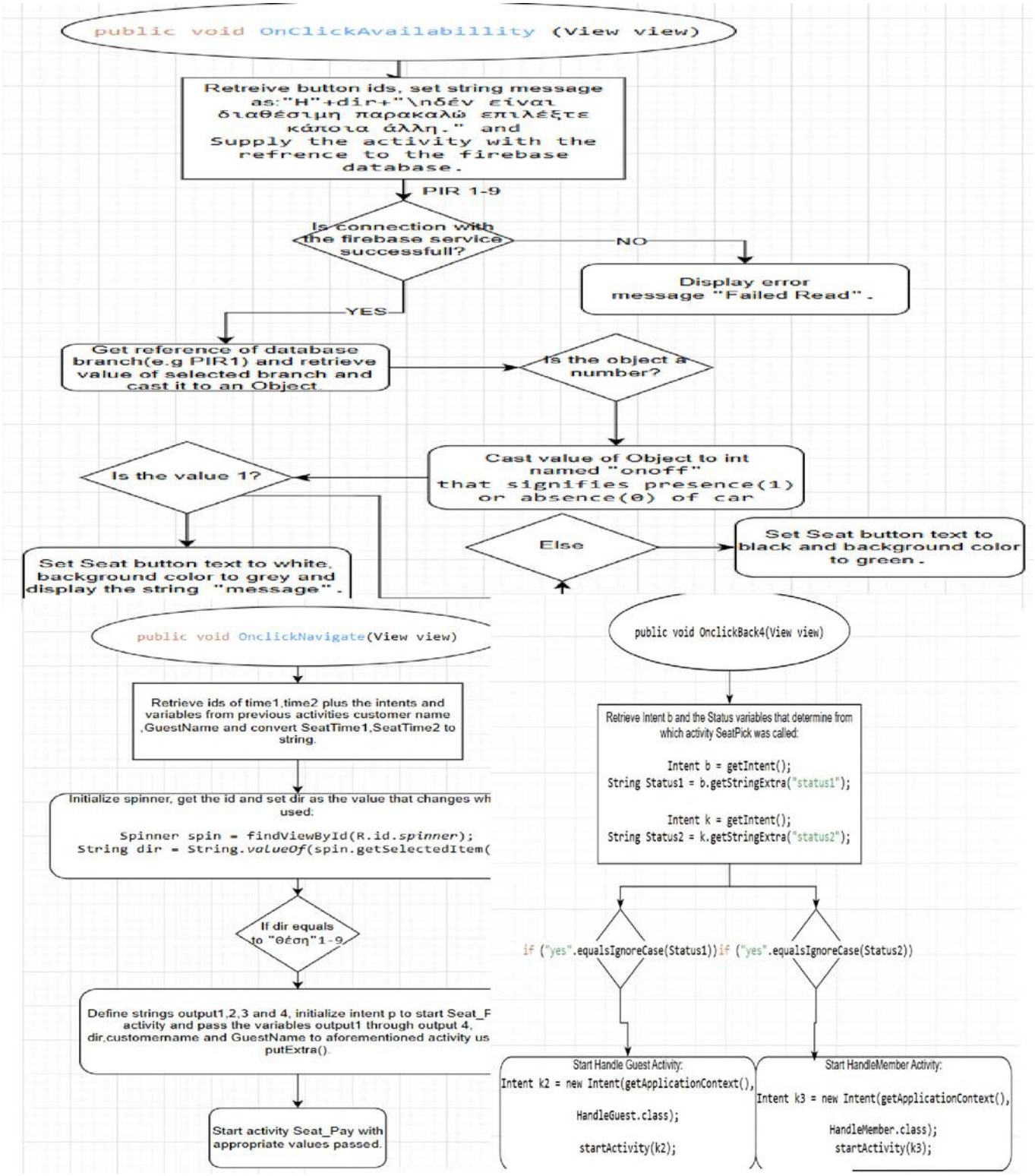
Η NewMember έχει την δουλειά της δημιουργίας καινούργιων λογαριασμών ανακτώντας τα δεδομένα που εισάγει ο χρήστης και εισάγοντας τα στην βάση δεδομένων(Flowchart New_Member).



Flowchart New_Member

3.3.5 Διάγραμμα της Seat Pick

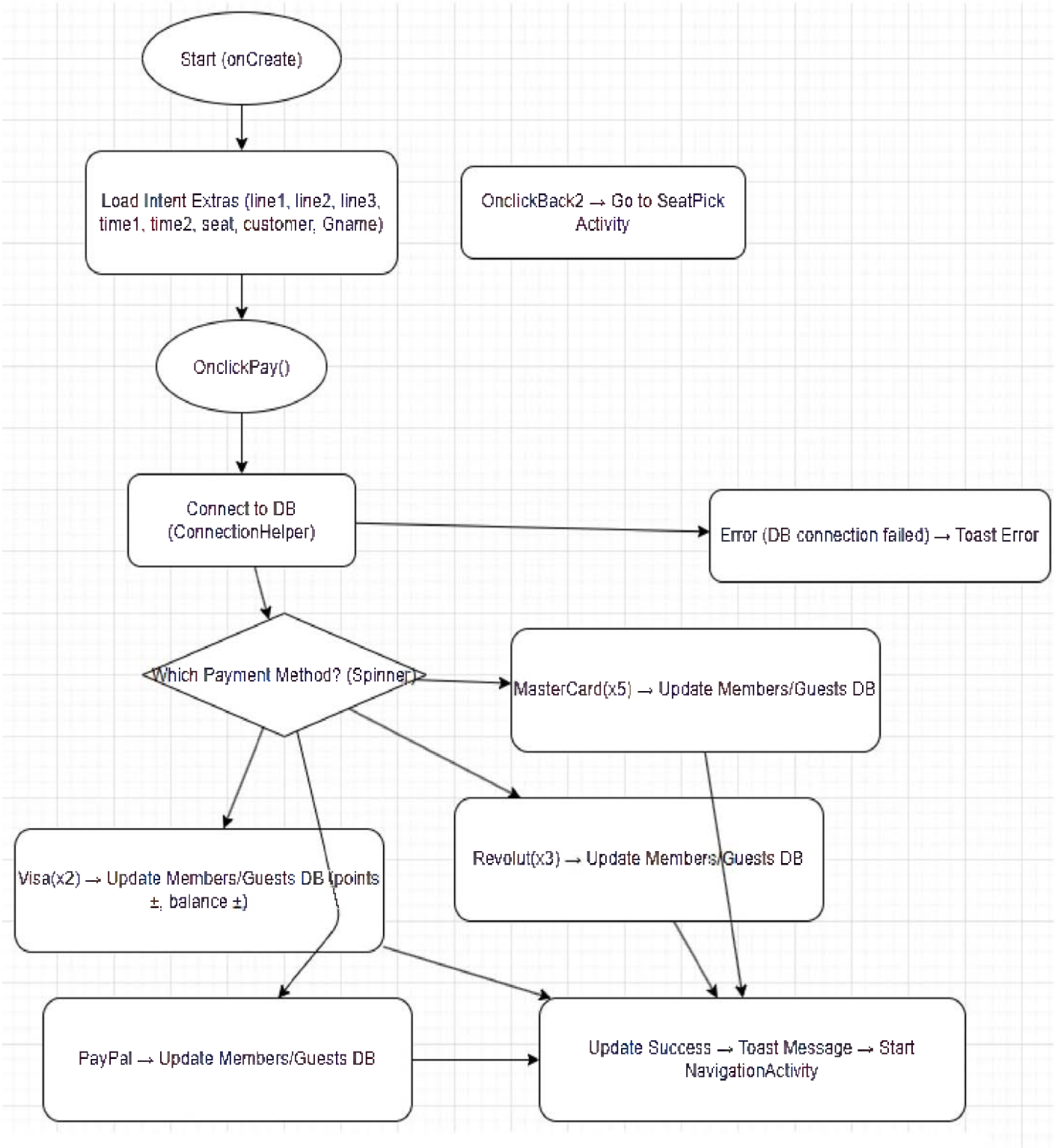
Η κλάση SeatPick ενημερώνει τον χρήστη για την διαθεσιμότητα των θέσεων αντλώντας δεδομένα από το firebase. Αφού ο χρήστης επιλέξει την θέση που θέλει και το χρονικό διάστημα για το οποίο την χρειάζεται τον κατευθύνει ανάλογα(Flowchart Seat_Pick).



Flowchart Seat_Pick

3.3.6 Διάγραμμα της Seat Pay

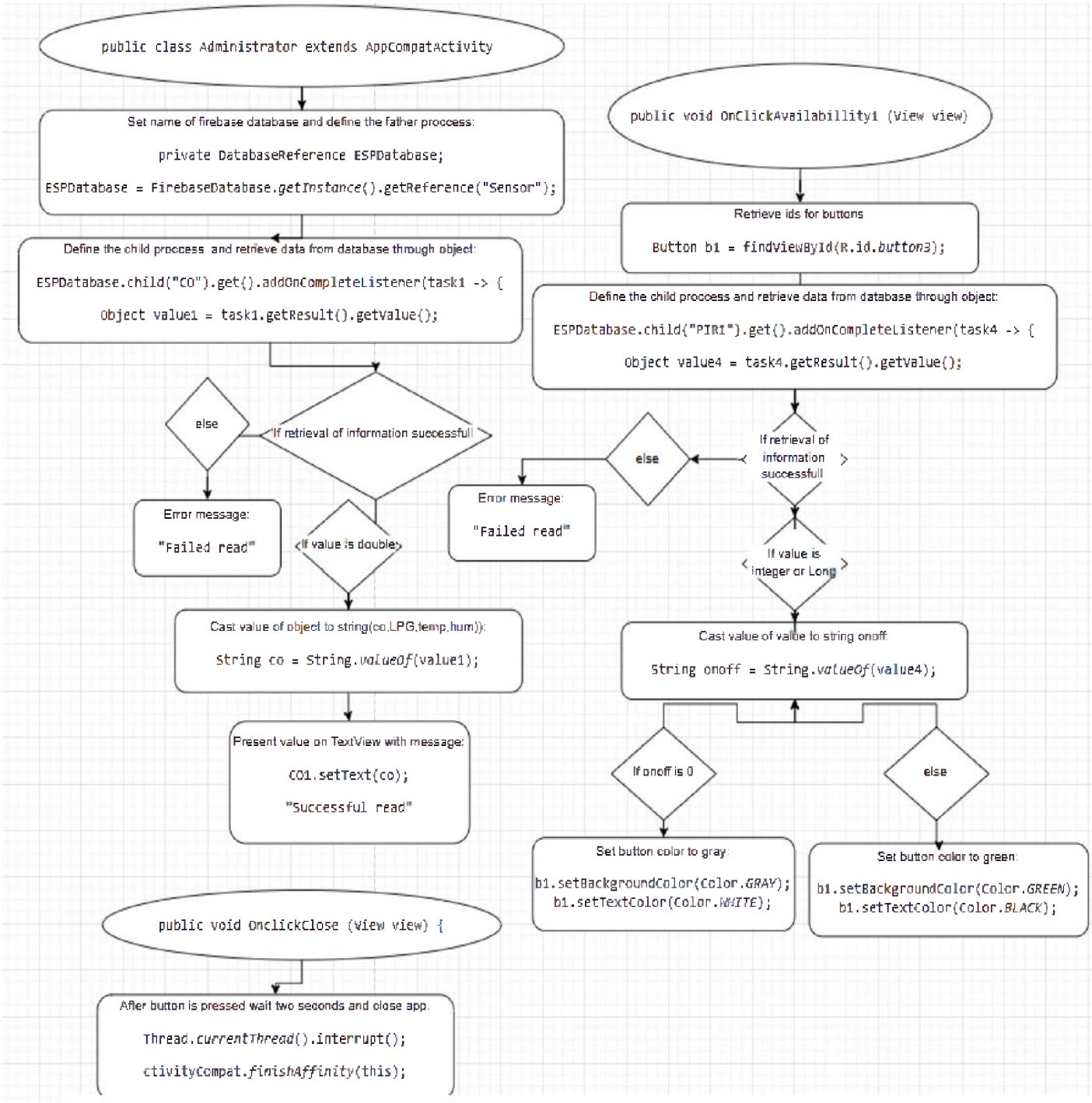
Η SeatPay επιδιώκει να προσομοιώσει την διαδικασία χρηματικής συναλλαγής. Λαμβάνει τις μεταβλητές αναγνώρισης του χρήστη και ανάλογα με την επιλογή πληρωμής του αφαιρεί το κατάλληλο ποσό από τον λογαριασμό του μαζί με τους πόντους(Flowchart Seat_Pay).



Flowchart Seat_Pay

3.3.7 Διάγραμμα της Administrator

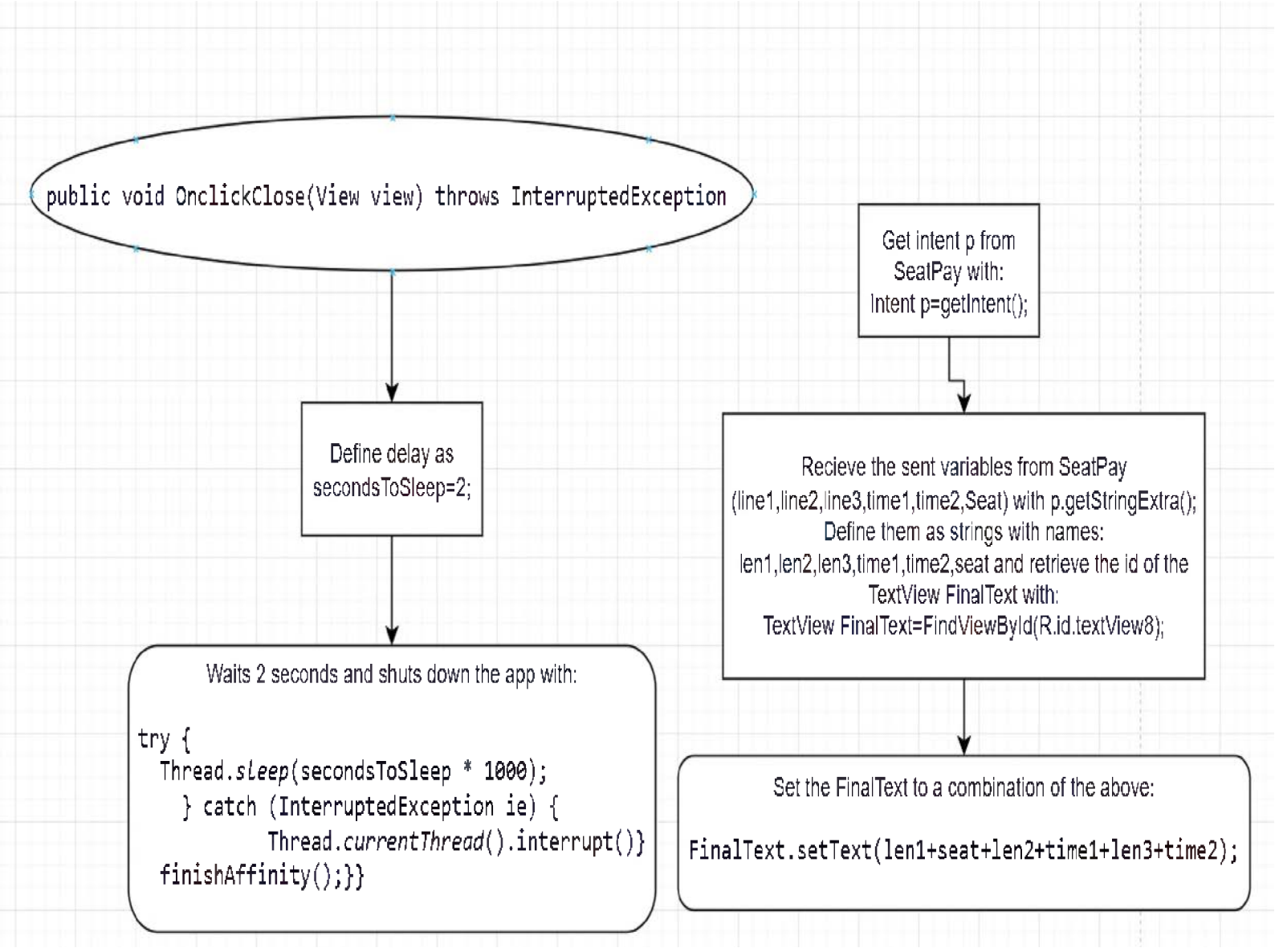
Η κλάση ξεκινάει συνδέοντας στο Firebase .Στα επόμενα βήματα διαβάζει τα δεδομένα που λαμβάνει από αυτό και στην συνέχεια με το πάτημα του κατάλληλου κουμπιού τα εμφανίζει στον χρήστη(Flowchart Administrator).



Flowchart Administrator

3.3.8 Διάγραμμα της Navigation Activity

Δουλειά της είναι η ανάκτηση μεταβλητών από προηγούμενες κλάσεις -κάτι που γίνεται πρώτο- και η εμφάνιση της κατάλληλης κινούμενης εικόνας για την καθοδήγηση του χρήστη . Το μόνο που μένει στο τέλος είναι η έξοδος του πελάτη η οποία γίνεται με κουμπί(Flowchart Navigation_Activity).



Flowchart Navigation_Activity

4 Σχεδίαση Βάσης Δεδομένων

Για να μπορεί να αποθηκεύει τους λογαριασμούς των πελατών η εφαρμογή χρειάζεται κάποιου τύπου βάσης δεδομένων. Η δουλειά της εκτείνεται στην αποθήκευση στοιχείων για τα μέλη του κλειστού χώρου στάθμευσης αλλά και για τον επισκέπτη που δεν έχει κάνει ακόμα λογαριασμό.

Για το μέλος αποθηκεύεται το όνομα χρήστη το συνθηματικό , το ονοματεπώνυμο ξεχωριστά, ο τύπος του αμαξιού με το οποίο σταθμεύει το, email ,ένα ενδεικτικό χρηματικό ποσό με το οποίο έκανε λογαριασμό και τους εκπτωτικούς πόντους που έχει κερδίσει με τις αγορές του.

Για τον επισκέπτη αποθηκεύεται ονοματεπώνυμο, ο τύπος του αμαξιού με το οποίο σταθμεύει, το email , και τέλος ένα ενδεικτικό χρηματικό ποσό ..

Για την βάση δεδομένων χρησιμοποιήθηκε η πλατφόρμα SSMS(Sql Management Studio) ενώ η βάση γράφτηκε με εντολές SQL.(Πίνακας Guests-Πίνακας Members)

4.1.1 Δημιουργία της βάσης:

```
CREATE TABLE GUESTS(
    guestID INT PRIMARY KEY,
    first_name varchar(30),
    last_name  varchar(30),
    car_type   varchar(30),
    email varchar(30)
    balance float
    )
```

	Column Name	Data Type	Allow Nulls
🔑	guestID	int	☐
	first_name	varchar(30)	☑
	last_name	varchar(30)	☑
	car_type	varchar(30)	☑
	email	varchar(30)	☑
	balance	float	☑
			☐

Πίνακας Guests.

```
CREATE TABLE MEMBERS(
    memberID INT PRIMARY KEY,
    first_name varchar(30),
    last_name  varchar(30),
    usr_name varchar(30),
    pass-word varchar(30),
    car_type   varchar(30),
    email varchar(30)
    balance float
    )
```

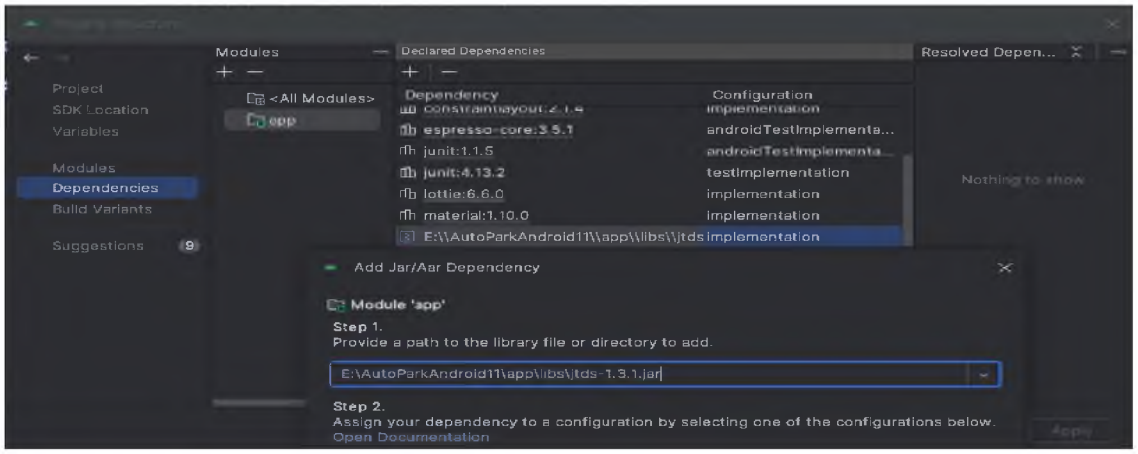
	Column Name	Data Type	Allow Nulls
🔑	memberID	int	☐
	first_name	varchar(30)	☑
	last_name	varchar(30)	☑
	usr_name	varchar(30)	☑
	pass_word	varchar(30)	☑
	car_type	varchar(30)	☑
	email	varchar(30)	☑
	balance	float	☑
	points	int	☑
			☐

Πίνακας Members.

4.2 Ενσωμάτωση στην εφαρμογή

Η βάση χρησιμοποιείται μέσα στην εφαρμογή όταν απαιτείται η αποθήκευση η άντληση δεδομένων για την λειτουργία της. Συνεπώς χρησιμοποιείται στην HandleGuest για την δημιουργία επισκέπτη, στην HandleMember για την σύγκριση συνθηματικών ,στην New Member για την δημιουργία καινούριου πελάτη και στην SeatPay για την εξόφληση των θέσεων. Για να επιτευχθούν όλα αυτά προϋποθέτετέ η χρήση κάποιων βιβλιοθηκών, εντολών και jar dependencies για την επικοινωνία μεταξύ Android Studio SQL Server Management Studio.

Για την επικοινωνία αρχικά χρειάζεται το αρχείο mysql-connector-j-9.1.0, ένα αρχείο τύπου jar το οποίο επιτρέπει το Android Studio να επικοινωνήσει με μία βάση SQL (Android Dependencies).



Android Dependencies

Στην συνέχεια πρέπει να εισαχθούν οι κατάλληλες εντολές στο Android Manifest (Permissions)(<uses-permissionandroid:name="android.permission.INTERNET"/> <usespermissionandroid:name="android.permission.ACCESS_NETWORK_STATE"/)

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AutoParking"
        tools:targetApi="31">
```

Permissions

Αφού γίνουν αυτά πρέπει να επιτευχθεί σύνδεση με την βάση δεδομένων.

Αυτήν την δουλειά θα την έχει η κλάση ConnectionHelper. η οποία θα καλείται κάθε φορά που θέλει μια δραστηριότητα να συνδεθεί στην βάση δεδομένων για να συγκρίνει, αντλήσει ή να αλλάξει δεδομένα μέσα σε αυτήν(Connection Class).

```java
package com.example.autoparkandroid11;

import android.os.StrictMode;
import android.util.Log;

import java.sql.Connection;
import java.sql.DriverManager;

public class ConnectionHelper {
    Connection con;  no usages
    String uname, pass, ip, port, database;  2 usages

    public Connection connectionClass() {
        ip = "192.168.1.3";
        database = "CUSTOMERS";
        uname = "sa";
        pass = "root12345";
        port = "1433";
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        Connection connection = null;
        String ConnectionURL = null;
        try {
            Class.forName( className: "net.sourceforge.jtds.jdbc.Driver");
            ConnectionURL = "jdbc:jtds:sqlserver://" + ip + ":" + port + ";" + "databasename=" + database + ";user=" + uname + ";password=" + pass + ";";
            connection= DriverManager.getConnection(ConnectionURL);
        } catch (Exception ex) {
            Log.e( tag: "Error", msg: "Connection error: ",ex);
        }
        return connection;
```

Connection Class

Αρχικά ορίζονται τα συνθηματικά για την σύνδεση στην πλατφόρμα του SSMS όομα χρήστη(uname) κωδικός(pass), διεύθυνση(ip), πύλη(port) και όνομα βάσης δεδομένων database

Περικυκλωμένο από μία μία try catch καλείται το "net.sourceforge.jtds.jdbc.Driver" ο οδηγός για την σύνδεση. Μέσα στην ίδια δίνονται τα στοιχεία σύνδεσης στον οδηγό ο οποίος τα περνάει με την εντολή :

```
ConnectionURL = "jdbc:jtds:sqlserver://" + ip + ":" + port + ";" + "databasename=" + da-
tabase + ";user=" + uname + ";password=" + pass + ";";.
```

Τέλος ανακτείται η διεύθυνση της σύνδεσης με την :

```
connection= DriverManager.getConnection(ConnectionURL);
```

Μέσα στην catch σε περίπτωση αποτυχίας σύνδεσης περναέι στο logcat το μήνυμα "Connection error: ", με το tag Error κάνοντας χρήση της βιβλιοθήκης: java.sql.SQLException;.

4.2.1 Ιδιότητες των εντολών εντός των δραστηριοτήτων

Για την επικοινωνία της κάθε δραστηριότητας με την βάση χρησιμοποιούνται τριών ειδών ερωτήματα. Κάθε ένα από αυτά εκτελείται σε διαφορετικές καταστάσεις για διαφορετικούς σκοπούς.

Το ερώτημα τύπου UPDATE έχει δομή:

```
UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;
```

Χρησιμοποιείται εκτενώς για ενημέρωση ήδη υπαρχόντων δεδομένων μέσα στην βάση και επομένως υπάρχει μόνο στην δραστηριότητα SeatPay ένα έως εννέα η οποία είτε αφαιρεί είτε προσθέτει ποσά στις στήλες balance και points αντίστοιχα.

Το ερώτημα τύπου :SELECT έχει δομή:

```
SELECT value1 FROM table1 WHERE value1 = 'somevalue' ;
```

Χρησιμοποείιται σαν ευρετήριο είτε για μεταβλητές που χρειάζονται για σύγκριση όπως είναι τα συνθηματικά του μέλους στην δραστηριότητα New_Member,

```
String query = ("SELECT usr_name, pass_word FROM MEMBERS WHERE usr_name = ? AND
pass_word = ?");
```

είτε για την επιλογή συγκεκριμένων μεταβλητών με ειδικές συνθήκες όπως είναι το υποερώτημα για την εύρεση χρήστη με αρκετούς πόντους για έκπτωση στην δραστηριότητα Seat_Pay

```
String sqlSELECT ="SELECT points FROM MEMBERS WHERE usr_name=? AND points>=250"
```

Σημειώνεται ότι τα αγγλικά ερωτηματικά αναπαριστούν μεταβλητή που ορίζει ο χρήστης από την εφαρμογή με το statement.setString(1, "Παράδειγμα")-όπου 1 η θέση του στο ερώτημα ακολουθούμενο από το string-και δεν υπάρχουν απαραίτητα στην βάση δεδομένων.

Για να καλεστεί απαιτεί τον ορισμό ενός prepared statement με την εντολή:

```
PreparedStatement st = connect.prepareStatement(sqlSELECT);
```

Το ερώτημα τύπου INSERT έχει δομή:

```
INSERT INTO table_name (column1, column2, column3) VALUES (value1, value2, value3);
```

Χρησιμοποιείται στην εφαρμογή αποκλειστικά για την εισαγωγή δεδομένων ενός καινούριου χρήστη, είτε μέλος είτε επισκέπτη. Συγκεκριμένα στις δύο παρακάτω δραστηριότητες :

1.Στην New_Member για την εισαγωγή των πληροφοριών του μέλους με το ερώτημα:

```
String sqlinsert = "INSERT INTO MEMBERS VALUES('" + first_name + "','" + last_name +
"','" + user_name + "','" + pass_word + "','" + car_type + "','" + email + "','" + 1000
+ "','" + 1000 + "')";
```

2. Στην HandleGuest για την εισαγωγή των πληροφοριών του επισκέπτη με το ερώτημα:

```
String sqlinsert="INSERT INTO GUESTS
VALUES('"+name+"','"+last_name+"','"+car_type+"','"+email+"','"+1000+"')"
```

Για την χρήση των παραπάνω ερωτημάτων χρειάζεται ένα Statement το οποίο θα δημιουργηθεί καλώντας την σύνδεση connect

```
Statement st= connect.createStatement();
```

και -σε αντίθεση με τα προηγούμενα ερωτήματα που αναφέρθηκαν – μία μεταβλητή Boolean η οποία τα εκτελεί με την εντολή

```
Boolean rs=st.execute(sqlinsert);
```

5 Κατασκευή Υλικού

Για να δημιουργηθεί ένα σύστημα για την παρακολούθηση του χώρου απαιτούνταν ένα σύνολο αισθητήρες και κάποιου τύπου μικροελεγκτής ο οποίος θα λάμβανε τα δεδομένα, θα τα επεξεργάζονταν και θα τα εμφάνιζε σε κάποια κατανοητή από τον χρήστη τιμή.

Για αυτήν την δουλειά χρησιμοποιήθηκαν ο μικροελεγκτής ESP32-WROOM32D ,εννέα αισθητήρες HC-SR501, ένας αισθητήρας ανίχνευσης μονοξειδίου του άνθρακα MQ-7, ένας αισθητήρας ανίχνευσης υγραερίου MQ2 και ένας αισθητήρας μέτρησης θερμοκρασίας και υγρασίας DHT22.

Οι αισθητήρες κίνησης τοποθετήθηκαν σε μία μακέτα για την ανίχνευση της χρήσης των θέσεων ενώ οι αισθητήρες αερίων, θερμοκρασίας και υγρασίας κλείστηκαν σε ένα κουτί παίρνοντας τακτικά μετρήσεις αναμένοντας σύνδεση με την εφαρμογή.

5.1 Επεξήγηση Κώδικα Arduino IDE

Αρχικά εισάγονται οι βιβλιοθήκες που θα χρησιμοποιηθούν οι <WiFi.h>, <Firebase_ESP_Client.h>

"addons/TokenHelper.h" "addons/RTDBHelper.h" και <DHT.h>. Ύστερα ορίζονται τα συνθηματικά για την σύνδεση στο τοπικό δίκτυο ssid και password μαζί με τα api key και databse url για την σύνδεση του προγράμματος στο firebase.

Αμέσως μετά ορίζονται τα pins των αισθητήρων mq6 mq7 και dht22 αντίστοιχα

```
 const int sensorPinCO  = 34;
const int sensorPinLPG = 33;
#define DHTPIN          14
#define DHTTYPE         DHT22,
```

οι μεταβλητές στις οποίες θα μπουν τα συνθηματικά του firebase που θα χρειαστούν για την σύνδεση, οι τιμές των αισθητήρων σαν float και int,

```
int   sensorValueCO, sensorValueLPG;
float voltageCO, voltageLPG;
float RS_CO, RS_LPG;
```

```
float ratioCO, ratioLPG;
float PPM_CO, PPM_LPG;
```

```
// Environmental readings
float hum, tempC, tempF;
```

καθώς και οι τιμές που θα βοηθήσουν το πρόγραμμα να τις μετατρέψει σε ppm.

Στην void setup ορίζεται το baud rate στα 9600 και ξεκινάει η σύνδεση στο τοπικό δίκτυο με την χρήση του WiFi.begin(WIFI_SSID και WIFI_PASSWORD);. Για όσο η σύνδεση δεν είναι επιτυχής η κονσόλα εμφανίζει σαν μήνυμα(.) και ("Connecting to WiFi");

ανά 300ms. Με το που γίνει η σύνδεση εμφανίζεται μήνυμα "Connected, IP: " ακολουθούμενο από την διεύθυνση στην οποία συνδέθηκε και προχωράει στην σύνδεση του firebase.

Στέλνονται τα συνθηματικά στο firebase και γίνεται αίτημα για σύνδεση. Εφόσον εγκριθεί θα εμφανιστεί μήνυμα "Firebase signup OK" και η μεταβλητή signupOK θα πάρει τιμή true. Ειδάλλως θα εμφανιστεί μήνυμα "Signup error: " ακολουθούμενο από τον λόγο του σφάλματος.

Τέλος για την void setup ενεργοποιείται ο dht22 με την dht.begin();.

Μέσα στην void loop γίνεται η εξής διαδικασία. Το πρόγραμμα παίρνει την μέτρηση του αισθητήρα που αυτό γίνεται με την analogRead(sensorPinLPG); και analogRead(sensorPinCO); για τους δύο αισθητήρες αερίων είτε με τις εντολές

```
  hum   = dht.readHumidity();
    tempC = dht.readTemperature();
    tempF = dht.readTemperature(true);
```

για τον αισθητήρα υγρασίας-θερμοκρασίας. Μόνο στην περίπτωση των αισθητήρων αερίων η τιμή που λαμβάνεται μετατρέπεται σε ppm με την χρήση της συνάρτησης

```
voltageLPG     = sensorValueLPG * (3.3 / 4095.0);
    RS_LPG         = ((3.3 - voltageLPG) / voltageLPG) * RL;
    ratioLPG       = RS_LPG / R0;
    PPM_LPG        = A * pow(ratioLPG, B);
    Serial.print("LPG: ");
    Serial.print(PPM_LPG);
    Serial.println(" ppm");
```

Τέλος για να μεταφερθούν τα δεδομένα στο firebase οι εντολές είναι οι ίδιες με την διαφορά της διεύθυνσης της τιμής μέσα στο firebase και στην τιμή που στέλνεται μέσα στην βάση του.

εάν η ενέργεια είναι επιτυχής if (Firebase.RTDB.setFloat(&fbdo, "Sensor/CO", PPM_CO)) {

όπου "Sensor/CO" διεύθυνση και όπου PPM_CO τιμή που εισάγεται γίνεται εμφάνιση στην κονσόλα Serial.println("CO saved to RTDB");. Εάν η ενέργεια αποτύχει

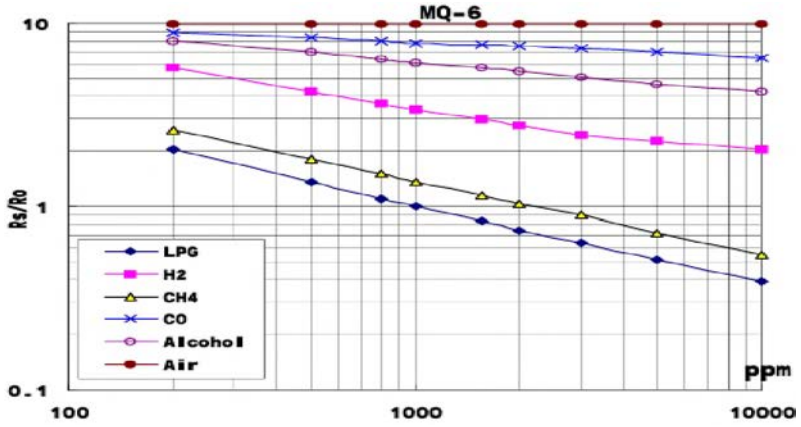
```
Serial.print("CO save failed: ");
    Serial.println(fbdo.errorReason());.
```

5.2 Αισθητήρες

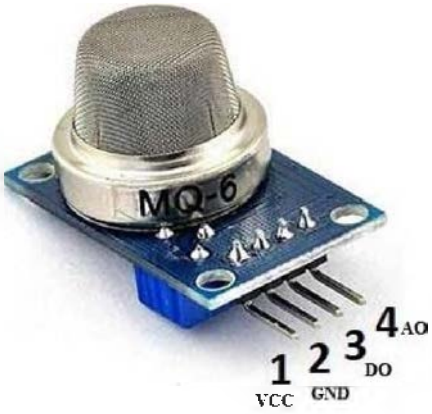
5.2.1 Αισθητήρας Υγραερίου MQ6

Πρόκειται για αισθητήρα μεταβλητής αντίστασης με τάση λειτουργίας 5V , κατανάλωση 750mW ανά μονάδα και αντίσταση 20KΩ[67] .

Λειτουργεί χρησιμοποιώντας κασσιτερίτη ο οποίος είναι ευαίσθητος σε υγραέριο και αλλάζει την αντίσταση μέσα στον αισθητήρα ανάλογα με την πυκνότητα του αερίου στο οποίο εκτίθεται. Αυτό επιστρέφει μια τιμή τάσης η οποία διαβάζεται μετά από το πρόγραμμα . Κάτι το οποίο επίσης μπορεί να επηρεάσει την αντίσταση του αισθητήρα είναι το ποσοστό οξυγόνου στον αέρα με 2% ελάχιστο και 21% επίπεδα κανονικής λειτουργίας. Το παρακάτω διάγραμμα δείχνει την σχέση αντίστασης και περιεκτικότητας αερίου ανάλογα με την αντίσταση στα αέρια τα οποία μπορεί να ανιχνεύσει ο αισθητήρας[69] με το μέγιστο να είναι 10000ppm.(Διάγραμμα ευαισθησίας αερίων MQ6).



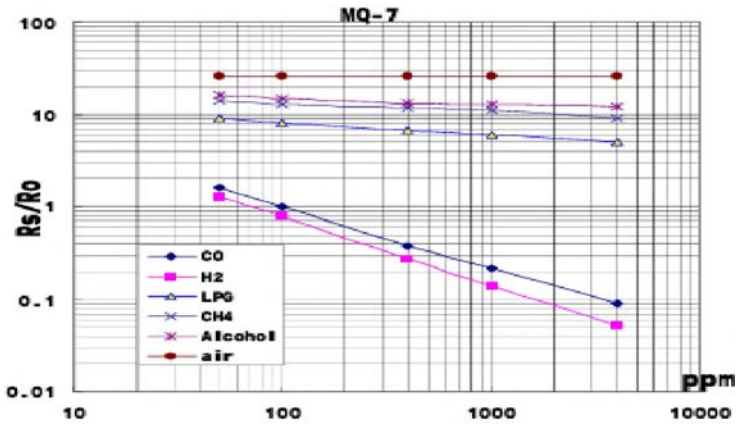
Διάγραμμα ευαισθησίας αερίων MQ6[67]



MQ6

5.2.2 Αισθητήρας μονοξειδίου του άνθρακα MQ7

Ο MQ7 είναι αισθητήρας με βάση χημικού ανίχνευσης τον διοξείδιο του κασσίτερου η SnO2 και έχει ευαισθησία ανίχνευσης από 20 μέχρι 2000 ppm. Με την χρήση ηλεκτροδίων οξειδίου του αργιλίου, ανάλογα με την έκθεση του υλικού σε διάφορα επίπεδα μονοξείδιου του άνθρακα μειώνεται και η αντίστασή του[69](Διάγραμμα Ευαισθησίας αερίων MQ7). Έτσι έχει ως αποτέλεσμα την μεγαλύτερη τάση στα άκρα του αισθητήρα και με ειδική συνάρτηση εξόδου μετατρέπεται η είσοδος σε ppm ή parts per million. Ο παραπάνω αισθητήρας λειτουργεί σε κύκλους τάσης από 1.5V για 90 δευτερόλεπτα και 5V για 90 δευτερόλεπτα για την αποφυγή επηρεασμού του αισθητήρα από άλλα αέρια εκτός του μονοξειδίου. Τέλος κανονική θερμοκρασία λειτουργίας είναι 20+-2 βαθμοί κελσίου και σε 65%+-5 υγρασία.[68][70]



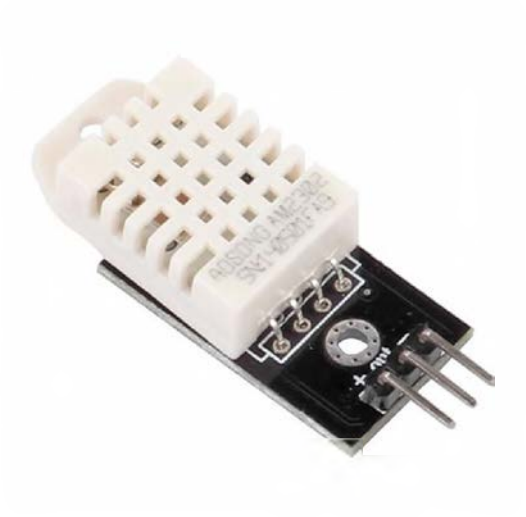
Διάγραμμα ευαισθησίας αερίων MQ7[68]

MQ7

5.2.3 Αισθητήρας Θερμοκρασίας-Υγρασίας DHT22

Ο DHT-22 είναι ένας χαμηλού κόστους αισθητήρας ανίχνευσης υγρασίας και μέτρησης θερμοκρασίας

με εύρος ανίχνευσης από 0%-100% για την υγρασία με ακρίβεια +- 2% και από -40 μέχρι 80 βαθμούς κελσίου με ακρίβεια +-0.5 βαθμούς κελσίου για την θερμοκρασία. Η τάση κανονικής λειτουργίας του κυμαίνεται στα 3.3-6 Volt και για την ανίχνευση των παραπάνω χρησιμοποιείται ένας πολυμερής πυκνωτής ο οποίος παίρνει τιμές το πολύ κάθε δύο δευτερόλεπτα. Ο συγκεκριμένος θα χρησιμοποιηθεί για την παρακολούθηση ή ακόμα και ρύθμιση των συνθηκών μέσα στον υπόγειο χώρο στάθμευσης εφόσον το κρίνει ο διαχειριστής και έχει τις υποδομές να το κάνει.[71].



DHT22

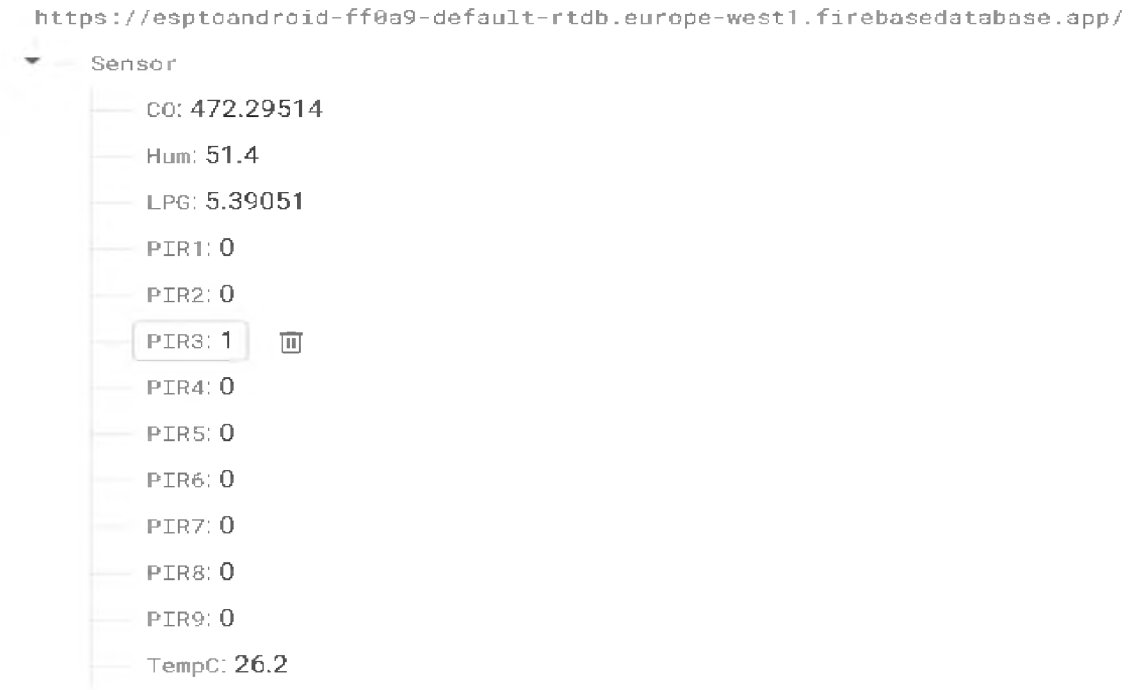
5.2.4 Αισθητήρας Εγγύτητας DC NO

Χρησιμοποιείται για την ανίχνευση αγώγιμων υλικών σε απόσταση όπως αλουμίνιο, χαλκός και ατσάλι. Είναι τύπου DC NO δηλαδή λειτουργεί με συνεχές ρεύμα και ο διακόπτης είναι ανοιχτός σε κατάσταση αδράνειας άρα δεν στέλνεται σήμα ανίχνευσης. Η τάση λειτουργίας του κυμαίνεται στα 12V, η κατανάλωση στα 300mA και η απόσταση ανίχνευσης στα 8 χιλιοστά.

Inductive Proximity Switch Sensor

5.3 Firebase

Η υπηρεσία του firebase λειτουργεί ως μεσάζοντας ανάμεσα από τον συλλέκτη δεδομένων ESP32 και της εφαρμογής android. Παρέχεται η δυνατότητα χρήσης βάσης δεδομένων πραγματικού χρόνου στην οποία τα δεδομένα αποθηκεύονται προσωρινά σε πτυχές- η κλαδιά-του δένδρου που ορίστηκε με όνομα Sensor και αντικαθίστανται με την άφιξη καινούριων στον επόμενο κύκλο ενημέρωσης δύο δευτερολέπτων.



Πίνακας ελέγχου Real-Time Database

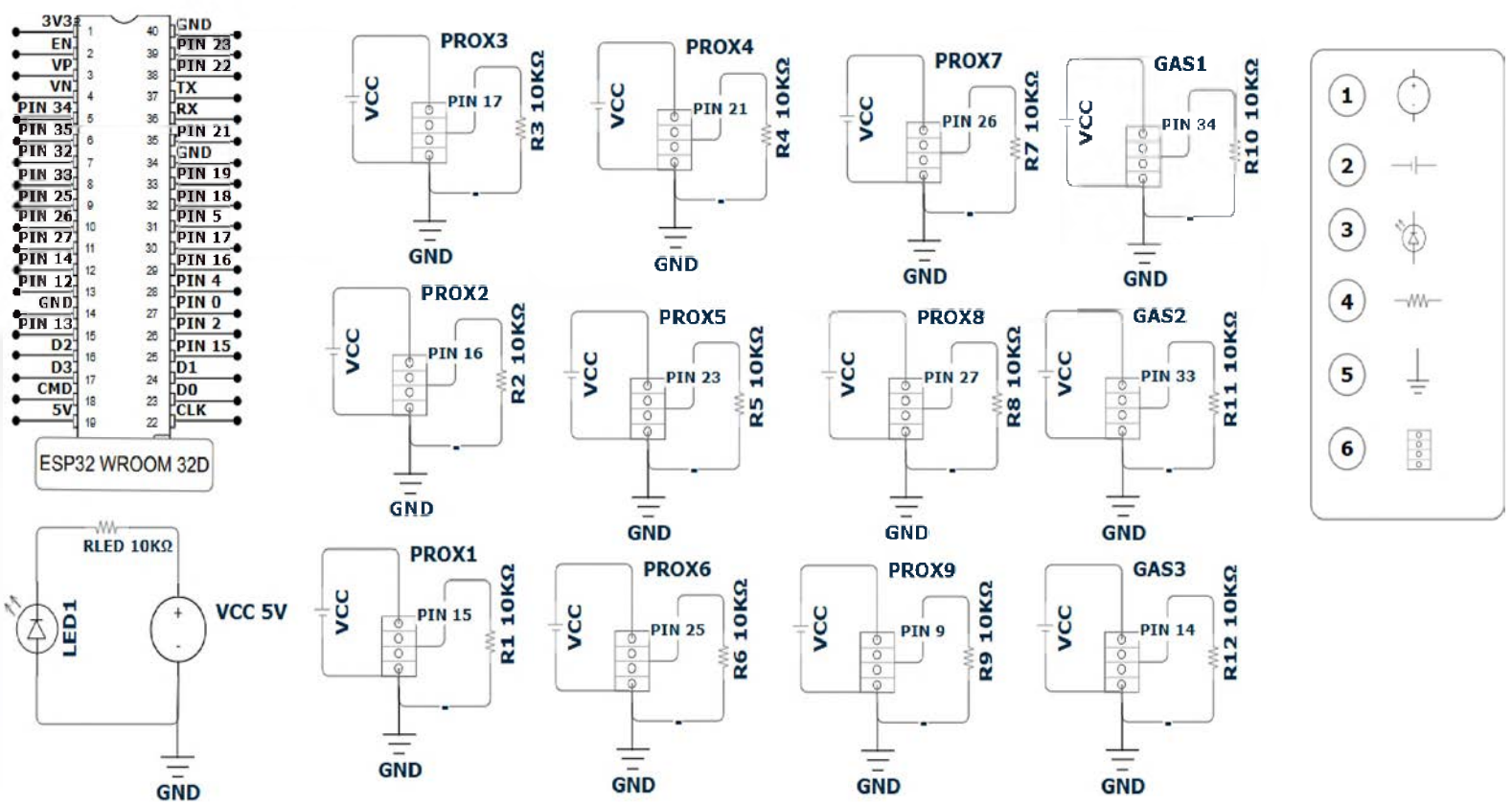
Όπως φαίνεται παραπάνω στην έχουμε τους αισθητήρες κίνησης ένα έως εννέα με την ονομασία "PIR", τους αισθητήρες που αποθηκεύουν τιμές σε ppm μονοξειδίου του άνθρακα με το αναγνωριστικό "CO"μαζί με τον αισθητήρα υγραερίου με όνομα "LPG" και τον συνδυαστικό αισθητήρα υγρασίας και θερμοκρασίας με τα ονόματα "Hum" και "Temp C" για ποσοστό υγρασίας και βαθμούς κελσίου αντίστοιχα.

Οι κανόνες που χρησιμοποιήθηκαν για την βάση επιτρέπουν την πρόσβαση από οποιονδήποτε είναι συνδεδεμένος στον λογαριασμό του και έχει πρόσβαση στα κλειδιά url του κατόχου.

```
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

5.4 Πλακέτα-υποδοχέας αισθητήρων και μικροελεγκτή

Για να μπορέσουν να λειτουργήσουν δώδεκα αισθητήρες ταυτόχρονα χρειάζεται σταθερή τάση 5V και περίπου 1A κάτι που το ESP32 WROOM 32D ΄δεν μπορεί να παρέχει αξιόπιστα από μόνο του. Αυτό οδηγεί τους αισθητήρες σε υπολειτουργία και πιθανή βλάβη κατά την χρήση τους. Για να λυθεί αυτό το πρόβλημα κατασκευάστηκαν δύο πλακέτες μία με υποδοχείς σήματος και τροφοδοσίας για κάθε έναν από τους αισθητήρες και μία για την τροφοδοσία μόνο των αισθητήρων εγγύτητας με βύσμα για εξωτερική τροφοδοσία από τροφοδοτικό. Δεύτερο πρόβλημα που λύνεται με την κατασκευή των πλακετών είναι η αποφυγή συνωστισμού των αισθητήρων και των καλωδίων που θα χρειάζονταν για την λειτουργία τους επιτυγχάνοντας μια πιο οργανωμένη και κατανοητή βάση για την εργασία.



5.4.1 Σχεδιάγραμμα Πλακέτας Υποδοχέων

Η δουλειά της παραπάνω πλακέτας είναι η φιλοξενία των τριών αισθητήρων αερίων και θερμοκρασίας-καθώς δουλεύουν με 5V και όχι 12V και η φιλοξενία των καλωδίων σήματος 3.3V δρομολογουμένων από την πλακέτα αισθητήρων εγγύτητας.

1. Τροφοδοτικό της τάξης των 5V και 3A το οποίο παρέχει σταθερή τάση στους υποδοχείς για την καλή λειτουργία των αισθητήρων.

2. Σύμβολο σύνδεσης με την πηγή τροφοδοσίας τάσης. Όπου υπάρχει αυτό υπάρχει και σύνδεση προς το τροφοδοτικό. Επισημαίνεται ότι οι υποδοχείς είναι σε παράλληλη ΄σύνδεση άρα έχουν την ίδια τάση στα άκρα τους.

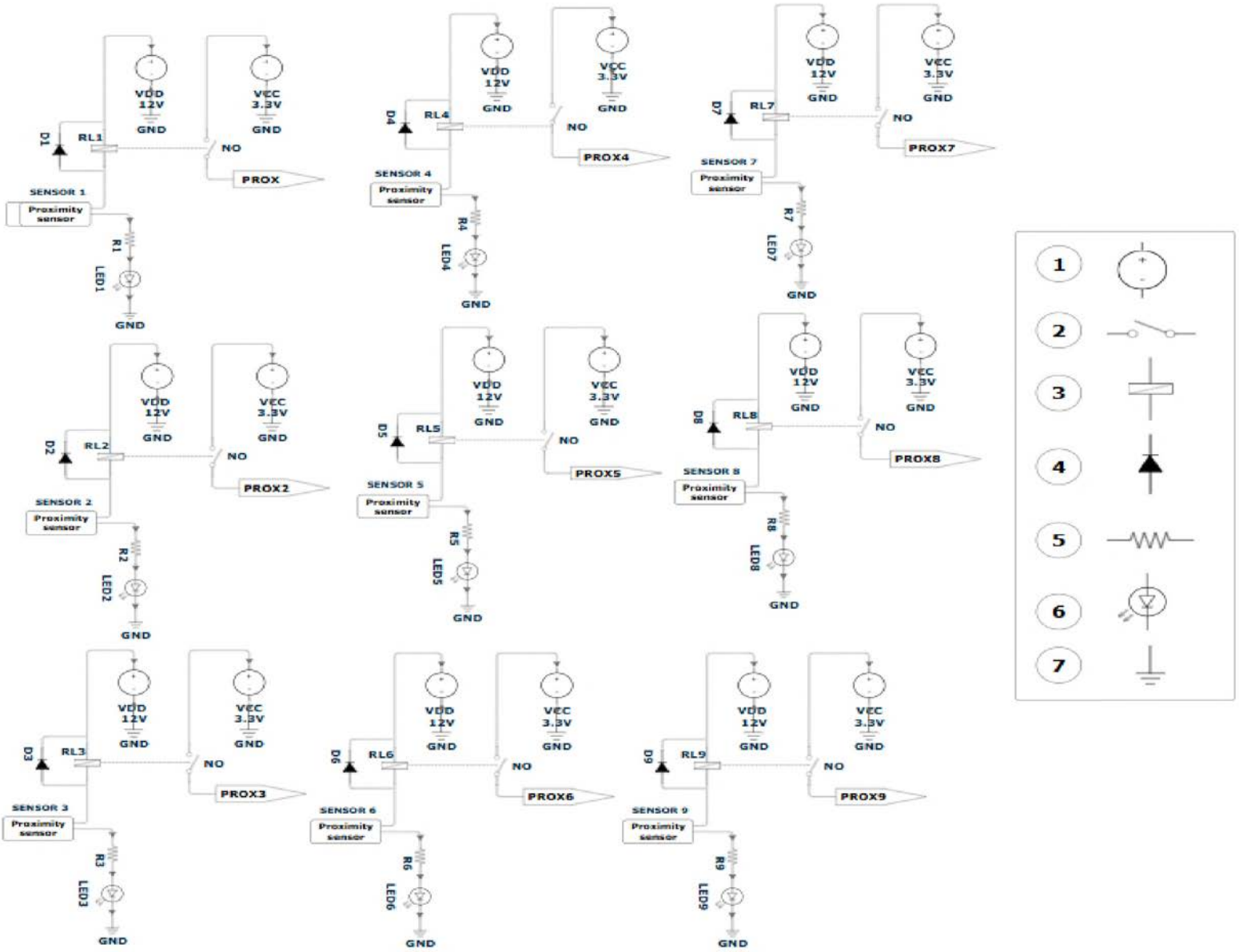
3. Δίοδος εκπομπής φωτός πράσινου χρώματος, σκοπός του οποίου είναι να ενημερώνει για το πότε υπάρχει τροφοδοσία στις πτυχές της πλακέτας και πότε όχι.

4.Αντίσταση της τάξης των 10ΚΩ . Μία χρειάζεται για την λειτουργία της διόδου φωτός για να μην καεί και μία για κάθε υποδοχέα με την ονομασία PROX 1 μέχρι PROX 9 . Δουλεία των προαναφερθέντων είναι η γείωση του PIN σήματος στα 0V έτσι ώστε να μην υπάρχουν παρεμβολές μεταξύ της τροφοδοσίας και αυτού. Τα PIN σήματος του ESP32 WROOM 32D έχουν εύρος τάσης από 0 έως 3.3V .όταν πρόκειται για το output των αισθητήρων GAS η τιμή αυτή θα ερμηνευτεί από τον μικροελεγκτή σαν ένα εύρος αριθμών από το 0 έως το 4095. Όταν η έξοδος του αισθητήρα είναι δυαδική όμως, το λογικό 0 αντιστοιχίζεται στα 0V και τα 3.3V στο το λογικό 1εντός κάποιων ορίων. Αν υπάρξει με κάποιο τρόπο παρασιτική τάση-δηλαδή εμπλακεί η τροφοδοσία με το σήμα- η έξοδος του αισθητήρα θα ερμηνεύεται μονίμως σαν 1. Επομένως χρησιμοποιείται η αντίσταση γείωσης για την αποφυγή του παραπάνω φαινομένου.

5.Σύμβολο της γείωσης. Όλες οι άκρες που έχουν αυτό το σύμβολο συνδέονται μεταξύ τους.

6.Υποδοχέας τριών θέσεων ο οποίος συνδέεται απευθείας με την γείωση και τροφοδοσία της πηγής για να κάνει εύκολη την σύνδεση των αισθητήρων. Κάθε ένας ανάλογα με τον συμβολισμό του καταλήγει και στο ανάλογο PIN πάνω στον μικροελεγκτή.

5.4.2 Σχεδιάγραμμα Πλακέτας Αισθητήρων



Σκοπός αυτής της πλακέτας είναι η φιλοξενία των εννέα αισθητήρων εγγύτητας με τάση λειτουργίας 12 V και να στέλνει τα σήματα τους στην πλακέτα υποδοχέων για τον προσδιορισμό κατάστασης των θέσεων.

1.Πηγή των 12V για την τροφοδοσία ειδικών αισθητήρων επαγωγής τύπου M18 .Για την τροφοδοσία των 3.3V χρησιμοποιείται διαιρέτης τάσης. Η πηγή των 3.3 V λειτουργεί ως το Output του αισθητήρα προς τον μικροελεγκτή. Δηλαδή αν το ESP32 λάβει 3.3V, το μεταφράζει ως λογικό 1 άρα ανιχνεύτηκε κίνηση. Εάν όμως λάβει 0V το μεταφράζει ως λογικό 0 άρα δεν ανιχνεύτηκε κίνηση.

2.Απλός διακόπτης τύπου NO δηλαδή Normally Open- Κανονικά Ανοιχτός.

3.Ρελέ με πηνίο της τάξης των 12V . Ρόλος του είναι το κλείσιμο του διακόπτη ΝΟ όταν ο αισθητήρας ανιχνεύσει κάποιο αγώγιμο υλικό στα 8 χιλιοστά.

4.Δίοδος για την αποφυγή

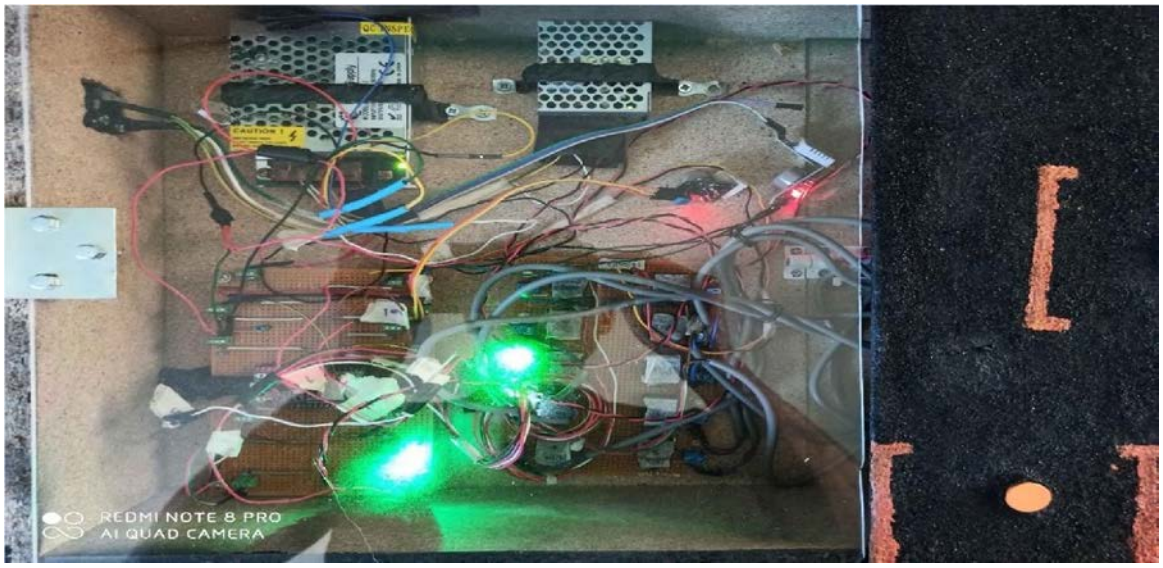
5.Αντίσταση των 10ΚΩ για την σωστή λειτουργία της φωτεινής διόδου LED.

6.Δίοδος τύπου LED πράσινου χρώματος. Δουλεία της είναι να ενημερώνει με το άναμμά της ότι ο αισθητήρας ανίχνευσε κίνηση άρα δουλεύει κανονικά.

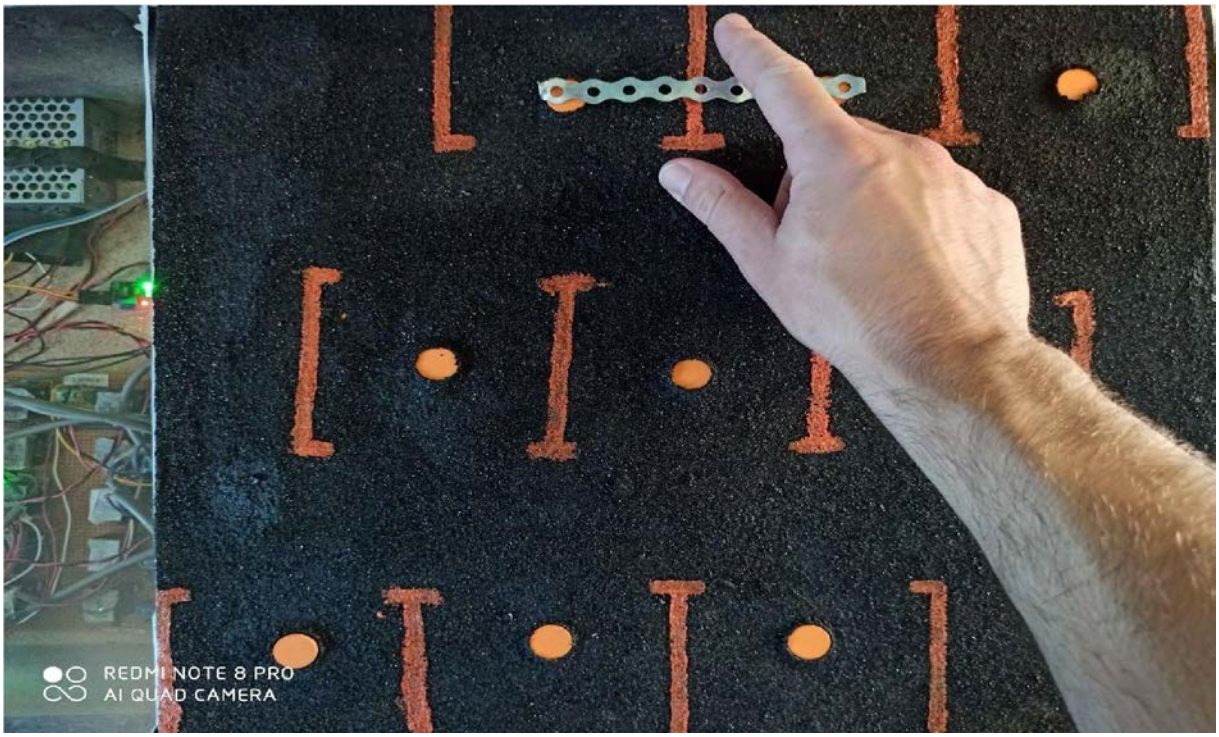
7.Γείωση.

5.5 Μακέτα

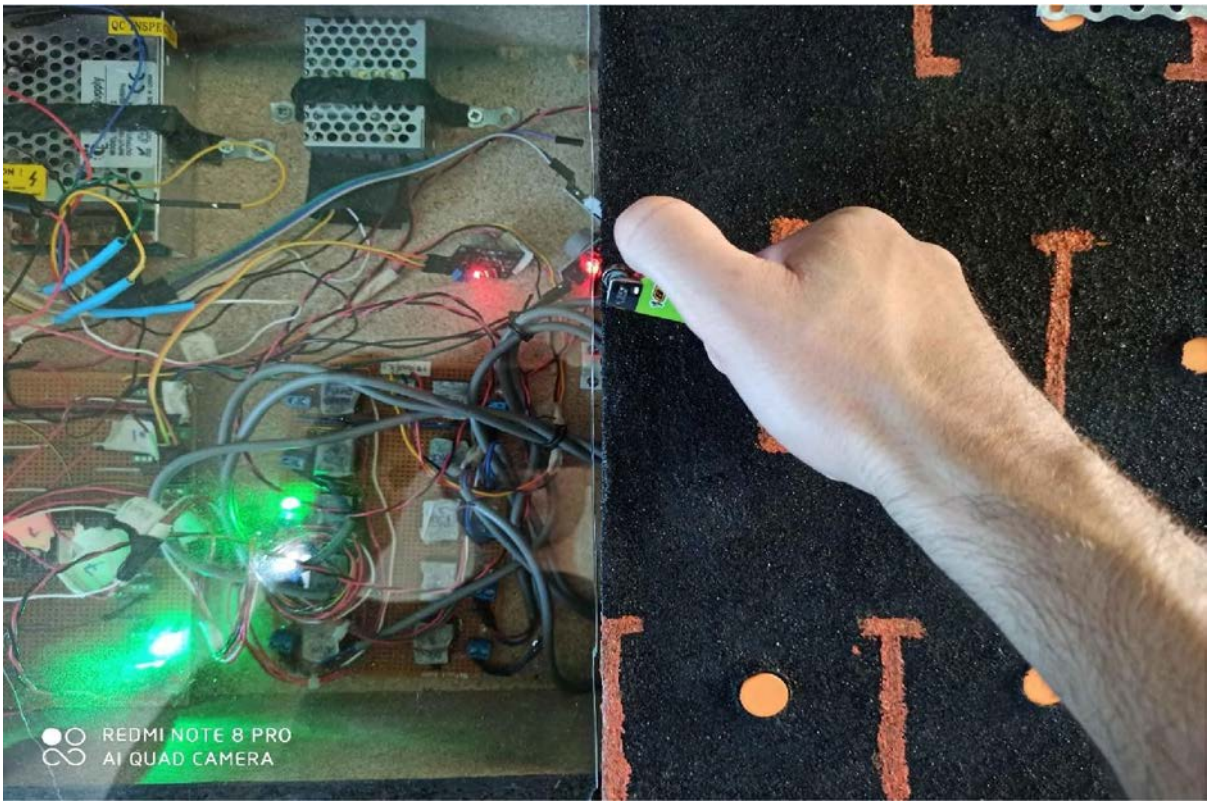
Η μακέτα λειτουργεί σαν δοχείο αποθήκευσης για τις πλακέτες, τους αισθητήρες και τα τροφοδοτικά καθώς και σαν αναπαράσταση του χώρου τον οποίο θέλει να προσομοιώσει. Έχει διαστάσεις 71 εκατοστά μήκος, 36 εκατοστά πλάτος και 10 εκατοστά ύψος. Για την πρόσβαση του χρήστη στις πλακέτες υπάρχει μία πόρτα από plexiglass με μεντεσέ από την πάνω μεριά της μακέτας(Σύστημα Πλακετών). Από την άλλη για την πρόσβαση στους αισθητήρες κίνησης χρησιμοποιείται μια πόρτα από την κάτω μεριά καθώς είναι στερεωμένοι στο "οδόστρωμα". Ο χρήστης μπορεί να ελέγξει την λειτουργία των αισθητήρων τοποθετώντας μεταλλικά αντικείμενα πάνω στους αισθητήρες εγγύτητας(Δοκιμή Κάλυψης Θέσεων) και χρησιμοποιώντας έναν αναπτήρα για προσομοίωση διαρροής υδρογονανθράκων και μονοξειδίου(Δοκιμή Υγραερίου-Δοκιμή Μονοξειδίου). Τα αποτελέσματα φαίνονται στην πλατφόρμα του firebase.



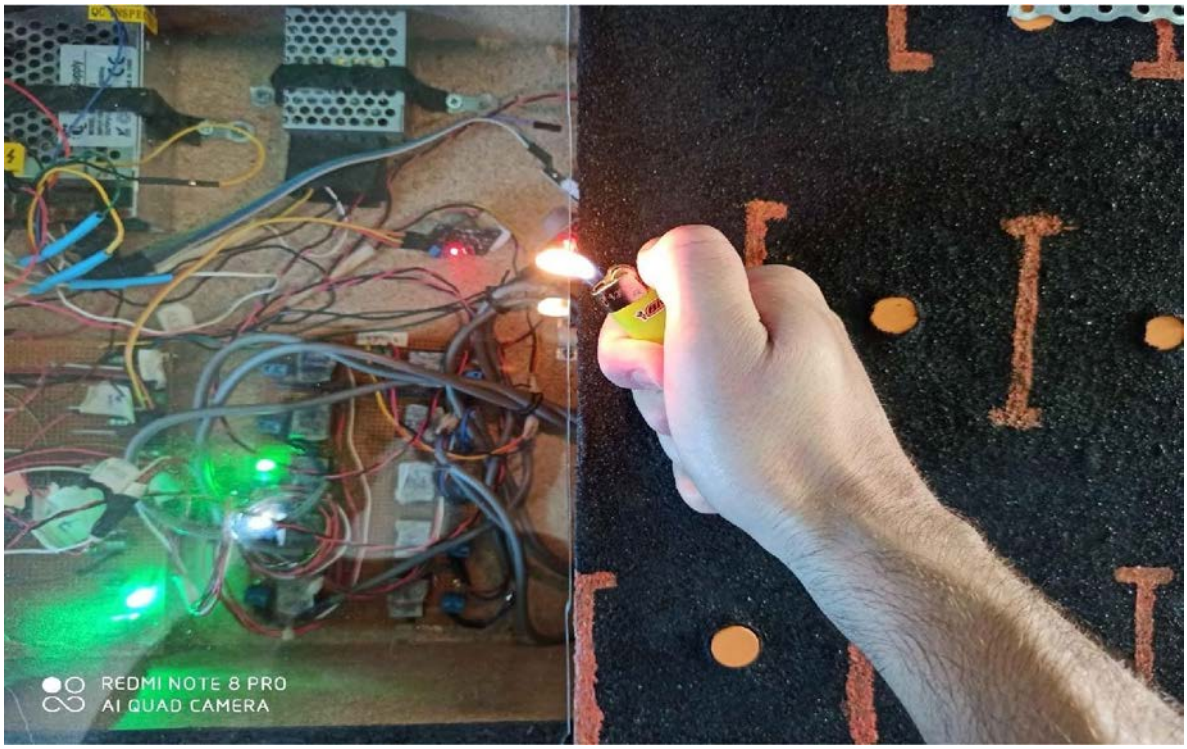
Σύστημα Πλακετών



Δοκιμή Κάλυψης Θέσεων



Δοκιμή Υγραερίου



Δοκιμή Μονοξειδίου

6 Συμπεράσματα

Η κατασκευή μιας εφαρμογής διαχείρισης ενός οποιουδήποτε χώρου απαιτεί να είναι και ταχεία και αξιόπιστη, καλύπτοντας τις ανάγκες του πελάτη και του διαχειριστή ταυτόχρονα. Παρά όλες τις απαιτήσεις ωστόσο, έχει επιτευχθεί η σταθερή και γρήγορη λειτουργία του πηγαίου κώδικα με λίγα έως ανύπαρκτα προβλήματα. Ο υποψήφιος πελάτης μπορεί να εισέλθει στην εφαρμογή, να δει την διαθεσιμότητα των θέσεων και να κάνει την παραγγελία του σε πολύ λογικό χρόνο. Η εφαρμογή καθοδηγεί τον πελάτη με την κατασκευή της από την εισαγωγή στον λογαριασμό του μέχρι το κλείσιμο της θέσης του με μηνύματα και κινούμενες εικόνες.

Ο ιδιοκτήτης από την άλλη μπορεί να δει τα επίπεδα υγρασίας, μονοξειδίου και διοξειδίου του άνθρακα μέσα στον χώρο καθώς και την κατάσταση των θέσεων στάθμευσης. Εμφανισιακά είναι επιεικώς ικανοποιητική και είναι εμφανές ότι η εφαρμογή με κάποιες τροποποιήσεις, κυρίως από μεριάς υλικού θα μπορούσε να χρησιμοποιηθεί από μία επιχείρηση η οποία θα έψαχνε έναν εύχρηστο τρόπο για την διαχείριση και επίβλεψη του χώρου της καθώς και για την εξυπηρέτηση των πελατών της. Όσον αφορά τον μικρό ελεγκτή και τη βάση δεδομένων ο ιδιοκτήτης έχει έναν πλήρως λειτουργικό επιτηρητή και εξυπηρετητή στην διάθεσή του, που χρειάζεται από λίγο έως καθόλου συντήρηση με την εξαίρεση του υπαλλήλου που θα το χρησιμοποιεί.

Αυτό που θα μπορούσε να βελτιώσει την εφαρμογή θα ήταν η απευθείας σύνδεση της με την τράπεζα για την πραγματοποίηση συναλλαγών, ώστε να μην χρειάζεται η προσομοίωσή τους. Για την επέκταση της βάσης δεδομένων SQL θα μπορούσαν να κρατηθούν λίστες με το πότε ένας πελάτης εισήλθε, εξήλθε, πόσο ξόδεψε και πότε έγιναν αυτές οι συναλλαγές. Ο ιδιοκτήτης θα μπορούσε να έχει πρόσβαση σε αυτές τις λίστες μέσω της βάσης δεδομένων αποθηκευμένα σε ένα αρχείο .σε ένα αρχείο csv.για λόγους ασφαλείας

Τέλος, για τον μικροελεγκτή ESP32 και τους αισθητήρες θα μπορούσε να φτιαχτεί μια ενιαία πλακέτα με υποδοχείς για την φιλοξενία αισθητήρων καλύτερης ποιότητας και ακρίβειας παίρνοντας υπόψιν τα λάθη που έγιναν στην αρχική κατασκευή. Η πλακέτα θα σχεδιάζονταν σε επαγγελματικό πρόγραμμα σχεδιασμού ηλεκτρονικών και θα υλοποιούνταν από εταιρία κατασκευής πλακετών κατά παραγγελία. Ευπρόσδεκτη θα

ήταν και η υλοποίηση συστήματος αυτόματου φωτισμού βασισμένη σε αισθητήρες κίνησης υπέρυθρων. Τέλος ένα σύστημα ταυτοποίησης οχημάτων με χρήση RFID θα μείωνε τον χρόνο αναμονής στις ουρές και θα καθιστούσε αχρείαστο τον υπάλληλο στην πύλη του χώρου καθώς συμβάλλει και στην ασφάλειά του.

Βιβλιογραφία

[1]RadioButton & RadioGroup Tutorial With Example In Android Studio (youtube.com))

[2]Android App Development in Java All-in-One Tutorial Series (4 HOURS!) (youtube.com)

[3]Internet of things - Wikipediahttps://en.wikipedia.org/wiki/Android_(operating_system)

[4]ARPANET - Wikipedia

[5]Android (operating system) - Wikipedia

[6] "CMU SCS Coke Machine Home Page". www.cs.cmu.edu. Retrieved 15 September 2024.

[7] "history_long.txt". ~coke. cs.cmu.edu. Retrieved 15 September 2024.

[8] "history_short.txt". ~coke. cs.cmu.edu. Retrieved 15 September 2024.

[9] "coke.history.txt". ~coke. cs.cmu.edu. Retrieved 15 September 2024.

[10] "The 'Only' Coke Machine on the Internet". cs.cmu.edu. Carnegie Mellon University. Retrieved 10 November 2014.

[11] "Internet of Things Done Wrong Stifles Innovation". InformationWeek. 7 July 2014. Retrieved 10 November 2014.

[12]F. Mattern, Floerkemeier, Christian (2010). "From the Internet of Computer to the Internet of Things" (PDF). Informatik-Spektrum. 33 (2): 107–121. Bibcode:2009InfSp..32..496H. doi:10.1007/s00287-010-0417-7. hdl:20.500.11850/159645. S2CID 29563772. Retrieved 3 February 2014.

[13] M. Weiser, (1991). "The Computer for the 21st Century" (PDF). Scientific American. 265 (3): 94–104. Bibcode:1991SciAm.265c..94W. doi:10.1038/scientificamerican0991-94. Archived from the original (PDF) on 11 March 2015. Retrieved 5 November 2014.

[14] R.S .Raji, (1994). "Smart networks for control". IEEE Spectrum. 31 (6): 49–55. doi:10.1109/6.284793. S2CID 42364553.

[15] J. Pontin, (29 September 2005). "ETC: Bill Joy's Six Webs". MIT Technology Review. Retrieved 17 November 2013.

[16]K. Lakhwani, (2020). Internet of Things (IoT) : Principles, Paradigms and Applications of IoT. Hemant Kumar Gianey, Joseph Kofi Wireko, Kamal Kant Hiran. [Place of publication not identified]. ISBN 9789389423365. OCLC 1188989203.

[17]K. Ashton, (22 June 2009). "That 'Internet of Things' Thing". rfidjournal.com. Archived from the original on 15 April 2013. Retrieved 9 May 2017.

[18] P. Magrassi, (2 May 2002). "Why a Universal RFID Infrastructure Would Be a Good Thing". Gartner research report G00106518.

[19] P. Magrassi, Berg, T (12 August 2002). "A World of Smart Objects". Gartner research report R-17-2243. Archived from the original on 3 October 2003.

[20] "Internet of Things – An action plan for Europe" (PDF). ec.europa.eu. Commission of the European Communities. 18 June 2009. COM(2009) 278 final.

[21] A. Wood, (31 March 2015). "The internet of things is revolutionizing our lives, but standards are a must". The Guardian.

[22] W. Stallings, (2016). Foundations of modern networking : SDN, NFV, QoE, IoT, and Cloud. Florence Agboma, Sofiene Jelassi. Indianapolis. ISBN 978-0-13-417547-8. OCLC 927715441.

[23] "StackPath". Industryweek.com. 21 December 2004. Retrieved 20 May 2022.

[24] D. Evans (April 2011). "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything" (PDF). CISCO White Paper.

[25]"Google's Android OS: Past, Present, and Future". PhoneArena. August 18, 2011. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[26] B. Elgin, (August 17, 2005). "Google Buys Android for Its Mobile Arsenal". Bloomberg Businessweek. Archived from the original on February 5, 2011. Retrieved March 12, 2017.

[27] J. Alabaster, (April 16, 2013). "Android founder: We aimed to make a camera OS". PC World. International Data Group. Archived from the original on May 10, 2017. Retrieved May 9, 2017.

[28] C. Welch, (April 16, 2013). "Before it took over smartphones, Android was originally destined for cameras". The Verge. Vox Media. Archived from the original on April 29, 2017. Retrieved May 9, 2017.

[29]L. Eadicicco, (March 27, 2015). "THE RISE OF ANDROID: How a flailing startup became the world's biggest computing platform". Business Insider. Axel Springer SE. Archived from the original on May 20, 2017. Retrieved May 9, 2017.

[30] A. Vance, (July 29, 2011). "Steve Perlman's Wireless Fix". Bloomberg Businessweek. Archived from the original on March 19, 2017. Retrieved March 12, 2017.

[31] McAfee, Andrew; Brynjolfsson, Erik (2017). Machine, Platform, Crowd : Harnessing Our Digital Future. New York: W.W. Norton. p. 166. ISBN 978-0-393-25429-7. OCLC 987909505.

[32]C. Haase, (August 13, 2021). "Excerpt: How Google bought Android—according to folks in the room". Ars Technica. Archived from the original on August 13, 2021. Retrieved August 13, 2021.

[33]F. Manjoo, (May 27, 2015). "A Murky Road Ahead for Android, Despite Market Dominance". The New York Times. Archived from the original on July 6, 2017. Retrieved March 12, 2017.

[34] R. Block, (August 28, 2007). "Google is working on a mobile OS, and it's due out shortly". Engadget. AOL. Archived from the original on March 12, 2017. Retrieved March 11, 2017.

[35] A. Sharma, Delaney, K.J. (August 2, 2007). "Google Pushes Tailored Phones To Win Lucrative Ad Market". The Wall Street Journal. Archived from the original on July 29, 2017. Retrieved July 24, 2017.

[36] M. McKay, (December 21, 2006). "Can iPhone become your phone?; Linksys introduces versatile line for cordless service". The Record (Bergen County). p. L9. Archived from the original on May 30, 2012. Retrieved February 21, 2012. And don't hold your breath, but the same cell phone-obsessed tech watchers say it won't be long before Google jumps headfirst into the phone biz. Phone, anyone?

[37] D. Ionescu, (April 26, 2012). "Original Android Prototype Revealed During Google, Oracle Trial". PC World. International Data Group. Archived from the original on February 11, 2017. Retrieved March 12, 2017.

[38] C. Ziegler,(April 25, 2012). "This was the original 'Google Phone' presented in 2006". The Verge. Vox Media. Archived from the original on March 25, 2017. Retrieved March 12, 2017.

[39] D. Aamoth, (September 23, 2008). "T-Mobile officially announces the G1 Android phone". TechCrunch. AOL. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[40] R. Gao, (September 23, 2016). "Android and its first purchasable product, the T-Mobile G1, celebrate their 8th birthdays today". Android Police. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[41] "Industry Leaders Announce Open Platform for Mobile Devices". Open Handset Alliance. November 5, 2007. Archived from the original on March 9, 2012. Retrieved March 12, 2017.

[42]E. Schonfeld, (November 5, 2007). "Breaking: Google Announces Android and Open Handset Alliance". TechCrunch. AOL. Archived from the original on June 22, 2017. Retrieved March 12, 2017.

[43]A. Rubin, (November 5, 2007). "Where's my Gphone?". Official Google Blog. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[44] T. Claburn, (September 19, 2007). "Google's Secret Patent Portfolio Predicts gPhone". InformationWeek. Archived from the original on March 17, 2008. Retrieved March 12, 2017.

[45] P.J Quintana (September 20, 2007). "Google's Strong Mobile-Related Patent Portfolio". Gigaom. Knowingly, Corp. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[46] T-Mobile launches G1, first Google Android phone – Video, retrieved November 21, 2023

[47] M. Menon, K. Murali , (July 3, 2016). "Android Nougat: Here's why Google names the OS after sweets". The Indian Express. Indian Express Limited. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[48]F. Ion, (May 15, 2013). "From Nexus One to Nexus 10: a brief history of Google's flagship devices". Ars Technica. Condé Nast. Archived from the original on June 24, 2017. Retrieved March 12, 2017.

[49]M. Smith, (August 28, 2013). "Android VP Hugo Barra leaves Google, joins Chinese phone maker Xiaomi (updated)". Engadget. AOL. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[50] E. Orion, (August 28, 2013). "Google's Android VP Hugo Barra joins Chinese phone maker Xiaomi". The Inquirer. Incisive Media. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[51] L. Page, (March 13, 2013). "Update from the CEO". Official Google Blog. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[52] C. Arthur, (March 13, 2013). "Andy Rubin moved from Android to take on 'moon-shots' at Google". The Guardian. Archived from the original on March 12, 2017. Retrieved March 12, 2017.

[53]R. Brandom, (August 10, 2015). "Google is reorganizing and Sundar Pichai will become new CEO". The Verge. Vox Media. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[54]J. Conditt, (August 10, 2015). "Google gets an overhaul and a new CEO: Sundar Pichai". Engadget. AOL. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

[55]M. Bergen, (October 9, 2015). "New Google CEO Sundar Pichai Makes First Major Executive Picks". Recode. Vox Media. Archived from the original on January 14, 2017. Retrieved March 12, 2017.

[56]A. Martonik, (October 9, 2015). "Sundar Pichai promotes Hiroshi Lockheimer to oversee Android, Chrome OS and Chromecast". Android Central. Mobile Nations. Archived from the original on February 23, 2017. Retrieved March 12, 2017.

[57] T.C Sottek, (May 19, 2019). "Google pulls Huawei's Android license, forcing it to use open source version". The Verge. VOX Media. Retrieved July 20, 2019. A dramatic escalation in the US war on Chinese tech firms

[58]J. Cook, (May 20, 2019). "Google restricts Huawei from using Android: Here's what that could mean for you". Technology Intelligence. The Telegraph. Archived from the original on January 10, 2022. Retrieved July 20, 2019. Huawei, which is the world's second largest seller of smartphones after Samsung, has long relied on Google's Android operating system to run its smartphones and tablets. The ban means that new Huawei phones will no longer be able to access certain apps, such as Google Maps and YouTube, and existing phones will not be able to update their Android operating systems.

[59] C. Reichert, (June 14, 2019). "Huawei moves to trademark its own OS while objecting to US ban". Tech News. CNET. Retrieved August 10, 2019. Huawei is moving to trademark the name of its operating system, "Hongmeng," in Peru.

[60] O. Sohail, (May 20, 2019). "Huawei's Own Smartphone Operating System Reportedly Named HongMeng OS, According to Foreign Sources". Mobile tech. Where Consumers Come First (Wccf). Retrieved August 10, 2019.

[61]Y. Jie, D. Strumpf, (May 24, 2019). "Who Needs Google's Android? Huawei Trademarks Its Own Smartphone OS". Tech. The Wall Street Journal. Retrieved August 10, 2019. Chinese tech giant plans to launch its own operating system this year as access to U.S. software is hit by export ban

[62] J. England, (June 14, 2019). "Huawei begins trademarking its Android replacement OS—HongMeng". Android Central. Retrieved August 10, 2019. The trademark's been filed in Canada, the European Union, Mexico, and more.

[63] J. Porter, (August 9, 2019). "Huawei's new operating system is called HarmonyOS". The Verge. Retrieved August 9, 2019.

[64]U. Shakir, (December 10, 2021). "Go read this story explaining in detail the scary Teams bug that blocked a 911 call". The Verge. Retrieved December 7, 2022.

[65] "This important Microsoft Teams for Android update fixes the strange 911 calling bug". ZDNET. Retrieved December 7, 2022.

[66]K. Raj, (January 10, 2017). The Internet is a vast global network of connected servers, computers, tablets and mobiles that is governed by standardprotocols for connected systems. It enables sending, receiving, or communication of information, connectivity with remote servers, cloud and analytics platforms. INTERNET OF THINGS Architecture and Design Principles. Retrieved January 5, 2025.

[67]https://components101.com/sites/default/files/component_datasheet/MQ-6%20Gas%20Sensor%20Datasheet.pdf

[68] https://cdn.sparkfun.com/assets/b/b/b/3/4/MQ-7.pdf

[69] draw.io.com

[70] https://www.circuits-diy.com/mq7-carbon-monoxide-co-gas-sensor-module/

[71] https://cdn.sparkfun.com/assets/f/7/d/9/c/DHT22..

Παράρτημα Α

- https://github.com/FotiosGiannouglidis/AutoParking/tree/main