

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό
κατάστημα**

Σταύρος Γιούτσος

Εισηγητής: Δρ. Ιωάννης Έλληνας, Καθηγητής

**ΑΘΗΝΑ
ΦΕΒΡΟΥΑΡΙΟΣ 2017**

Τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα
Σταύρος Γιούτσος
Α.Μ. 43085**

Εισηγητής:

Δρ Ιωάννης Έλληνας, Καθηγητής

Ημερομηνία εξέτασης 24/2/2017

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/ηΓιούτσος Σταύρος.....,
τουΙωάννη....., με αριθμό μητρώου
.....43085..... φοιτητής/τρια του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε.
του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου,
δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια
πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα
πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο
συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και
άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το
Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης
του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από
αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο
θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε.
πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την
ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο
άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε ύστερα από πολύ κόπο, υπομονή και επιμονή σε ένα ενδιαφέρον πεδίο, όπως αυτό της τηλεξυπηρέτησης πωλήσεων. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω ιδιαίτερα για τις συμβουλές του και την γενικότερη επικοινωνία που είχαμε καθ' όλη την διάρκεια μέχρι το πέρας της πτυχιακής μου.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου που με στήριξε σε κάθε μου βήμα και είναι περήφανη για την πορεία μου και τα επιτεύγματα μου μέχρι σήμερα.

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία αφορά την τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα χωρίς την φυσική παρουσία υπαλλήλων. Αυτό το πεδίο έχει ήδη ξεκινήσει να αναπτύσσεται με ταχύς ρυθμούς σε όλο τον κόσμο και αυτό γιατί αφορά το μέλλον του σύγχρονου marketing καθώς με την ταχεία ανάπτυξη της τεχνολογίας λιγότευει η αναγκαιότητα φυσικής παρουσίας ανθρώπων. Η μελέτη επικεντρώνεται στους τρόπους με τους οποίους μπορεί ένας πελάτης να εξυπηρετηθεί σε ένα φυσικό κατάστημα, για περιήγηση ή για λοιπές ανάγκες του, χωρίς την παρουσία ανθρώπινου δυναμικού στο χώρο.

ABSTRACT

The present thesis concerns the service provided to customers in real-time in a physical store without the need of natural presence of employees. This field has already started developing vastly around the world and that is because it is the future of all shop since with the technological advance the necessity of natural presence of employees is becoming less important as time goes by. This study focuses on the means that can achieve the goal of a customer getting assisted on a physical store for his need without natural presence of employees.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Τηλεξυπηρέτηση σε πραγματικό χρόνο

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: τηλεπικοινωνία, φυσικό μαγαζί, πραγματικός χρόνος, Internet, φυσική παρουσία

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΕΙΣΑΓΩΓΗ.....	1
1.1	Περιγραφή του αντικειμένου της πτυχιακής εργασίας	1
1.2	Ιστορική αναδρομή	1
2.	ΚΑΤΑΣΤΗΜΑΤΑ.....	3
2.1	Ορισμός	3
2.2	Ιστορική αναδρομή	3
2.3	Μορφές καταστημάτων	4
2.3.1	Φυσικά	4
2.3.2	E-shop	5
3.	ΕΞΥΠΗΡΕΤΗΣΗ ΠΕΛΑΤΩΝ.....	7
3.1	Ορισμός	7
3.2	Μορφές εξυπηρέτησης	7
3.2.1	Διαπροσωπική	7
3.2.2	Τηλέφωνο	8
3.2.3	E-mail	8
4.	ΠΡΑΓΜΑΤΙΚΟΣ ΧΡΟΝΟΣ.....	9
4.1	Ορισμός	9
4.2	Εφαρμογές	9
5.	Arduino & Wi-fi Shield.....	11
5.1	Η λογική των μικροεπεξεργαστών Arduino	11
5.2	Χαρακτηριστικά του μικροεπεξεργαστή Arduino Uno	12
5.3	Η λογική των Shield	13
5.4	Adafruit CC 3000 Wi-Fi Shield	14
5.5	Προγραμματισμός για το Adafruit CC3000 Wi-Fi Shield	15
6.	ΚΑΤΑΣΤΗΜΑ ΧΩΡΙΣ ΥΠΑΛΛΗΛΟΥΣ.....	17
6.1	Μορφή και διάταξη	17

6.2	Απαραίτητες προϋποθέσεις	18
6.3	Το ειδικό περιβραχιόνιό	21
6.4	Το ηλεκτρικό scooter	21
6.5	Η διεπαφή του οχήματος	22
7.	ΕΦΑΡΜΟΓΗ.....	23
7.1	Γενική περιγραφή	23
7.2	Σύνδεση με Wi-fi	23
7.3	Εξαρτήματα και κατασκευή	25
7.4	Ηλεκτρικό κύκλωμα – τροφοδοσία	27
7.5	Ηλεκτρονικό κύκλωμα	27
7.6	Πρόγραμμα Arduino	28
7.7	Interface	30
7.8	Πρόγραμμα κινητού	30
8.	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	33
8.1	Σύνοψη της πτυχιακής εργασίας	33
8.2	Προβλήματα κατά την υλοποίηση	33
8.3	Προοπτικές	34
9.	ΠΑΡΑΡΤΗΜΑ Α'.....	37
10.	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	47

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 6.2.1: Διάγραμμα ροής του πελάτη στο κατάστημα	19
Σχήμα 6.4.1: Πρωτότυπο ηλεκτρικού scooter με ενσωματωμένη διεπαφή και καλάθια πάνω από τις ρόδες	21
Σχήμα 7.4.1: Ηλεκτρικό κύκλωμα	27
Σχήμα 7.5.1: Ηλεκτρονικό κύκλωμα	28

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.2.1: Αυτόματος πωλητής	1
Εικόνα 2.2.1: Μαγαζί στην αρχαία Ρώμη	3
Εικόνα 2.3.2.1: Κατάστημα E-Shop	5
Εικόνα 4.2.1: Παρατήρηση χρηματιστηρίου σε πραγματικό χρόνο	9
Εικόνα 5.2.1: Arduino Uno	13
Εικόνα 5.3.1: Sparkfun MP3 Player Shield	14
Εικόνα 5.4.1: Adafruit CC3000 Wi-Fi Shield & Breakout Board	15
Εικόνα 6.1.1: Ηλεκτρικό Scooter Segway	17
Εικόνα 6.5.1: Παράδειγμα διεπαφής	22
Εικόνα 7.2.1: Http request χρήστη	24
Εικόνα 7.2.2: Http request server	24
Εικόνα 7.2.3: Http request κίνησης μπροστά και σταματήματος του οχήματος .	24
Εικόνα 7.3.1: Robot Chassis	26
Εικόνα 7.3.2: Κατασκευή με χρήση Arduino & Wi-Fi Shield	26
Εικόνα 7.7.1: Σελίδα ελέγχου της κατασκευής	30
Εικόνα 7.8.1: Στιγμιότυπο οθόνης κινητού	32
Εικόνα 7.8.2: Στιγμιότυπο διεπαφής ελέγχου	32

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ICSP In-Circuit Serial Programming

I/O Input / Output

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της πτυχιακής εργασίας και γίνεται μια ιστορική αναδρομή γύρω από τις μεθόδους που έχουν παρουσιαστεί σε αυτήν την περιοχή.

1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι τηλεξυπηρέτηση πελατών σε πραγματικό χρόνο. Με απλά λόγια εξετάζονται οι μέθοδοι και τα μέσα που χρειάζονται έτσι ώστε ένα μαγαζί να λειτουργεί με ελάχιστο ή και καθόλου προσωπικό και ένα πελάτης να μπορεί παρόλα ταύτα να εξυπηρετείται σε πραγματικό χρόνο. Να μπορεί, δηλαδή ο πελάτης, να κάνει ερωτήσεις για προϊόντα και να κάνει περιήγηση σε χώρο καταστήματος με την ανάλογη καθοδήγηση μέσω ενός συστήματος που θα δίνει την δυνατότητα σε έναν υπάλληλο να δουλεύει εκτός χώρου. Με την επικοινωνία να είναι σε πραγματικό χρόνο επισημαίνεται πως η διαπροσωπική επικοινωνία δεν χάνεται, αλλά αλλάζει μορφή.

1.2 Ιστορική αναδρομή

Τα πρώτα δείγματα συστημάτων που δίνουν την δυνατότητα στους πελάτες να εξυπηρετηθούν χωρίς την χρήση υπαλλήλων ήταν οι ευρέως πλέον διαδεδομένοι αυτόματοι πωλητές (vending machines).



Εικόνα 1.2.1: Αυτόματος πωλητής

Οι σύγχρονοι αυτόματοι πωλητές παρουσιάστηκαν για πρώτη φορά στις αρχές του 1880 από τον Percival Everitt και γρήγορα άρχισαν να εμφανίζονται σε σημεία ενδιαφέροντος όπως σταθμούς τρένων.[1]

Αφαίρεσαν έτσι, για πρώτη φορά την διαπροσωπική επικοινωνία , χωρίς να ξεφεύγουμε πάντα από το φυσικό μέρος, και ήταν το πρώτο στάδιο που σηματοδοτούσε το μέλλον των καταστημάτων και της τεχνολογίας.

Συστήματα βέβαια όπως αυτό που μελετάμε δεν έχουν κάνει ακόμα την επίσημη εμφάνιση τους. Υπάρχουν παρεμφερή συστήματα που έχουν κάνει την εμφάνιση τους τόσο στην Αμερική όσο και στην Σουηδία με ταχύς ρυθμούς ανάπτυξης αλλά όχι κάτι που να αντιπροσωπεύει σε όλο το μήκος κύματος το σύστημα που θέλουμε να περιγράψουμε.

ΚΕΦΑΛΑΙΟ 2

ΚΑΤΑΣΤΗΜΑΤΑ

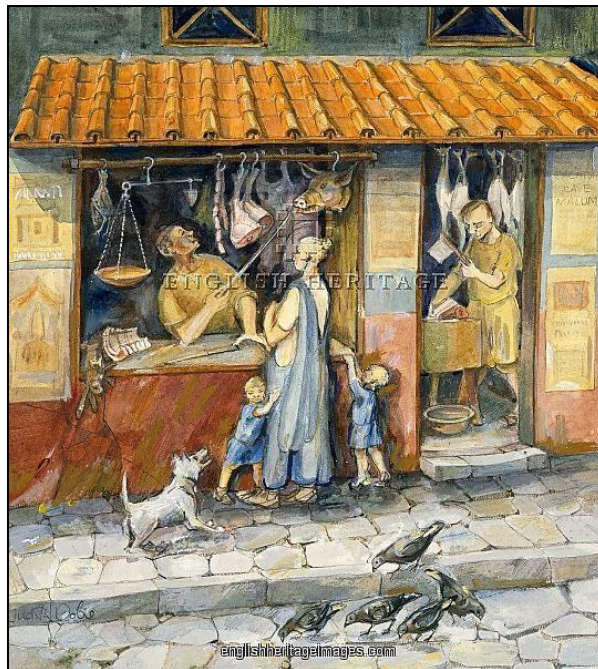
Σε αυτό το κεφάλαιο αναλύονται τα καταστήματα. Αναλύεται η ιστορική διαδρομή μέχρι τα σημερινά καταστήματα και το πως αυτά λειτουργούν.

2.1 Ορισμός

Ως ένα κατάστημα ορίζεται ο κλειστός χώρος στον οποίο εκτίθενται εμπορεύματα προς πώληση. Στον χώρο υπάρχουν συνήθως υπάλληλοι που βοηθούν στην πιο γρήγορη εξυπηρέτηση των πελατών καθώς και το ταμείο όπου αφού αποφασίσουν τα αγαθά τα οποία θέλουν να προμηθευτούν μπορούν να προβούν στην αγορά τους με το αντίστοιχο νόμισμα που ορίζει η εκάστοτε χώρα και να λάβουν την απόδειξη πληρωμής τους.

2.2 Ιστορική Αναδρομή

Η ιστορία των καταστημάτων μας πάει πολύ πίσω και ξεκινάει από την αρχαία Ρώμη όπου εκεί έγινε εκτενής η χρήση μαγαζιών για πώληση κρέατος, μαναβικής και όπλων όπως βέλη και τόξα. Τα καταστήματα αυτά ήταν απλά στην δομή τους από ξύλο ενώ το γυαλί ήταν προνόμιο για εκείνα τα χρόνια.[2]



Εικόνα 1.2.1: Μαγαζί στην αρχαία Ρώμη

Με την πάροδο των χρόνων κάθε πόλη είχε πλέον από ένα τουλάχιστον κατάστημα φτάνοντας στον 17^ο αιώνα όπου οι αριθμοί των καταστημάτων ανέβηκαν ραγδαία.[2]

Το 18^ο αιώνας ήταν μια σημαντική περίοδος ανάπτυξης των καταστημάτων με πολλές εφευρέσεις και αλλαγές στο μέχρι τότε τοπίο. Συγκεκριμένα, η τιμή του γυαλιού έπεσε δραματικά και έτσι τα μαγαζιά ξεκίνησαν να έχουν στις βιτρίνες τους γυαλί. Το 1852 εφευρέθηκε για πρώτη φορά μια μηχανή δημιουργίας χαρτοσακούλων από τον Francis Rolle και το 1871 η Margaret Knight εφηύρε μια μηχανή δημιουργίας χαρτοσακούλων με ίσιο πάτο. Η ταμειακή μηχανή επίσης εφευρέθηκε και άμεσα καθιερώθηκε από τον James Ritty το 1879.[2]

Το 1930 άνοιξε το πρώτο πολυκατάστημα στην Αμερική και το 1937 εφευρέθηκε το πρώτο καρτσάκι αγορών. Το 1966 στην Αγγλία εμφανίστηκε η πρώτη πιστωτική κάρτα και το 1978 χρησιμοποιήθηκε για πρώτη φορά ο κωδικός προϊόντος(barcode) ενώ το πρώτο πράγμα που πέρασε από μηχανήμα σκαναρίσματος κωδικού προϊόντος ήταν ένα πακέτο τσαγιού.[2]

Τέλος, τα πρώτα ηλεκτρονικά καταστήματα κάνουν την εμφάνιση τους τον 21 αιώνα με αυξανόμενους ρυθμούς για την ευκολία που παρέχουν σε απομακρυσμένες και μη περιοχές να μπορούν να προμηθεύονται οποιοδήποτε αγαθό κατευθείαν στο σπίτι τους.

2.3 Μορφές καταστημάτων

Τα καταστήματα στη σύγχρονη εποχή μπορούμε να τα διακρίνουμε σε 2 κύριες κατηγορίες σχετικά με τον τρόπο που ο πελάτης προμηθεύεται και παραλαμβάνει τα αγαθά του.

2.3.1 Φυσικά καταστήματα

Τα φυσικά καταστήματα είναι τα καταστήματα που εξ ορισμού έχουν φυσική-υλική υπόσταση. Είναι αυτά τα οποία υπήρχαν ανέκαθεν από τα αρχαία χρόνια στα οποία ο πελάτης πρέπει ο ίδιος με φυσική παρουσία, παρευρισκόμενος δηλαδή στον χώρο, να επιλέξει τα αγαθά που θέλει.

Αυτή η μορφή είναι καθαρά διαπροσωπική καθώς ο πελάτης έρχεται σε επαφή τόσο με άλλους πελάτες που παρευρίσκονται στο χώρο όσο και με το προσωπικό του μαγαζιού, που είναι έτοιμο να του λύσει ακόμα και την πιο μικρή απορία ή να τον ενημερώσει άμεσα για κάποια μορφή προσφοράς προϊόντων.

2.3.2 E-shop καταστήματα

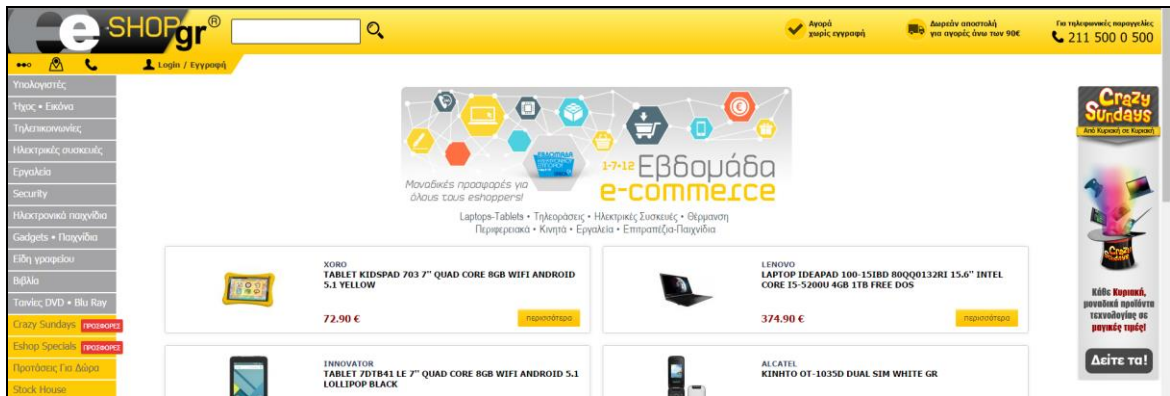
Τα καταστήματα E-shop είναι τα καταστήματα νέας γενιάς που δεν έχουν φυσική υπόσταση και παρέχουν στον πελάτη την δυνατότητα να προμηθευτεί αυτό που ζητάει μέσω Internet από το σπίτι του ή από οποιοδήποτε άλλο μέρος όπως επίσης και να του σταλούν σε όποιο μέρος εκείνος επιθυμεί την ίδια ή τις επόμενες μέρες.

Τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

Αυτή η μορφή καταστημάτων χάνει μεν τον διαπροσωπικό χαρακτήρα που είχαν οι συναλλαγές μέχρι σήμερα αλλά προσφέρει μια πληθώρα άλλων πλεονεκτημάτων σε αντάλλαγμα.

Όπως αναφέρθηκε και παραπάνω ο πελάτης πλέον ,με την μόνη προϋπόθεση να έχει σύνδεση στο Internet, δεν χρειάζεται να παρευρεθεί στο χώρο του, το οποίο σημαίνει ότι του εξοικονομείται χρόνος είτε για κάποια άλλη δουλειά, είτε για να περάσει χρόνο με την οικογένεια του ή για κάποιο χόμπι του. Η αναζήτηση των προϊόντων που θέλει ο πελάτης γίνεται πολύ πιο γρήγορα καθώς μέσω των διάφορων πλατφόρμων αναζήτησης που παρέχει κάθε κατάστημα E-shop μπορεί άμεσα και γρήγορα να βρει ακριβώς αυτό που ζητάει ή ακόμα και αν δεν το έχει να μπορέσει γρήγορα και άμεσα να μεταβεί σε άλλο E-shop για να το προμηθευτεί από εκεί.

Επίσης, σημαντικό είναι πως με αυτό τον τρόπο αυτό κάποιος πελάτης μπορεί πλέον να προμηθευτεί προϊόντα από άλλες χώρες. Είναι μια απίστευτη δυνατότητα γιατί καμία χώρα δεν μπορεί να παρέχει τα πάντα στις σύγχρονες ανάγκες του εκάστοτε πελάτη. Έτσι, μέσω του Internet, μπορεί να συνδεθεί και με άλλες χώρες για να βρει αυτό που ζητάει ή παρόλο που μπορεί να το βρει και στην Ελλάδα, ίσως να το βρει σε κάποια καλύτερη τιμή!



Εικόνα 2.3.2.1: Κατάστημα E-Shop

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΚΕΦΑΛΑΙΟ 3

ΕΞΥΠΗΡΕΤΗΣΗ ΠΕΛΑΤΩΝ

Σε αυτό το κεφάλαιο αναλύεται η εξυπηρέτηση πελατών και συγκεκριμένα τι σημαίνει αυτό και με ποιους τρόπους το βλέπουμε στην σύγχρονη εποχή μας.

3.1 Ορισμός

Ως εξυπηρέτηση πελατών ορίζουμε την διαδικασία κατά την οποία ένας πελάτης δέχεται διευκρινήσεις ή κάποιου είδους βοήθεια ή παροχή υπηρεσιών. Όταν για παράδειγμα ο πελάτης αγοράζει ένα ελαττωματικό προϊόν τότε επικοινωνεί με το κατάστημα που το αγόρασε και ο υπεύθυνος του καταστήματος εξυπηρετεί τον πελάτη και του δίνει οδηγίες και πληροφορίες για την αντιμετώπιση του προβλήματος.

3.2 Μορφές εξυπηρέτησης

Με την εξέλιξη της τεχνολογίας και των σύγχρονων αναγκών η ανάγκη των πελατών να εξυπηρετηθούν για κάτι αυξάνεται. Η εξυπηρέτηση των πελατών έχει πάρει διάφορες μορφές και συνεχώς εξελίσσεται.

3.2.1 Διαπροσωπική

Η πρώτη διαχρονική μορφή εξυπηρέτησης που παραμένει ακόμα και σήμερα η σημαντικότερη και πιο συχνή είναι η διαπροσωπική. Ο πελάτης έχει την δυνατότητα να συνομιλήσει με τον υπάλληλο του καταστήματος που βρίσκεται πρόσωπο με πρόσωπο, να ανταλλάξουν απόψεις και η μορφή συνομιλίας να έχει άμεσο, ευθύ και ζωντανό χαρακτήρα.

Η διαπροσωπική επικοινωνία κάνει τους ανθρώπους πιο άνετους και δημιουργεί το αίσθημα ότι ο υπάλληλος με τον οποίο συνομιλεί ο πελάτης τον καταλαβαίνει απόλυτα και οτιδήποτε και να συμβεί ή θελήσει θα εξυπηρετηθεί άμεσα και γρήγορα.

Αυτή η μορφή υπήρχε ακόμα και όταν η τεχνολογία ήταν πρακτικά ανύπαρκτη και είναι κάτι που την έχει κάνει να εκτιμάτε ιδιαίτερα στην σημερινή εποχή τόσο από την σύγχρονη κοινωνία αλλά ακόμα περισσότερο από τις λιγότερο ανεπτυγμένες κοινωνίες όπου η τεχνολογία δεν έχει ακμάσει ακόμα.

3.2.2 Τηλέφωνο

Μια από τις πιο συνηθισμένες, στο σύγχρονο κόσμο, μορφές εξυπηρέτησης είναι η εξυπηρέτηση μέσω τηλεφώνου.

Ο χρήστης μπορεί κάνοντας απλά μια κλήση στο κατάστημα που τον ενδιαφέρει μπορεί άμεσα να ενημερωθεί για ότι τον ενδιαφέρει και να επιλύσει τα προβλήματα που αντιμετωπίζει σχετικά με κάποιο προϊόν ή υπηρεσία, είτε να κάνει κάποια παραγγελία ή απλά να ενημερωθεί.

Σε αυτή την μορφή ο πελάτης επικοινωνεί με κάποιον εκπρόσωπο άμεσα οπότε ο λόγος είναι πιο ευθύς και άμεσος. Μπορεί να υπάρξει αλληλεπίδραση και να προσφέρει αρκετές απαντήσεις σε διάφορα ερωτήματα που έχει ο πελάτης ή που προκύπτουν στην πορεία για την καλύτερη κατανόηση του.

Το μεγάλο πλεονέκτημα της μορφής αυτής είναι ότι ο πελάτης δεν χρειάζεται να παρευρεθεί στο χώρο για να επικοινωνήσει με κάποιο εκπρόσωπο και έτσι να εξοικονομήσει χρόνο.

3.2.3 E-mail

Η μορφή εξυπηρέτησης βρίσκει πολλές εφαρμογές στην σύγχρονη εποχή του internet και βοηθάει και αυτή στην διευκόλυνση των πελατών.

Ο πελάτης, έχοντας πάντα πρόσβαση σε κάποια διεπαφή (tablet, κινητό, υπολογιστή κτλ.) και πρόσβαση στο internet μπορεί να συντάξει ένα ηλεκτρονικό μήνυμα (E-mail) και να το αποστείλει στο κατάστημα όπου κάποιος εκπρόσωπος θα το διαβάσει και θα απαντήσει αναλόγως με διευκρινήσεις και οδηγίες.

Αυτή η μορφή χάνει εντελώς την αμεσότητα που έχουν οι προηγούμενες μορφές αλλά ο λόγος είναι γραπτός και επομένως τις περισσότερες φορές πιο επίσημος και πιο σωστά διατυπωμένος.

Παρόλα αυτά αρκετές φορές επειδή τα μηνύματα που δέχονται τα καταστήματα είναι πολύ μεγάλος υπάρχουν καθυστερήσεις ως προς την απάντηση και πολλές φορές υπάρχουν αντιδράσεις.

ΚΕΦΑΛΑΙΟ 4

ΠΡΑΓΜΑΤΙΚΟΣ ΧΡΟΝΟΣ

Σε αυτό το κεφάλαιο γίνεται μια συνοπτική περιγραφή του πραγματικού χρόνου και μερικές σύγχρονες εφαρμογές για την καλύτερη κατανόηση του.

4.1 Ορισμός

Ως πραγματικός χρόνος ορίζεται το γεγονός το οποίο συμβαίνει ζωντανά και δεν είναι προγεγραμμένο ή μαγνητοσκοπημένο. Είναι μια ενέργεια που την παρακολουθείς να εξελίσσεται ή να διαφοροποιείται την στιγμή που γίνεται.

4.2 Εφαρμογές

Για να καταλάβουμε καλύτερα αυτή την έννοια πρέπει να δώσουμε μερικά παραδείγματα που το παρατηρούμε.

Σύγχρονες εφαρμογές παρατηρούνται σε πολλούς τομείς. Κλασικό παράδειγμα είναι η παρατήρηση της τοποθεσίας των πλοίων και των αεροπλάνων σε ηλεκτρονικούς χάρτες. Με αυτό τον τρόπο μπορούν να καθοριστούν πορείες καθώς και να αποφευχθούν κίνδυνοι και να υπάρξει καλύτερη οργάνωση γενικότερα σε αυτά τα δύο σημαντικά μέσα που φέρουν τόσους ανθρώπινες ζωές.

Άλλο παράδειγμα που παρατηρούμε είναι ο δείκτης του χρηματιστηρίου. Η παρατήρηση του αλλάζει τα δεδομένα σε πραγματικό χρόνο και επιφέρει θετικές και αρνητικές αλλαγές κάθε στιγμή.



Εικόνα 4.2.1: Παρατήρηση χρηματιστηρίου σε πραγματικό χρόνο

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΚΕΦΑΛΑΙΟ 5

ARDUINO & WI-FI SHIELD

Σε αυτό το κεφάλαιο αναλύονται τα χαρακτηριστικά του μικροεπεξεργαστή που χρησιμοποιήθηκε για την υλοποίηση της κατασκευής καθώς και του Shield που ήταν απαραίτητο για να υπάρξει σύνδεση με το δίκτυο Wi-Fi.

5.1 Η λογική των μικροεπεξεργαστών Arduino

Οι μικροεπεξεργαστές είναι μικρά συστήματα που καλύπτουν ένα μεγάλο εύρος εφαρμογών σε προσωπικό αλλά και σε επαγγελματικό επίπεδο ανά τον κόσμο. Είναι ολοκληρωμένα συστήματα που περιλαμβάνουν τα βασικά στοιχεία των υπολογιστών όπως επεξεργαστική ισχύ και μνήμη RAM και με τον κατάλληλο προγραμματισμό και συνδεσμολογία μπορούν να υλοποιήσουν οποιαδήποτε τεχνολογική εφαρμογή.

Χωρίζονται σε κατηγορίες ανάλογα με τις δυνατότητες του καθενός και για να μπορούν να ανταπεξέλθουν ικανοποιητικά στις ανάγκες της σύγχρονης αγοράς. Οι τιμές ποικίλουν αναλόγως. Αυτό που τους καθιστά τόσο εύχρηστους είναι το γεγονός ότι δεν χρειάζεται ένα ολοκληρωμένο υπολογιστικό σύστημα για δημιουργία διαφόρων project αλλά εξοικονομούν και χώρο και χρήματα υλοποιώντας τις βασικές λειτουργίες ενός κανονικού υπολογιστικού συστήματος.

Συγκεκριμένα οι μικροεπεξεργαστές Arduino ακολουθούν μια πολύ απλή λογική με την οποία υλοποιούν εφαρμογές. Διαθέτουν πόρτες οι οποίες μπορούν να χρησιμοποιηθούν είτε ως έξοδοι είτε ως είσοδοι. Κάθε πόρτα για να λειτουργήσει είτε ως έξοδος ή είσοδος πρέπει να οριστεί αναλόγως μέσα στο πρόγραμμα που αποθηκεύεται στην μνήμη του επεξεργαστή και υλοποιεί την εφαρμογή. Οι πόρτες ακολουθούν την λογική του 0 και 1 από 0-5Volt αναλόγως. Χωρίζονται επίσης σε αναλογικές και ψηφιακές για την ευρύτερη κάλυψη εφαρμογών.

Χάρη τις πόρτες αυτές μπορούν να συνδεθούν πάνω διάφορα συστήματα όπως ολοκληρωμένα, αισθητήρια ή ακόμα και να δημιουργηθούν συμπλέγματα από μικροεπεξεργαστές για την υλοποίηση πολύπλοκων εφαρμογών.

Παρόλα αυτά επειδή είναι απλά στην δομή τους συστήματα πρέπει να δίνεται έμφαση στην τάση που δίνεται ως τροφοδοσία αλλά και στις εισόδους/εξόδους καθώς μια λάθος σύνδεση ή υπέρταση θα μπορούσε να κάψει το σύστημα πολύ πιο εύκολα από ότι ένας υπολογιστής.

5.2 Χαρακτηριστικά του μικροεπεξεργαστή Arduino Uno

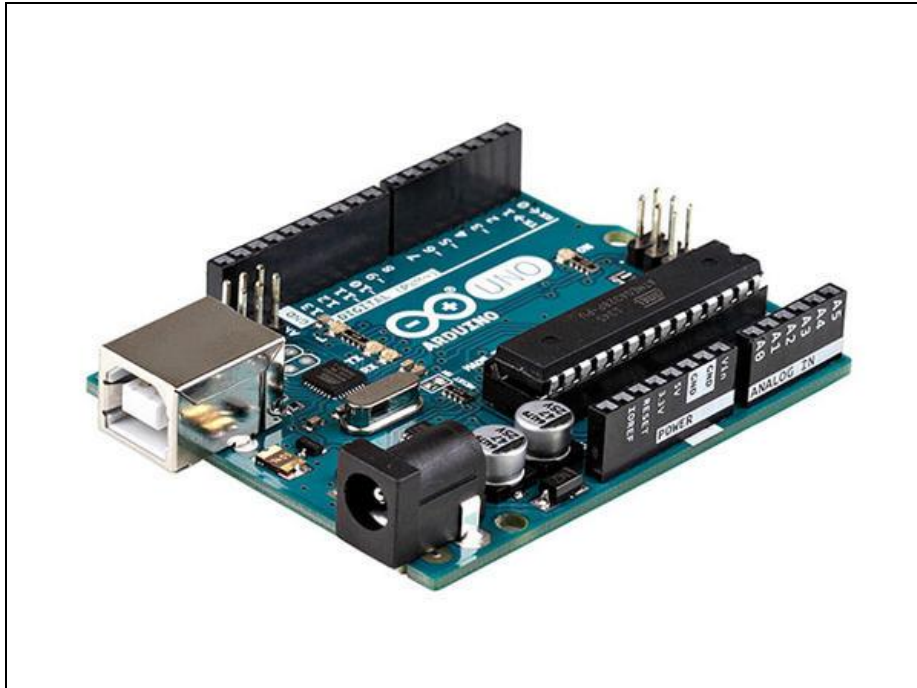
Ο μικροεπεξεργαστής Arduino UNO βασίζεται στον μικροελεγκτή ATmega328. Έχει 14 ψηφιακές I/O(από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν σαν

Τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

PWM έξοδοι), 6 αναλογικές εισόδους, έναν κρύσταλλο 16 MHz crystal, μια θύρα USB , μια θύρα τροφοδοσίας, μια κεφαλίδα ICSP και ένα κουμπί για reset. [3]

Χαρακτηριστικά:

- Μικροελεγκτής ATmega328
- Λειτουργία στα 5V
- Προτεινόμενη τάση εισόδου 7-12V
- Όριο τάσης εισόδου 6-20V
- Ψηφιακές εισοδοί/έξοδοι 14
- PWM I/O Pins 6
- Αναλογικοί εισοδοί Pins 6
- DC ρεύμα για κάθε I/O Pin 40 mA
- DC ρεύμα για 3.3V 50 mA
- Flash Memory 32 KB
- Flash Memory για Bootloader 0.5 KB
- SRAM 2 KB
- EEPROM 1 KB
- Ταχύτητα ρολογιού 16 MHz



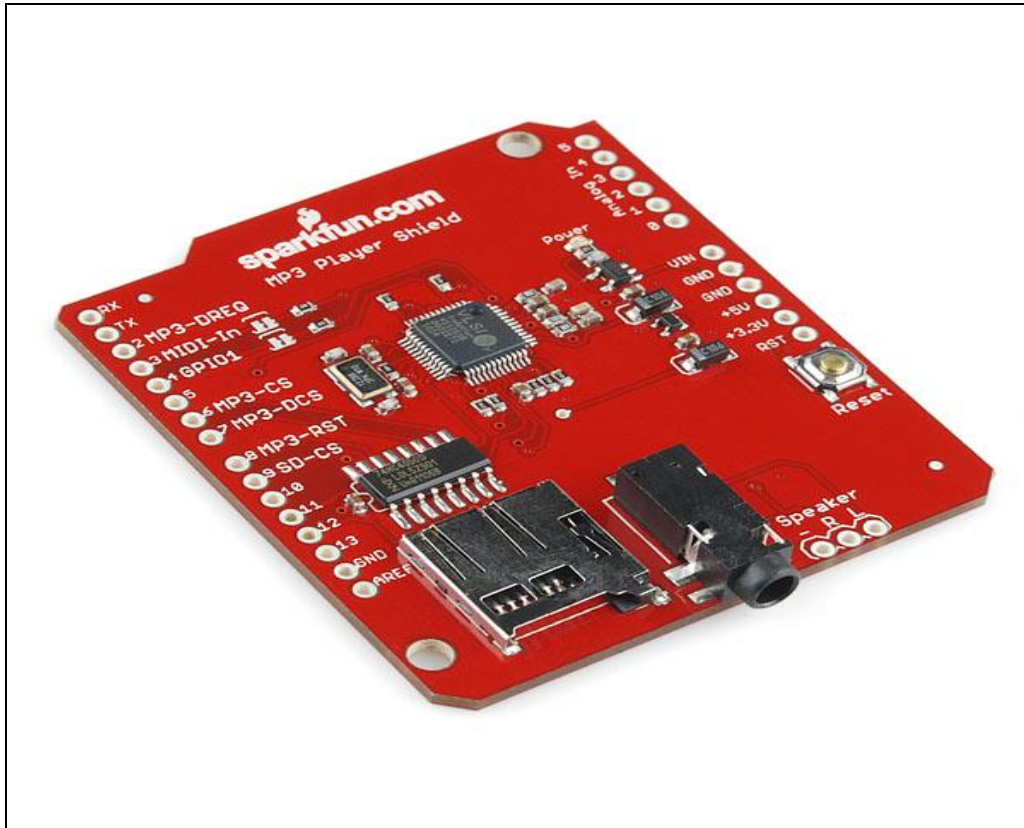
Εικόνα 5.2.1: Arduino Uno

5.3 Η λογική των Shield

Στο κεφάλαιο 5.1 αναλύθηκε η λογική των μικροεπεξεργαστών και ποιες οι λειτουργίες που υποστηρίζουν για την ανάπτυξη διαφόρων project. Παρ' ότι παρέχουν μια πληθώρα δυνατοτήτων, αρκετές φορές η ανάγκη δημιουργίας κάποιων ιδιαίτερων project απαιτεί συγκεκριμένες λειτουργίες οι οποίες αρκετές φορές δεν υποστηρίζονται από τους μικροεπεξεργαστές. Το κενό λειτουργιών αυτό στους μικροεπεξεργαστές Arduino, έρχονται να καλύψουν τα διάφορα shield.

Τα shield ουσιαστικά είναι πλακέτες οι οποίες συνδέονται με τους μικροεπεξεργαστές και αυξάνουν τις λειτουργίες που παρέχουν.

Υπάρχουν διάφορες εταιρείες κατασκευής Shield όπως η Sparkfun και η Adafruit και υπάρχει μια τεράστια ποικιλία μέσα από την οποία ο χρήστης μπορεί να βρει αυτό που του ταιριάζει ανάλογα το project που θέλει να υλοποιήσει. Ενδεικτικά μερικά από τα shield που υπάρχουν είναι Wi-fi shield, Motor Shield, MP3 Player Shield, GPS Shield, Weather Shield κ.α.



Εικόνα 5.3.1: Sparkfun MP3 Player Shield

5.4 Adafruit CC 3000 Wi-Fi Shield

Για την εφαρμογή που αναπτύχθηκε το σημαντικότερο και αδιάσπαστο κομμάτι της δεν ήταν άλλο από το Wi-Fi Shield CC3000 από την Sparkfun.

Το Wi-Fi shield CC3000 έχει απλή εγκατάσταση και είναι εύκολο στην χρήση του ακόμα και από νέους στον χώρο των μικροεπεξεργαστών. Χρησιμοποιεί Serial Peripheral Interface(SPI) αντί για Universal Asynchronous Receiver/Transmitter(UART) που χρησιμοποιούν τα περισσότερα Wi-Fi Shield που κυκλοφορούν και είναι ιδιαίτερα σημαντικό καθώς παρέχει την απαραίτητη ταχύτητα για μεταφορά δεδομένων όσο αργά ή γρήγορα θέλει ο χρήστης.[4]

Το συγκεκριμένο module βρίσκεται στην αγορά σε 2 εκδόσεις. Η μια είναι το breakout board δηλαδή ξεχωριστό board το οποίο απαιτεί συνδεσμολογία καλωδίων με τον μικροεπεξεργαστή, ενώ το δεύτερο και το οποίο χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι το Shield το οποίο συνδέεται απευθείας πάνω στον μικροεπεξεργαστή Arduino και μπορεί να χρησιμοποιηθεί άμεσα με τον κατάλληλο προγραμματισμό.



Εικόνα 5.4.1: Adafruit CC3000 Wi-Fi Shield & Breakout Board

5.5 Προγραμματισμός για το Adafruit CC3000 Wi-Fi Shield

Για την εκκίνηση και την χρήση του Shield χρειάζονται μερικές απλές εισαγωγές στο πρόγραμμα και έπειτα η γλώσσα C με την οποία προγραμματίζουμε τον μικροεπεξεργαστή μας παρέχει τις απαραίτητες εντολές σε πολλή απλή μορφή.

Συγκεκριμένα για το Shield οι εντολές που είναι απαραίτητες ανεξαρτήτως εφαρμογής είναι :

1. Η εισαγωγή της βιβλιοθήκης του module.
`#include <Adafruit_CC3000.h>`
`#include <Adafruit_CC3000_Server.h>`
2. Ορισμοί των pins μέσω των οποίων επικοινωνεί πρακτικά ο μικροεπεξεργαστής με το Wi-Fi Shield.
`#define ADAFRUIT_CC3000_IRQ 3`
`#define ADAFRUIT_CC3000_VBAT 5`
`#define ADAFRUIT_CC3000_CS 10`

3. Δήλωση των συγκεκριμένων pins για το module CC3000.
**Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS,
ADAFRUIT_CC3000_IRQ, ADAFRUIT_CC3000_VBAT,
SPI_CLOCK_DIVIDER);**

4. Ορισμός του ονόματος του δικτύου στο οποίο θα συνδέεται το Shield καθώς και του κωδικού αλλά και τον τύπο κωδικοποίησης που εφαρμόζεται.

```
#define WLAN_SSID "xxxxNETxxxx"
```

```
#define WLAN_PASS "0123456"
```

```
#define WLAN_SECURITY WLAN_SEC_WPA2
```

5. Ορισμός της πόρτας με την οποία θα επικοινωνεί το Shield στο τοπικό δίκτυο και δήλωση της πόρτας αυτής στο module.

```
#define LISTEN_PORT 80
```

```
Adafruit_CC3000_Server httpServer(LISTEN_PORT);
```

6. Τέλος κάνουμε εκκίνηση του server και έπειτα το πρόγραμμα περιμένει κάποιον χρήστη για να συνδεθεί στην πόρτα με την οποία επικοινωνεί.

```
httpServer.begin();
```

```
Adafruit_CC3000_ClientRef client = httpServer.available();
```

Από εκεί, ο τρόπος με τον οποίο κάνουμε οποιαδήποτε λειτουργία που αφορά τον χρήστη που συνδέθηκε στην πόρτα επικοινωνίας όπως για παράδειγμα να του προβάσουμε ένα μήνυμα ξεκινάει με client. και ακολουθούν οι τυπικές εντολές της γλώσσας C. Πχ. client.println ("Hello World");

ΚΕΦΑΛΑΙΟ 6

ΚΑΤΑΣΤΗΜΑ ΧΩΡΙΣ ΥΠΑΛΛΗΛΟΥΣ

Σε αυτό το κεφάλαιο αναλύεται η λογική και ο τρόπος με τον οποίο μπορεί να υλοποιηθεί και να λειτουργήσει ένα κατάστημα με την μέθοδο της τηλεξυπηρέτησης πελατών σε φυσικό κατάστημα.

6.1 Μορφή και διάταξη

Παρόλο που η μορφή του project που περιγράφεται είναι πολύ απλή πρέπει να γίνει μια γενική περιγραφή του για να γίνει απολύτως κατανοητό από όλους.

Το ζητούμενο λοιπόν, είναι ο πελάτης με το που εισέρχεται στο κατάστημα να έρχεται σε επαφή με ένα σύστημα “καλωσορίσματος” στο οποίο ο πελάτης θα έχει διαπροσωπική επαφή μέσω συστήματος video κλήσης με έναν υπάλληλο ο οποίος θα ρωτάει τον πελάτη με ποιον τρόπο θα πλοηγηθεί στο κατάστημα και αν ο τρόπος που επέλεξε θα μπορούσε να είναι επιβλαβής είτε για τον πελάτη είτε για το κατάστημα. Οι τρόποι θα είναι δύο. Ο πελάτης θα έχει την δυνατότητα να πλοηγηθεί στο κατάστημα είτε παραλαμβάνοντας ένα ειδικά διαμορφωμένο ηλεκτρικό scooter τύπου **Segway** όπως αυτό φαίνεται στην φωτογραφία από κάτω είτε χωρίς αυτό, είτε λόγω αδυναμίας χρήσης είτε λόγω προσωπικών προτιμήσεων, αλλά θα χρειαστεί να φορέσει στο χέρι του ένα ειδικό περιβραχιόνιο του οποίου η χρήση θα αναλυθεί μετέπειτα.



Εικόνα 6.1.1: Ηλεκτρικό Scooter Segway

Το scooter που θα παραλαμβάνει ο πελάτης θα παρέχει μια φιλική και απλή προς τον κοινό χρήστη διεπαφή (interface) η οποία θα του παρέχει αρκετές δυνατότητες με την σημαντικότερη από όλες, να μπορεί ο πελάτης να έχει άμεση

επαφή μέσω συστήματος video κλήσης σε πραγματικό χρόνο με έναν υπάλληλο του καταστήματος, εάν αυτός το επιλέξει. Ο πελάτης θα μπορεί να πλοηγηθεί στο κατάστημα με το scooter και να αγοράσει ότι επιθυμεί μεταφέροντας το στο ταμείο μέσω ενός ειδικού καλαθιού που θα είναι προσαρμοσμένο στο scooter είτε μέσω καλαθιού που θα παρέχεται στον χρήστη με την είσοδο του στο κατάστημα και εφόσον αυτός επιλέξει να μην χρησιμοποιήσει το scooter.

Με το τέλος των αγορών του ο πελάτης θα πληκτρολογεί σε μια ξεχωριστή διεπαφή τα στοιχεία της κάρτας του και θα περνάει από ένα σύστημα το οποίο θα ανιχνεύει αυτόματα τα προϊόντα και θα αφαιρεί τον λογαριασμό από τον προσωπικό λογαριασμό του πελάτη. Ένα τέτοιο σύστημα παρουσιάστηκε πρόσφατα από την Amazon με όνομα “Just walk out technology”(2). Έπειτα θα περνάει από τον αρχικό χώρο του μαγαζιού όπου θα αφήνεται το scooter ή το περιβραχιόνιο και θα μπορεί να αφήσει τον χώρο του μαγαζιού.

6.2 Απαραίτητες προϋποθέσεις

Για να μπορέσει αυτό το σύστημα να λειτουργήσει σωστά θα πρέπει να πληρούνται αρκετά κριτήρια τόσο για την ασφάλεια των πελατών όσο και για την ασφάλεια των ίδιων των μηχανημάτων του καταστήματος.

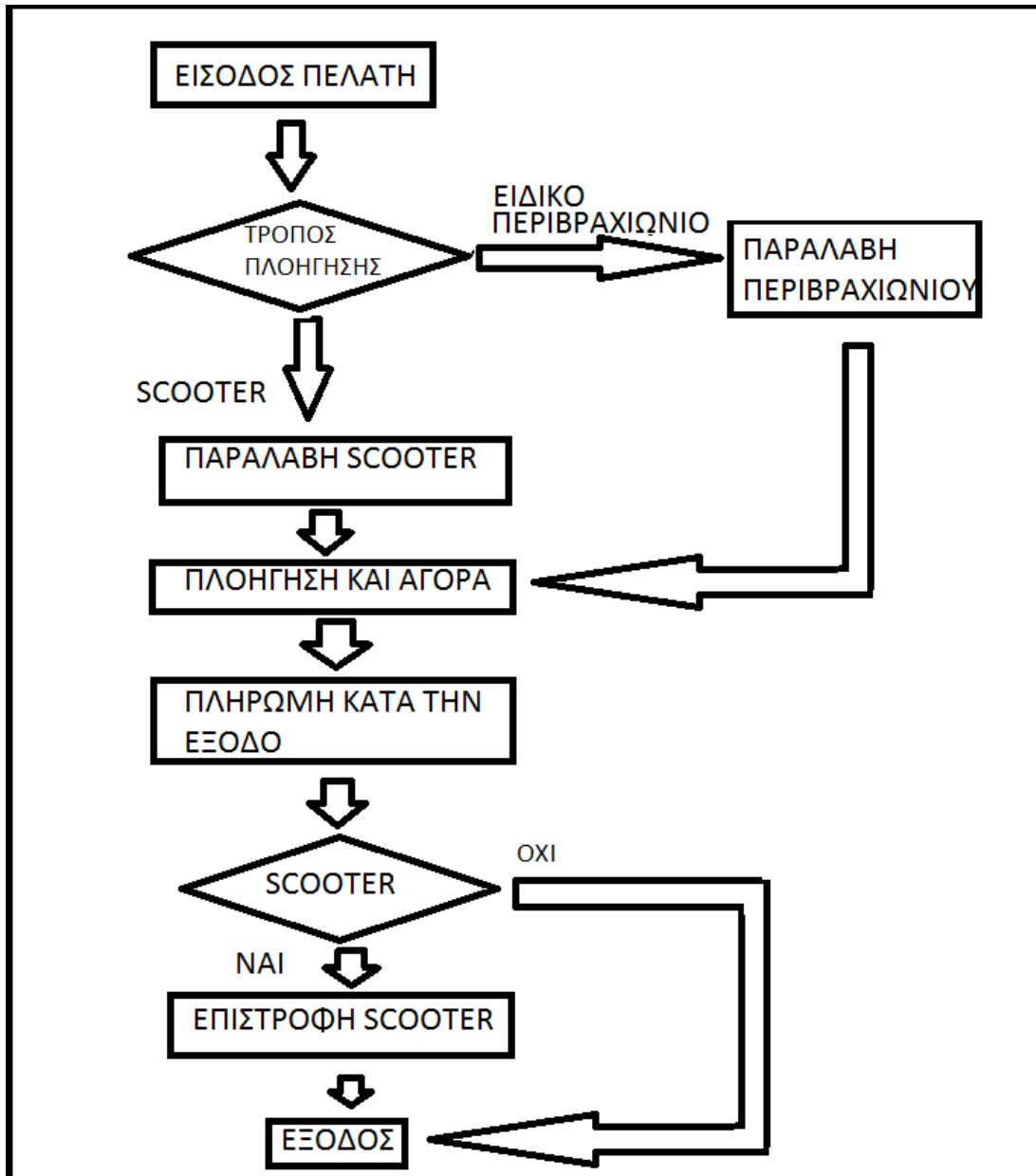
Όσων αφορά το κατάστημα:

- Ο χώρος θα πρέπει να είναι ευρύς και ίσως με περισσότερους από έναν ορόφους για την καλύτερη και ασφαλέστερη διέλευση των scooter στο κατάστημα.
- Τα scooter και τα περιβραχιόνια θα πρέπει να είναι εγκεκριμένα από όλους τους αρμόδιους φορείς για την ασφάλεια και την λειτουργικότητα τους τόσο προς το περιβάλλον όσο και προς τους πελάτες.
- Τα προϊόντα θα πρέπει να στεγάζονται σε λογικά ύψη ώστε ο πελάτης να μπορεί να τα προμηθευτεί χωρίς να χρειάζεται να κατέβει από το scooter.
- Η διεπαφή που θα παρέχει το scooter στον πελάτη θα πρέπει να είναι ιδιαίτερα φιλική χωρίς να χρειάζεται ή οποιαδήποτε κατατόπιση ως προς τις επιλογές της.

Όσων αφορά του πελάτες:

- Θα πρέπει να υπάρχει μέγιστο όριο βάρους το οποίο αν ξεπερνιέται να μην δίνεται στο πελάτη η δυνατότητα χρήσης του scooter προς αποφυγή ζημιάς του εξοπλισμού.

- Να μην δίνεται στον πελάτη η δυνατότητα χρήσης του scooter αν πάσχει από εμφανή προβλήματα υγείας ή εάν η ηλικία του καθιστά επίφοβη την χρήση του οχήματος.
- Οποιαδήποτε ογκώδη αποσκευή έχει ο πελάτης μαζί του κατά την είσοδο του στο κατάστημα να αφήνεται σε ειδικά διαμορφωμένο χώρο.



Σχήμα 6.2.1: Διάγραμμα ροής του πελάτη στο κατάστημα

6.3 Το ειδικό περιβραχιόνιό

Όπως αναφέραμε ο πελάτης εκτός από το scooter τύπου Segway θα μπορεί να πλοηγηθεί στον χώρο χωρίς αυτό φορώντας ένα ειδικό περιβραχιόνιό. Ο λόγος ύπαρξης του είναι για την διαφύλαξη τόσο της δικιάς του αλλά και τον υπόλοιπων

πελατών που παρευρίσκονται στο χώρο. Το περιβραχιόνιό αυτό θα λειτουργεί σαν πομπός ώστε να μπορέσουν να αποφευχθούν τυχών ατυχήματα μεταξύ των scooter και των πεζών πελατών καθώς θα αφήνουν το στίγμα του σε πραγματικό χρόνο σε έναν ενιαίο ηλεκτρονικό χάρτη που θα περιέχει όλους του πελάτες πεζούς και μη.

6.4 Το ηλεκτρικό scooter

Όπως αναφέρθηκε και παραπάνω για την πλοήγηση στο κατάστημα ο πελάτης θα μπορεί να χρησιμοποιεί ένα ειδικό ηλεκτρικό scooter τύπου Segway.

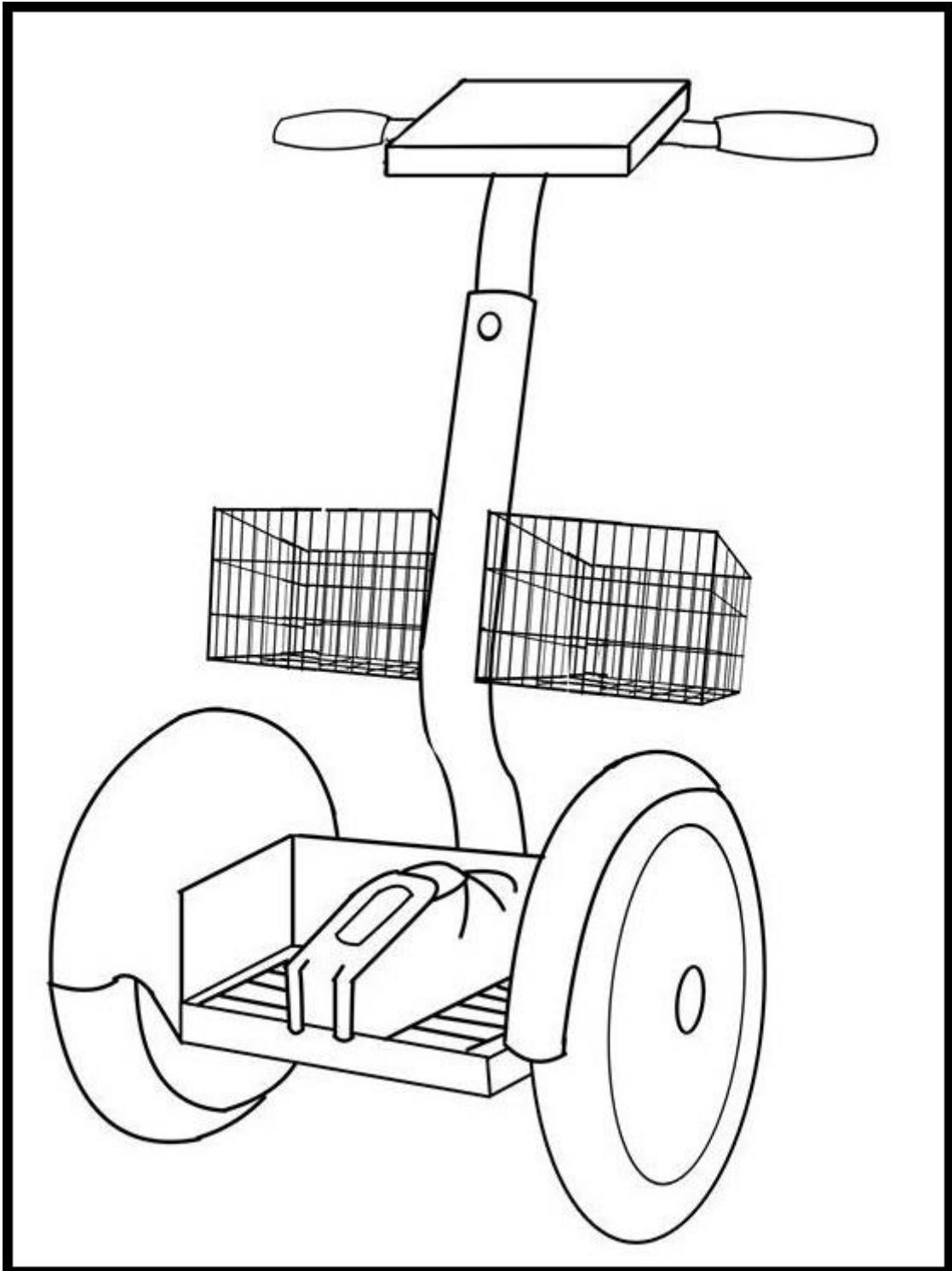
Διαλέγουμε να χρησιμοποιήσουμε αυτό το είδος scooter γιατί:

- Είναι αρκετά φιλικό προς το περιβάλλον.
- Οι διαστάσεις του είναι ιδανικές για συνύπαρξη με περισσότερα στον ίδιο χώρο.
- Είναι αισθητικά εύμορφο για την γενικότερη κομψότητα που πρέπει να έχει ένα κατάστημα.
- Συνδυάζει την πλοήγηση στον χώρο και την διεπαφή που περιγράφουμε μειώνοντας την πολυπλοκότητα και την επικινδυνότητα.

Πάνω από τις ρόδες θα υπάρχουν δύο καλάθια αντίστοιχα τα οποία αφενός θα στηρίζονται πάνω από τις ρόδες αλλά και από το κεντρικό σύστημα για μεγαλύτερη ασφάλεια και ισορροπία. Οι πελάτες θα μπορούν με αυτά να μεταφέρουν ποσότητες προϊόντων αλλά δεδομένου του γενικότερου συστήματος η χωρητικότητα θα μπορούσε για κάποιους πελάτες να θεωρηθεί μη επαρκή.

Το scooter για να μπορέσει να λειτουργήσει θα πρέπει να υπάρχει απαραίτητα σύνδεση στο Internet μέσω εσωτερικού συστήματος αυτόματης ασύρματης δικτύωσης(Wi-Fi). Δηλαδή το scooter αυτόματα με το που μπαίνει σε κατάσταση λειτουργίας θα συνδέεται στο δίκτυο του καταστήματος. Είναι ζωτικής σημασίας γιατί παρόλο που κάποιος πελάτης θα μπορούσε να το χρησιμοποιήσει για την μεταφορά του, οι περισσότερες λειτουργίες του και η πολυτέλεια που παρέχει θα ήταν μη λειτουργική.

Για την ασφάλεια του ως προς τους υπόλοιπους πελάτες, το scooter, θα περιλαμβάνει ένα ανεπτυγμένο σύστημα αισθητήρων το οποίο θα ανιχνεύει τυχών εμπόδια είτε αυτό εννοεί άλλους πελάτες είτε οποιοδήποτε άλλο εμπόδιο.



Σχήμα 6.4.1: Πρωτότυπο ηλεκτρικού scooter με ενσωματωμένη διεπαφή και καλάθια πάνω από τις ρόδες

Το scooter θα μπορεί να μεταφέρει τους πελάτες με άνεση αλλά πρέπει να αναφερθεί πως το συνολικό βάρος πάνω σε αυτό δεν θα πρέπει να ξεπερνάει τα 110 κιλά συνυπολογίζοντας τον πελάτη αλλά και τα προϊόντα που μεταφέρει. Για τον λόγο αυτό στην διεπαφή ο πελάτης θα ενημερώνεται αν τυχόν ξεπεράσει τα

100 κιλά και θα προτείνεται να ολοκληρώσει την αγορά του με τα προϊόντα που ήδη έχει ενώ αν πλησιάσει το ανώτερο όριο τότε για την καλύτερη ασφάλεια του πελάτη, του οχήματος και των υπόλοιπων πελατών θα οδηγείται αυτόματα στο ταμείο.

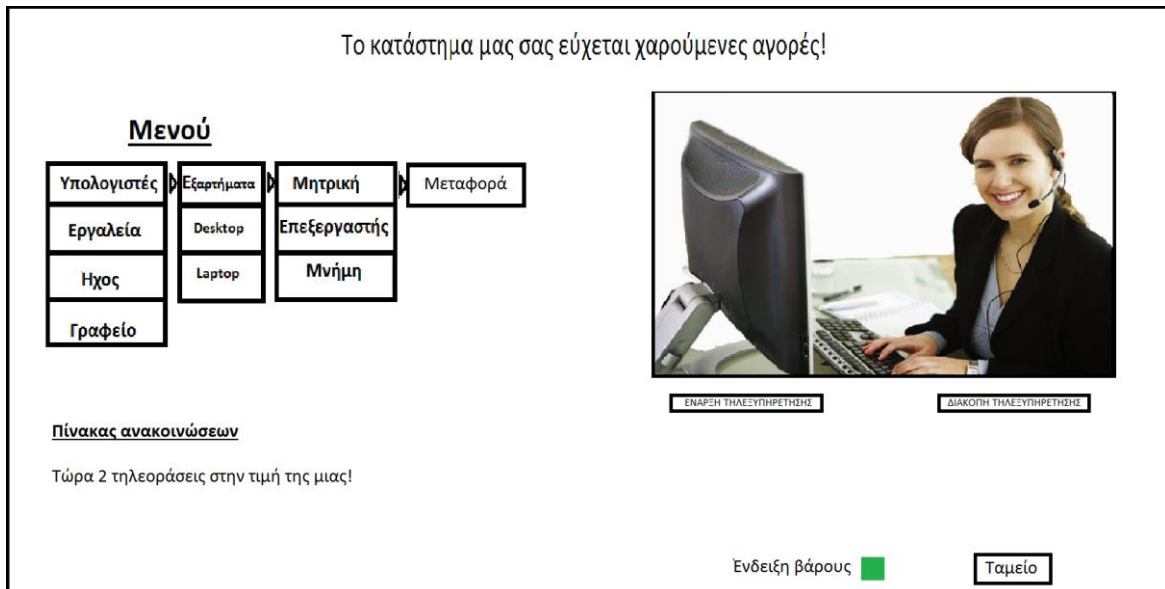
Η διεπαφή θα βρίσκεται ανάμεσα στα χερούλια τα όποια θα κρατάει ο πελάτης κατά την μεταφορά του και θα είναι απλή και κατανοητή από όλους.

Οι διαστάσεις της δεν θα ξεπερνάνε τις 15 ίντσες και προς ευκολία των πελατών θα είναι οθόνη αφής και θα έχει και σύστημα φωνητικών εντολών.

6.5 Η διεπαφή του οχήματος

Το πιο σημαντικό κομμάτι του συστήματος αυτού είναι η διεπαφή και οι δυνατότητες που θα παρέχει αυτή στον χρήστη.

Η διεπαφή όπως είπαμε θα είναι μια οθόνη επαφής 15 ιντσών και μέσω ενσωματωμένου μικροφώνου θα περιλαμβάνει και σύστημα φωνητικών εντολών.



Εικόνα 6.5.1: Παράδειγμα διεπαφής

Η σημαντικότερη λειτουργία της διεπαφής είναι η δυνατότητα που παρέχει στον πελάτη να συνομιλήσει με κάποιον υπάλληλο του καταστήματος ο οποίος θα έχει πλήρη ασύρματο έλεγχο του οχήματος και θα μπορεί να μεταφέρει τον πελάτη όπου εκείνος θέλει ή για να του παρουσιάσει τις επιλογές που έχει σύμφωνα με τις ανάγκες του. Με αυτό το τρόπο ο πελάτης θα έχει μια διαπροσωπική επαφή πλέον με το κατάστημα και θα μπορούν καλύτερα και άμεσα να εξυπηρετηθούν οι ανάγκες του νιώθοντας πιο οικεία με το περιβάλλον. Θα μπορεί να κάνει ερωτήσεις, σχόλια ή ακόμα και να υπάρχει ένα είδος αλληλεπίδρασης συμβούλων και από τον πελάτη προς τον υπάλληλο αλλά και αντίστροφα.

Τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

Η λοιπές λειτουργίες που θα παρέχει η διεπαφή θα είναι οι εξής:

1. Συνοπτικό μενού με τα προϊόντα και μεταφορά σε αυτά.
2. Σημαντικές ενημερώσεις ή προβλήματα τόσο στο κατάστημα όσο και για το ίδιο το όχημα.
3. Ένδειξη βάρους.
4. Επιλογή μεταφοράς σε τομέα προϊόντων ή στο ταμείο.

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΚΕΦΑΛΑΙΟ 7

ΕΦΑΡΜΟΓΗ

Σε αυτό το κεφάλαιο αναλύεται η υλοποίηση ενός Demo οχήματος που κινείται με εντολές του χρήστη μέσα από το internet. Πρακτικά είναι η διαδικασία που μπορεί να κάνει ο πωλητής όταν ο πελάτης θέλει να τον πάει σε κάποιο τομέα προϊόντων.

7.1 Γενική περιγραφή

Το όχημα που υλοποιήθηκε, για τις ανάγκες της πτυχιακής, αφορά τον έλεγχο του οχήματος μέσω ηλεκτρονικής διεπαφής στο internet. Η βάση του οχήματος είναι ο μικροεπεξεργαστής του, Arduino Uno, μέσω του οποίου προγραμματίστηκε για να εκτελεί τις λειτουργίες που είναι απαραίτητες ενώ πάνω στο όχημα υπάρχει ενσωματωμένο ένα κινητό τηλέφωνο όπου ο χρήστης βλέπει στο interface την πορεία του οχήματος.

Ο μικροεπεξεργαστής του οχήματος, λοιπόν, λειτουργεί σαν server ο οποίος παράγει ένα interface στο χρήστη, τοπικά, μέσω του οποίου ο χρήστης μπορεί, μέσω κουμπιών του πληκτρολογίου του, να ελέγξει την πορεία του οχήματος και να το κατευθύνει όπου χρειάζεται.

Το ενσωματωμένο στο όχημα κινητό, έχοντας εγκατεστημένη μια ειδική εφαρμογή, παρέχει στον χρήστη την εικόνα αλλά και τον ήχο από το κινητό στο interface όπου γίνεται και ο έλεγχος του οχήματος. Με αυτό τον τρόπο ο χρήστης μπορεί να βλέπει που ακριβώς πηγαίνει το όχημα ανά πάσα στιγμή και με μια μικρή καθυστέρηση να λαμβάνει τον ήχο από το μικρόφωνο του κινητού.

7.2 Σύνδεση με Wi-fi

Επικοινωνία σε επίπεδο δικτύου πρακτικά σημαίνει ανταλλαγή μηνυμάτων. Η σύνδεση του μικροεπεξεργαστή με το ενσωματωμένο Wi-Fi Shield με το τοπικό δίκτυο δεν διαφέρει.

Αρχικά, ο μικροεπεξεργαστής δημιουργεί έναν server που λειτουργεί σε τοπικό επίπεδο. Ο server αυτός διαχειρίζεται τις εντολές που του δίνει ο χρήστης μέσω μιας σελίδας που δημιουργεί στο τοπικό δίκτυο. Με την είσοδο του στην σελίδα ο χρήστης στέλνει ένα μήνυμα(http request) στον server δηλαδή στο Arduino ότι θέλει να επικοινωνήσει μαζί του. Το http request είναι της μορφής:

1. GET / HTTP/1.1
2. Host: 192.168.1.4
3. Connection: keep-alive
4. Upgrade-Insecure-Requests: 1

5. User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36
6. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
7. Accept-Encoding:gzip, deflate, sdch
8. Accept-Language: en-US,en;q=0.8

Με την σειρά του ο server στέλνει ένα μήνυμα στο χρήστη ως απάντηση και η επικοινωνία μετά μπορεί να ξεκινήσει.

1. GET / favicon.ico HTTP/1.1
2. Host: 192.168.1.4
3. Connection: keep-alive
4. User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36
5. Referer: http://192.168.1.4/
6. Accept-Encoding:gzip, deflate, sdch
7. Accept-Language: en-US,en;q=0.8

Με την ολοκλήρωση αυτής της διαδικασίας ο server είναι έτοιμος να δεχτεί τις εντολές του χρήστη μέσω της σελίδας που δημιουργεί και να κινήσει το όχημα αναλόγως. Ο χρήστης με το πάτημα κάποιου κουμπιού W,A,S,D για μπροστά, αριστερά, πίσω, δεξιά αναλόγως κινεί το όχημα προς την ανάλογη κατεύθυνση. Το όχημα θα κινείται προς την ανάλογη κατεύθυνση μέχρις ότου ο χρήστης πατήσει το κουμπί E για να σταματήσει. Τα αιτήματα κίνησης και σταματήματος που στέλνονται στον server έχουν την εξής μορφή και είναι τύπου GET.

1. GET /Front&nocache582619.528372357 HTTP/1.1
2. Host: 192.168.1.4
3. Connection: keep-alive
4. User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36
5. Accept: */*
6. Referer: http://192.168.1.4/
7. Accept-Encoding:gzip, deflate, sdch

8. Accept-Language: en-US,en;q=0.8

7.3 Εξαρτήματα και κατασκευή

Η λίστα με τα εξαρτήματα που χρειάστηκαν για την υλοποίηση της κατασκευής είναι η εξής:

1. Arduino Uno <https://www.arduino.cc/en/Main/ArduinoBoardUno>
2. Adafruit Wi-Fi shield CC3000 <https://www.adafruit.com/product/1491>
3. Battery 12V 3.3Ah http://www.vision-batt.com/site/product_files/cp1232.pdf
4. Robot Rover Chassis <http://grobotronics.com/mini-robot-rover-chassis-kit-2wd-with-dc-motors.html>
5. Motor Driver L293D <http://www.ti.com/product/L293D>
6. Voltage Regulators L7805 & L7806
7. 2 Πυκνωτές 100μF & 2 10μF
8. Διάφορα καλώδια, πολύμετρο, κολλητήρι, απογυμνωτής, τσιμπίδα κα. εργαλεία και γενικής φύσεως απαραίτητα.

Αρχικά, πρέπει να συνδεθεί ο μικροεπεξεργαστής Arduino Uno ώστε να μπορέσει να γίνει δυνατή η σύνδεση με το τοπικό δίκτυο Wi-Fi. Η σύνδεση του shield με τον μικροεπεξεργαστή γίνεται μέσω της κεφαλίδας ICSP. Για την στήριξη του shield πάνω στον μικροεπεξεργαστή χρησιμοποιούνται pins που εισέρχονται σε όλα τα I/O και προεξέχουν έτσι ώστε να μπορέσουν να συνδεθούν με τα I/O του shield. Για μεγαλύτερη σταθερότητα τα pins κολλιούνται με την χρήση κολλητηριού επάνω στο shield και έτσι το σύστημα με την κατάλληλη κωδικοποίηση είναι έτοιμο για λειτουργία.

Στην βάση της κατασκευής έχουμε το Robot Chassis πάνω στο οποίο υλοποιείται όλη η κατασκευή και μας παρέχει έτοιμες θέσεις για τους 2 κινητήρες που χρειάζονται και μια βοηθητική ρόδα.



Εικόνα 7.3.1: Robot Chassis

Ακριβώς πάνω στην βάση αυτή είναι η μπαταρία από την οποία τροφοδοτείται το κύκλωμα, δένοντάς την με tire-ups ενώ στο ύψος των 2 κυρίων τροχών χρησιμοποιώντας μερικά standoffs δίνεται το κατάλληλο ύψος για την χρήση 2 επιπλέον στρωμάτων, ένα για το ηλεκτρονικό κύκλωμα και ένα για τον μικροεπεξεργαστή Arduino Uno όπως φαίνεται στην εικόνα παρακάτω όπου φαίνεται ολοκληρωμένα η συνολική κατασκευή.



Εικόνα 7.3.2: Κατασκευή με χρήση Arduino & Wi-Fi Shield

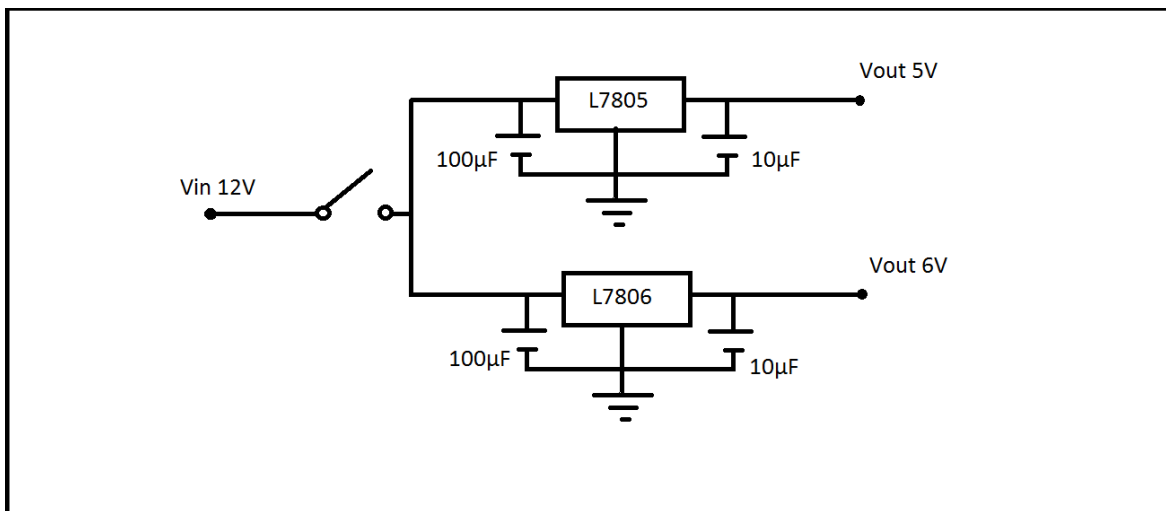
7.4 Ηλεκτρικό κύκλωμα – τροφοδοσία

Το ηλεκτρικό κύκλωμα αποτελείται από την μπαταρία 12V 3.3Ah από την οποία τροφοδοτείται τόσο ο μικροεπεξεργαστής Arduino Uno όσο και το ολοκληρωμένο L293D από το οποίο τροφοδοτούνται οι κινητήριιοι τροχοί του οχήματος.

Επειδή το ολοκληρωμένο L293D έχει μία είσοδο τάσεως (V_{in}) για την λειτουργία του και μία είσοδο τάσης για τους τροχούς οι οποίοι είναι της τάξεως των 6V, είναι απαραίτητοι 2 ρυθμιστές τάσεως L7805 & L7806 για έξοδο τάσης στα 5V & 6V αντίστοιχα. Για την σταθεροποίηση της τάσης που παράγουν συνδέουμε στην είσοδο και στην έξοδο των ρυθμιστών από έναν πυκνωτή 100 μ F & 10 μ F αντίστοιχα.

Το κύκλωμα υλοποιείται πάνω σε ένα διάτρητο prototype board και όλες οι συνδέσεις μεταξύ των εξαρτημάτων γίνεται με καλαΐ.

Οπότε το ηλεκτρικό κύκλωμα έχει την ακόλουθη μορφή:



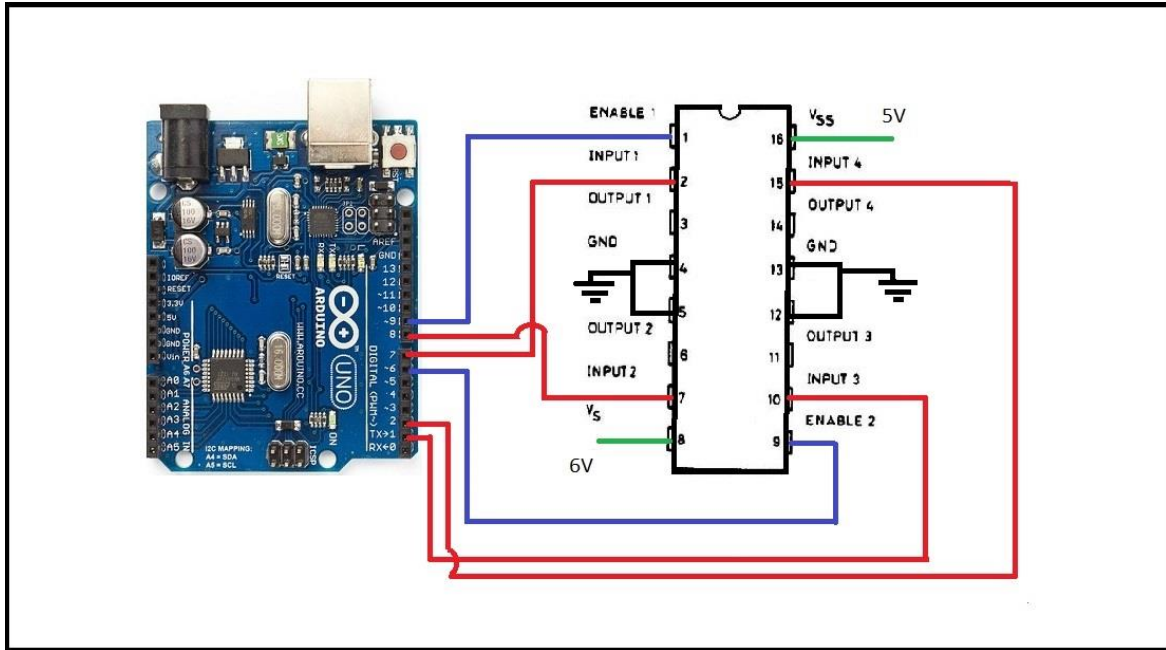
Σχήμα 7.4.1: Ηλεκτρικό κύκλωμα

7.5 Ηλεκτρονικό κύκλωμα

Το ηλεκτρονικό κύκλωμα αποτελείται από το ολοκληρωμένο L293D και την σύνδεση του με το Arduino Uno.

Το ολοκληρωμένο L293D αναλαμβάνει την κίνηση των τροχών ανάλογα με το σήμα που δέχεται στις εισόδους του από τον μικροεπεξεργαστή Arduino.

Στο παρακάτω σχήμα βλέπουμε τις επιμέρους εισόδους και εξόδους που απαρτίζουν το ολοκληρωμένο L293D και πως συνδέεται με τον μικροεπεξεργαστή αναλυτικά.



Σχήμα 7.5.1: Ηλεκτρονικό κύκλωμα

Τα Output 1 & 2 είναι οι έξοδοι που συνδέονται στον Α κινητήριο τροχό και τα Output 3 & 4 αντίστοιχα για τον Β.

Οι εισόδοι V_S και V_{SS} παίρνουν τάση από την έξοδο των σταθεροποιητών L7805 & L7806.

Το Wi-Fi shield δεσμεύει αρκετά από τα I/O του μικροεπεξεργαστή οπότε η επιλογή των I/O για το L293D ήταν περιορισμένη και αυστηρώς καθορισμένη.

7.6 Πρόγραμμα Arduino

Ο μικροεπεξεργαστής για να λειτουργήσει πρέπει να προγραμματιστεί ανάλογα με την λειτουργία που θέλουμε να πραγματοποιήσει και να συνδεθεί σωστά με τα υπόλοιπα μέρη του συστήματος.

Για να αναλυθεί το πρόγραμμα το δυνατόν καλύτερα θα το χωρίσουμε σε κατηγορίες γραμμών ως εξής:

- **Γραμμές 1-15:** Αρχικά το πρόγραμμα κάνει κάποιες γενικές δηλώσεις για να γίνει σωστά η μετάφραση του προγράμματος από τον μικροεπεξεργαστή. Επίσης δηλώνονται ποιες I/O (inputs/outputs) θα χρησιμοποιήσει το Wi-Fi shield καθώς και τα στοιχεία του router που θα συνδεθεί και την πόρτα (port) στην οποία θα συνδέεται.
- **Γραμμές 16-24:** Στο void setup γίνονται οι δηλώσεις για τις εξόδους και τις εισόδους του μικροεπεξεργαστή και έπειτα γίνεται εκκίνηση του server με την εντολή `http.Server.begin()`;
- **Γραμμές 25-249:** Στο void loop περιλαμβάνεται όλη η κυρίως λειτουργία του μικροεπεξεργαστή.

- **Γραμμές 25-39:** Το πρόγραμμα αφού ελέγχει αν υπάρχει κάποιος χρήστης συνδεδεμένος στο server λαμβάνει το http request του χρήστη και αφού ολοκληρωθεί τότε με την σειρά του ο server στέλνει ένα http request απάντησης και έπειτα η επικοινωνία έχει ολοκληρωθεί και μπορεί να συνεχιστεί.
- **Γραμμές 40-81:** Αφού ολοκληρωθεί η παραπάνω διαδικασία τότε ελέγχεται το request που στέλνει ο χρήστης και αν περιλαμβάνει μία από τις λέξεις Front, Back, Right, Left, Stop τότε δίνει σήμα High στα ανάλογα Outputs για να κινηθεί το όχημα αναλόγως. Εφόσον, το πρώτο request δεν περιλαμβάνει ποτέ κάτι από αυτά τότε το πρώτο που γίνεται είναι να παρουσιαστεί στο χρήστη το interface μέσα από το οποίο μπορεί να ελέγξει το όχημα.
- **Γραμμές 82-147:** Σε αυτές τις γραμμές υπάρχει ο κώδικας css για την μορφοποίηση του interface.
- **Γραμμές 148-198:** Έπειτα είναι ο κώδικας που λαμβάνει τα κουμπιά που πατήθηκαν στο πληκτρολόγιο, ο οποίος είναι γραμμένος σε γλώσσα JavaScript.
- **Γραμμές 199-212:** Σε αυτό το σημείο στέλνεται στο server η λέξη Front, Back, Left, Right, Stop αν πατήθηκε W,S,A,D,E αναλόγως για να εκτελέσει την ανάλογη πορεία,
- **Γραμμές 213-249:** Τέλος, υπάρχει ο κώδικας html που έχει το περιεχόμενο της σελίδας.

7.7 Interface

Το interface που προβάλλει ουσιαστικά ο μικροεπεξεργαστής είναι το μέσο με το οποίο μπορούμε να ελέγξουμε την κίνηση του και να δούμε τι βρίσκεται μπροστά του μέσω του ενσωματωμένου κινητού.



Εικόνα 7.7.1: Σελίδα ελέγχου της κατασκευής

Με την σύνδεση του ο χρήστης θα πρέπει να περιμένει 30 δευτερόλεπτα, που φαίνονται από έναν αντίστροφο μετρητή, πριν ξεκινήσει να στέλνει εντολές κίνησης στο όχημα. Αυτός ο χρόνος είναι ουσιαστικά για να γίνει σωστή εκκίνηση της επικοινωνίας μεταξύ του χρήστη και του server και για να φορτώσει πλήρως η σελίδα και να μην υπάρξουν κολλήματα.

Έπειτα ο χρήστης με την χρήση των κουμπιών A,W,S,D μπορεί να μετακινήσει το όχημα μέχρι να πατήσει το κουμπί για σταμάτημα (E). Είναι σημαντικό ο χρήστης να μην αφήνει στέλνει πολλά διαφορετικά σήματα ταυτόχρονα ή σε πολύ μικρά χρονικά διαστήματα γιατί κάποιες φορές η επικοινωνία μπορεί να είναι πιο αργή από ότι χρειάζεται το σύστημα και το σύστημα να υπολειπουργήσει.

7.8 Πρόγραμμα κινητού

Η σύνδεση του κινητού με το interface που παρέχει ο μικροεπεξεργαστής γίνεται μέσω μιας πολύ εύχρηστης εφαρμογής που ονομάζεται IP Webcam φτιαγμένη από τον Pavel Khlebovich. <https://play.google.com/store/apps/details?id=com.pas.webcam&hl=el>

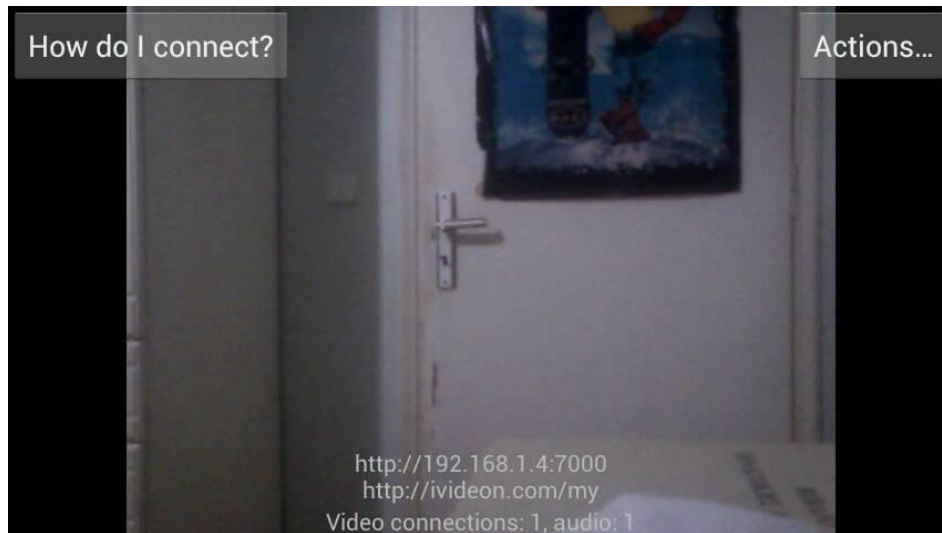
Η εφαρμογή αυτή με την εκκίνηση της στο κινητό τηλέφωνο δημιουργεί σε μια συγκεκριμένη πόρτα του τοπικού δικτύου ένα παράθυρο στο οποίο προβάλλεται η εικόνα από την κάμερα του κινητού τηλεφώνου. Επίσης, η εφαρμογή αυτή μας παρέχει εκτός από την εικόνα και τον ήχο του μικροφώνου του κινητού τηλεφώνου σε μια ξεχωριστή πόρτα από αυτή της εικόνας.

Η εικόνα παρόλο που προβάλλεται σε πραγματικό χρόνο, ο ήχος έχει μια καθυστέρηση της τάξεως των 3 δευτερολέπτων.

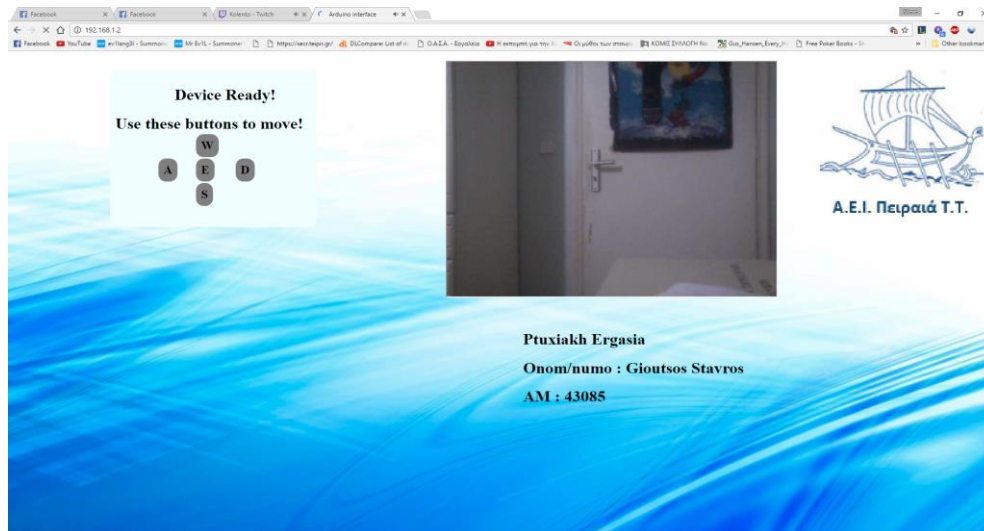
Στο interface μας εισάγουμε την εικόνα και των ήχο με την χρήση 2 iframe κλάσεων μέσα στις οποίες υπάρχει το link για το παράθυρο του ήχου και της εικόνας. Και έτσι με αυτό τον τρόπο έχουμε και την εικόνα και τον ήχο σε ένα παράθυρο μαζί με τον έλεγχο του οχήματος.

Παρακάτω βλέπουμε ένα στιγμιότυπο της οθόνης του κινητού και του interface ελέγχου της εφαρμογής σε πραγματικό χρόνο

Τηλεξυπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα



Εικόνα 7.8.1: Στιγμιότυπο οθόνης κινητού



Εικόνα 7.8.2: Στιγμιότυπο διεπαφής ελέγχου

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

ΚΕΦΑΛΑΙΟ 8

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

Σε αυτό το κεφάλαιο γίνεται μια σύνοψη της πτυχιακής εργασίας και της κατασκευής που υλοποιήθηκε. Εξάγονται κάποια συμπεράσματα για την δυνατότητα υλοποίησης σε πραγματικό περιβάλλον καθώς και προβλήματα που προέκυψαν κατά την υλοποίηση της κατασκευής.

8.1 Σύνοψη της πτυχιακής εργασίας

Η πτυχιακή εργασία αυτή περιείχε δύο κύρια μέρη. Ένα είναι το κομμάτι που αφορούσε την μελέτη του project “Τηλεξυπηρέτησης πελατών σε πραγματικό χρόνο για φυσικό κατάστημα” και ένα το κομμάτι που αφορούσε την κατασκευή ενός δείγματος που σε κανονικές διαστάσεις και καλύτερη υλοποίηση θα μπορούσε να είναι η βάση για το εγχειρήματος που περιγράφεται στο πρώτο μέρος.

Όσων αφορά το πρώτο μέρος, εξετάστηκαν και διευκρινίστηκαν αρκετές επιμέρους λεπτομέρειες και παράγοντες που επηρεάζουν αρνητικά ή θετικά το θέμα που περιγράφεται. Περιεγραφήκαν τόσο η μέθοδος με την οποία θα μπορούν οι πελάτες να εξυπηρετούνται οι πελάτες σε ένα φυσικό κατάστημα καθώς και ο τρόπος με τον οποίο θα γίνεται η περιήγηση στο κατάστημα, όσο και οι προδιαγραφές και τα επιμέρους συστήματα τα οποία πρέπει να προσαρμοστούν στον χώρο ο οποίος θα λειτουργήσει το εν λόγω σύστημα.

Όσων αφορά την κατασκευή, δόθηκαν οδηγίες για την υλοποίηση της και διευκρινήσεις για το πρόγραμμα που υλοποιεί την κίνηση του οχήματος και το interface μέσω του οποίου ο χρήστης μπορεί να ελέγξει το όχημα.

8.2 Προβλήματα κατά την υλοποίηση

Κατά την συγγραφή της μελέτης και της κατασκευής παρουσιάστηκαν διάφορα προβλήματα που έχριζαν άμεσης επίλυσης και καθυστέρησαν σημαντικά την ολοκλήρωση της.

Όσων αφορά την συγγραφή το κύριο πρόβλημα ήταν η έλλειψη οποιαδήποτε παρόμοιας τεχνολογίας που να προσομοιώνει το συγκεκριμένο θέμα που περιγράφεται σε κάποιο βαθμό έτσι ώστε να υπάρχει κάποια βάση πάνω στην οποία θα στηριζόταν η μελέτη μου. Το σύστημα που περιγράφεται έχει πολύ μικρές ομοιότητες σε κάποιους τομείς της με άλλες τεχνολογίες αλλά το κύριο μέρος που αφορά την τηλεξυπηρέτηση πελατών είναι προϊόν δικής μου έρευνας και σκεπτικού υλοποίησης με ότι αυτό συνεπάγεται. Προσπάθησα να μην ξεφύγω από την σφαίρα του εφικτού σύμφωνα πάντα με την σημερινή τεχνολογία και ταυτόχρονα να προσθέσω τις ιδέες μου για την μορφή που θα μπορούσαν να

πάρουν και να υλοποιήσουν το εν λόγω project.

Όσον αφορά την κατασκευή, ούτε εκεί έλλειψαν διάφορα προβλήματα που αντιμετωπίστηκαν άμεσα και επιτυχώς. Το κύριο πρόβλημα που αντιμετώπισα ήταν ότι δεν είχα αρκετή επαφή με τους μικροεπεξεργαστές έως καθόλου όταν ξεκίνησα να δουλεύω την κατασκευή και επομένως χρειάστηκαν αρκετές ώρες διαβάσματος και προσωπικής δουλειάς ώστε να φτάσω σε ένα σημείο που να μπορώ να υλοποιήσω την κατασκευή αυτή και να καταλαβαίνω απόλυτα πως λειτουργεί σε κάθε σημείο της. Χρειάστηκαν αρκετές ώρες δουλειάς για την κατασκευή της και μεγάλη καθυστέρηση της οφείλεται σε προσωπική αδυναμία κάλυψης της στο οικονομικό κομμάτι.

Επίσης, σημαντικό πρόβλημα που είχα να αντιμετωπίσω κατά τον προγραμματισμό της εφαρμογής, ήταν το μικρό μέγεθος μνήμης του μικροεπεξεργαστή το οποίο με περιόρισε στην βασικές λειτουργίες καθώς πρόσθετες λειτουργίες θα έκαναν τις ήδη υπάρχουσες ελαττωματικές ή την κατασκευή στο γενικότερο σύνολο της μη λειτουργική. Το πρόβλημα επιλύθηκε σε μεγάλο βαθμό με την χρήση της εικονικής μνήμης(flash) αντί της κανονικής και έτσι το σύστημα δούλεψε όπως ήταν προγραμματισμένο εξ αρχής.

8.3 Προοπτικές

Το θέμα που μελετάται σε αυτή την πτυχιακή αλλά και η κατασκευή που υλοποιήθηκε πέραν της ανάλυσης που έγινε, διαθέτουν φυσικά αρκετές προοπτικές εξέλιξης.

Θεωρώ πως το θέμα που ερευνάται είναι η κατεύθυνση προς την οποία οδηγείται η κοινωνία και η τεχνολογία και πως εν καιρό αυτή η τεχνολογία θα εδραιωθεί και θα λειτουργεί σε κάθε γωνιά του κόσμου. Με την ευρεία χρήση της τεχνολογίας που περιγράφεται θα παρουσιαστούν καινοτόμες ιδέες τόσο στο κομμάτι της μετακίνησης των πελατών στο χώρο αλλά και την επικοινωνία τους με υπάλληλους του καταστήματος για την καλύτερη διευκόλυνση των πελατών.

Μία από αυτές τις ιδέες θα ήταν η δημιουργία προφίλ πελατών έτσι ώστε μετέπειτα να ομαδοποιηθούν και να μπορούν να εξυπηρετούνται από συγκεκριμένους υπαλλήλους κατά ομάδες μειώνοντας έτσι το ανθρώπινο δυναμικό που χρειάζεται για να εξυπηρετηθεί ο μεγάλος όγκος πελατών.

Στο κομμάτι της κατασκευής, πέραν της υλοποίησης της σε πραγματικά μεγέθη για την κάλυψη των αναγκών του θέματος που ερευνήθηκε, υπάρχουν πολλές προοπτικές εξέλιξης όπως προσθήκη αισθητήρων για έλεγχο του οχήματος στο χώρο σε πραγματικό χρόνο.

ΠΑΡΑΡΤΗΜΑ Α΄

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της παρούσας εφαρμογής.

```
1. #include <Adafruit_CC3000.h>
2. #include <Adafruit_CC3000_Server.h>
3. #include <SPI.h>
4. #include "utility/debug.h"
5. #include "utility/socket.h"
6. #define ADAFRUIT_CC3000_IRQ 3
7. #define ADAFRUIT_CC3000_VBAT 5
8. #define ADAFRUIT_CC3000_CS 10
9. Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS,
    ADAFRUIT_CC3000_IRQ,
    ADAFRUIT_CC3000_VBAT,SPI_CLOCK_DIVIDER);
10.#define WLAN_SSID "xxxx"
11.#define WLAN_PASS "123456xxx"
12.#define WLAN_SECURITY WLAN_SEC_WPA2
13.#define LISTEN_PORT 80
14. Adafruit_CC3000_Server httpServer(LISTEN_PORT);
15.String HTTP_req;
16.void setup()
17.{
18.pinMode(1, OUTPUT);
19.pinMode(2, OUTPUT);
20.pinMode(6, OUTPUT);
21.pinMode(7, OUTPUT);
22.pinMode(8, OUTPUT);
23.pinMode(9, OUTPUT);
24.httpServer.begin();
25.}
26.void loop()
27.{
```

```
28. Adafruit_CC3000_ClientRef client = httpServer.available();
29. if (client) { // αν υπάρχει συνδεδεμένος χρήστης
30.   boolean currentLineIsBlank = true;
31.   while (client.connected()) {
32.     if (client.available()) { //αν τα δεδομένα του client είναι διαθέσιμα προς
        ανάγνωση
33.       char c = client.read(); // διαβάζει 1 byte (χαρακτήρα) από τον client
34.       HTTP_req += c; // Σώσιμο του HTTP request 1 χαρακτήρα την φορά
35.       if (c == '\n' && currentLineIsBlank) {
36.         client.println(F("HTTP/1.1 200 OK"));
37.         client.println(F("Content-Type: text/html"));
38.         client.println(F("Connection: keep-alive"));
39.         client.println();
40.         if (HTTP_req.indexOf("Front") > -1){
41.           digitalWrite(1, HIGH);
42.           digitalWrite(2, LOW);
43.           digitalWrite(7, HIGH);
44.           digitalWrite(8, LOW);
45.           analogWrite (6, 122);
46.           analogWrite (9, 122);
47.         } else if (HTTP_req.indexOf("Back") > -1) {
48.           digitalWrite(1, LOW);
49.           digitalWrite(2, HIGH);
50.           digitalWrite(7, LOW);
51.           digitalWrite(8, HIGH);
52.           analogWrite (6, 122);
53.           analogWrite (9, 122);
54.         } else if (HTTP_req.indexOf("Stop") > -1) {
55.           digitalWrite(1, LOW);
56.           digitalWrite(2, LOW);
57.           digitalWrite(7, LOW);
58.           digitalWrite(8, LOW);
59.           analogWrite (6, 0);
60.           analogWrite (9, 0);
```



```
61.} else if (HTTP_req.indexOf("Right") > -1) {
62.digitalWrite(1, HIGH);
63.digitalWrite(2, LOW);
64.digitalWrite(7, LOW);
65.digitalWrite(8, LOW);
66.analogWrite (6, 122);
67.analogWrite (9, 122);
68.} else if (HTTP_req.indexOf("Left") > -1) {
69.digitalWrite(1, LOW);
70.digitalWrite(2, LOW);
71.digitalWrite(7, HIGH);
72.digitalWrite(8, LOW);
73.analogWrite (6, 122);
74.analogWrite (9, 122);
75.} else {
76.digitalWrite(1, LOW);
77.digitalWrite(2, LOW);
78.digitalWrite(7, LOW);
79.digitalWrite(8, LOW);
80.digitalWrite(6, LOW);
81.digitalWrite(9, LOW);
82.client.println (F ("      <!DOCTYPE html>                                "));
83.client.println (F("      <html>                                "));
84.client.println (F("      <head>                                "));
85.client.println(F("      <style>                                "));
86.client.println(F("      body {                                "));
87.client.println(F("          background-image:          url('https://scontent.fath1-
      1.fna.fbcdn.net/v/t31.0-
      8/16112982_1539749336053705_677109467139448820_o.jpg?oh=6668903
      876ea29cc76e8e29ebbb52a57&oe=58D81810');                                "));
88.client.println(F("      }                                "));
89.client.println(F("      .buttons {                                "));
90.client.println(F("          margin-top          : 40px;                                "));
91.client.println(F("          margin-left          : 100px;                                "));
```

```

92. client.println(F("          height          : 400px;          "));
93. client.println(F("          width           : 400px;          "));
94. client.println(F("          border-radius    : 15px;          "));
95. client.println(F("          background-color  : azure;       "));
96. client.println(F("          padding          : 10px;        "));
97. client.println(F("          text-align       : center;      "));
98. client.println(F(" }                          "));
99. client.println(F("    iframe {                "));
100. client.println(F("        width             : 640px;       "));
101. client.println(F("        height            : 480px;       "));
102. client.println(F("        position          : absolute;     "));
103. client.println(F("        top               : 20px;        "));
104. client.println(F("        left              : 750px;       "));
105. client.println(F("    }                       "));
106. client.println(F("    #W {                    "));
107. client.println(F("        margin            : auto;         "));
108. client.println(F("        width             : 40px;        "));
109. client.println(F("        border-radius     : 10px;        "));
110. client.println(F("        background-color  : grey;        "));
111. client.println(F("        padding          : 10px;        "));
112. client.println(F("    }                       "));
113. client.println(F("    #A {                    "));
114. client.println(F("        display           : inline-block; "));
115. client.println(F("        width             : 40px;        "));
116. client.println(F("        border-radius     : 10px;        "));
117. client.println(F("        background-color  : grey;        "));
118. client.println(F("        padding          : 10px;        "));
119. client.println(F("    }                       "));
120. client.println(F("    #E {                    "));
121. client.println(F("        margin            : auto 20px;   "));
122. client.println(F("        display           : inline-block; "));
123. client.println(F("        width             : 40px;        "));
124. client.println(F("        border-radius     : 10px;        "));
125. client.println(F("        background-color  : grey;        "));

```

```

126. client.println(F("          padding          : 10px;          "));
127. client.println(F(" }          "));
128. client.println(F(" #D {          "));
129. client.println(F("          display          : inline-block;          "));
130. client.println(F("          width           : 40px;          "));
131. client.println(F("          border-radius    : 10px;          "));
132. client.println(F("          background-color  : grey;          "));
133. client.println(F("          padding          : 10px;          "));
134. client.println(F(" }          "));
135. client.println(F(" #S {          "));
136. client.println(F("          position         : absolute;          "));
137. client.println(F("          width           : 40px;          "));
138. client.println(F("          border-radius    : 10px;          "));
139. client.println(F("          background-color  : grey;          "));
140. client.println(F("          padding          : 10px;          "));
141. client.println(F(" }          "));
142. client.println(F(" #name{          "));
143. client.println(F("          position         : absolute;          "));
144. client.println(F("          top              : 550px;          "));
145. client.println(F("          left             : 850px;          "));
146. client.println(F(" }          "));
147. client.println(F(" </style>          "));
148. client.println(F(" <title>Arduino interface</title>          "));
149. client.println(F(" <script>          "));
150. client.println(F(" var mousedownID = -1;          "));
151. client.println(F(" var direction = 0;          "));
152. client.println(F(" var id = 0;          "));
153. client.println(F(" var test = 0;          "));
154. client.println(F(" var i = 30;          "));
155. client.println(F(" (function()          "));
156. client.println(F(" {          "));
157. client.println(F("          var timer = setInterval(function(){          "));
158. client.println(F("          document.getElementById('Timer').innerHTML =          "));
    i-=1;          "));

```

```
159. client.println(F("          if (i-1 == 0){          "));
160. client.println(F("          clearInterval(timer);          "));
161. client.println(F(" document.getElementById('State').innerHTML=
'Device Ready!';          "));
162. client.println(F(" document.getElementById('Timer').innerHTML=
null;"));
163. client.println(F("    }},1000);          "));
164. client.println(F("          window.addEventListener('keydown',
keypress_handler, false);          "));
165. client.println(F(" function keypress_handler(event)          "));
166. client.println(F(" {          "));
167. client.println(F("          if (i-1 == 0) {          "));
168. client.println(F("          if(event.keyCode == 87)          "));
169. client.println(F("          {          "));
170. client.println(F("              direction = 'Front';          "));
171. client.println(F("              id = 1;          "));
172. client.println(F("              request(direction,id);          "));
173. client.println(F("          }          "));
174. client.println(F("          if(event.keyCode == 83)          "));
175. client.println(F("          {          "));
176. client.println(F("              direction = 'Back';          "));
177. client.println(F("              id = 2;          "));
178. client.println(F("              request(direction,id);          "));
179. client.println(F("          }          "));
180. client.println(F("          if(event.keyCode == 65)          "));
181. client.println(F("          {          "));
182. client.println(F("              direction = 'Left';          "));
183. client.println(F("              id = 3;          "));
184. client.println(F("              request(direction,id);          "));
185. client.println(F("          }          "));
186. client.println(F("          if(event.keyCode == 69)          "));
187. client.println(F("          {          "));
188. client.println(F("              var direction = 'Stop';          "));
189. client.println(F("              id = 5;          "));
```

```

190. client.println(F("                request(direction,id);                "));
191. client.println(F("                }                "));
192. client.println(F("                if(event.keyCode == 68                "));
193. client.println(F("                {                "));
194. client.println(F("                var direction = 'Right';                "));
195. client.println(F("                id = 4;                "));
196. client.println(F("                request(direction,id);                "));
197. client.println(F("                }                "));
198. client.println(F("        }}                "));
199. client.println(F("                function request(direction,id                "));
200. client.println(F("                {                "));
201. client.println(F("                if(direction == 'Stop') {                "));
202. client.println(F("        document.getElementById('State').innerHTML=                "));
        'Stopped';                "));
203. client.println(F("                }else{                "));
204. client.println(F("        document.getElementById('State').innerHTML=                "));
        'Moving ' + direction;                "));
205. client.println(F("                }                "));
206. client.println(F("                if(id != test){                "));
207. client.println(F("                nocache = '&nocache' + Math.random() *                "));
        1000000;                "));
208. client.println(F("        var request = new XMLHttpRequest();                "));
209. client.println(F("        request.open('GET',direction + nocache, true);                "));
                "));
210. client.println(F("        request.send(null);                ")); [5]
211. client.println(F("        test = id;                }                "));
212. client.println(F("    }}();</script>                "));
213. client.println(F("    </head>                "));
214. client.println(F("                <body>                "));
215. client.println(F("        <iframe                src='http://192.168.1.4:7000/video'>                "));
        </iframe>                "));
216. client.println(F("        <iframe                hidden='hidden'                "));
        src='http://192.168.1.4:7000/audio.wav'> </iframe>                "));
217. client.println(F("                <div class='buttons'>                "));

```

```

218. client.println(F("      <h1 id='State'> Loading: </h1>          "));
219. client.println(F("      <h1 id='Timer'> 30          </h1>          "));
220. client.println(F("      <h1 id='Use'>          Use these buttons to move!
      </h1>          "));
221. client.println(F("      <h2 id='W'>W</h2>          "));
222. client.println(F("      <h2 id='A'>A</h2>          "));
223. client.println(F("      <h2 id='E'>E</h2>          "));
224. client.println(F("      <h2 id='D'>D</h2>          "));
225. client.println(F("      <h2 id='S'>S</h2>          "));
226. client.println(F("      </div>          "));
227. client.println(F("      <div id='name'>          "));
228. client.println(F("      <h1>Ptuxiakh Ergasia</h1>          "));
229. client.println(F("      <h1>Onom/numo : Gioutsos Stavros</h1>"));
230. client.println(F("      <h1>AM          : 43085</h1>          "));
231. client.println(F("      </div>          "));
232. client.println(F("      </body>          "));
233. client.println(F("      </html>          "));
234.          }
235.          HTTP_req = "";          // το request τελειώνει με κενό χαρακτήρα
236.          break;          }
237.          // κάθε γραμμή κειμένου που λαμβάνεται από τον χρήστη
          τελειώνει με \n
238.          if (c == '\n') {
239.              // ο τελευταίος χαρακτήρα της γραμμής κειμένου που λήφθηκε
240.              // ξεκίνημα νέας γραμμής με τον επόμενο χαρακτήρα
241.              currentLineIsBlank = true;
242.          }
243.          else if (c != '\r') {
244.              // ένας χαρακτήρας λήφθηκε από τον χρήστη
245.              currentLineIsBlank = false;
246.          }
247.      } }
248.      delay(1);          // δίνουμε χρόνο στον διακομιστή να λάβει τα δεδομένα
249.      client.stop(); } } // κλείσιμο της σύνδεσης

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] History of vending machines www.vendingmachine.umwblogs.org/antecedents
- [2] Tim Lambert, A brief History of Shops <http://www.localhistories.org/shopping.html>
- [3] Γενικές πληροφορίες και χαρακτηριστικά για το Arduino Uno <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- [4] Γενικές πληροφορίες και χαρακτηριστικά για το Adafruit CC 3000 Wi-Fi Shield <https://learn.adafruit.com/adafruit-cc3000-wifi/overview>
- [5] Χρήσιμα στοιχεία όσον αφορά την αποστολή http request κτλ <https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/web-server-read-switch-using-AJAX/>

Τηλευπηρέτηση πωλήσεων σε πραγματικό χρόνο για φυσικό κατάστημα

