



Ανάπτυξη Web 2.0 Mashup Διαδικτυακής Εφαρμογής Διαχείρισης Στόλου

Διπλωματική Εργασία για την ολοκλήρωση των σπουδών του Μεταπτυχιακού
Προγράμματος
"Προηγμένα Τηλεπικοινωνιακά Συστήματα και Δίκτυα"

Πανεπιστήμιο Πελοποννήσου

Συντάκτης: Χρηστάκος Γεώργιος

Υπεύθυνος Καθηγητής: Νίκος Τσελίκας

Τρίπολη 2011

Περιεχόμενα

Πρόλογος

Ευχαριστίες

Περίληψη

Κεφάλαιο 1: Εισαγωγή

1.1 Δομή Εργασίας

1.2 Στόχοι Εργασίας

1.3 Μεθοδολογία Εργασίας

Κεφάλαιο 2: Περιγραφή Αναγκών και Ανάλυση Εφαρμογής

2.1 Περιγραφή Αναγκών

2.2 Εργαλεία ανάπτυξης εφαρμογής και εξοπλισμός

2.3 Περιγραφή εφαρμογής

Κεφάλαιο 3: Περιγραφή της Βάσης Δεδομένων και Ανάλυση Κώδικα

3.1 Βάσεις Δεδομένων

3.2 Πελάτης

3.3 Διαχειριστής

3.4 Ταξί

3.5 Google Maps

Κεφάλαιο 4: Μελλοντικές Επεκτάσεις

4.1 Ιστορικό Κινήσεων

4.2 Ιστορικό Διαδρομών

4.3 Περιοχές εξαίρεσης - συναγεμίου

4.4 Περιορισμένη κίνηση των ταξί

4.5 Διευθύνσεις πελάτη

4.6 Κατάσταση κλήσεων πελάτη

4.7 Διάφορες εξειδικεύσεις

Κεφάλαιο 5: Βιβλιογραφία

Πρόλογος

Η πτυχιακή αυτή εργασία υλοποιήθηκε στο Πανεπιστήμιο Πελοποννήσου στα πλαίσια του Μεταπτυχιακού προγράμματος “Προηγμένα Τηλεπικοινωνιακά Συστήματα και Δίκτυα”.

Η εστίαση της έρευνας στα Γεωγραφικά Πληροφοριακά Συστήματα (GPS) έχει μεταφερθεί, κατά τα τελευταία χρόνια, από τα παραδοσιακά συστήματα διαχείρισης δεδομένων (μοντελοποίηση, δεικτοδότηση, ερωτήσεις) σε νέες και συναρπαστικές προκλήσεις που ανακύπτουν από την εξέλιξη νέων τεχνολογιών. Δύο από τα μεγαλύτερα πρόσφατα επιτεύγματα αυτών των τεχνολογιών, ο Παγκόσμιος Ιστός (www) και η ανάπτυξη συστημάτων εντοπισμού ακριβείας, έχουν μεγάλη επίδραση στα GPS.

Ο Ιστός διευκολύνει την ανταλλαγή και ενοποίηση των δεδομένων. Κινούμαστε με γοργούς ρυθμούς προς έναν κόσμο καταμεμημένων υπηρεσιών, κάθε μία εξειδικευμένη σε έναν συγκεκριμένο χειρισμό πληροφορίας, που συνεργάζονται, ανταλλάσσουν και ενοποιούν τα δεδομένα.

Τα συστήματα εντοπισμού συνιστούν μια άλλη «προκλητική» και ενδιαφέρουσα περιοχή. Το Global Positioning System (GPS) και το νέο ευρωπαϊκό πρόγραμμα δορυφορικών συστημάτων Galileo¹ (http://en.wikipedia.org/wiki/Galileo_%28satellite_navigation%29) που μπορούν να προσδιορίσουν τη θέση ενός κινούμενου αντικείμενου με πολύ μεγάλη ακρίβεια (λίγα εκατοστά).

Από την άλλη, υπάρχει η ανάγκη για πολυάριθμες εφαρμογές που σχετίζονται με κινούμενα αντικείμενα. Κατά τα τελευταία χρόνια έχουμε γίνει μάρτυρες αρκετών εξελίξεων στα κινητά υπολογιστικά συστήματα (mobile computing) και στις ασύρματες τεχνολογίες. Οι συσκευές χειρός, τα έξυπνα κινητά τηλέφωνα και τα ασύρματα τερματικά αυξάνονται καθημερινά σε αριθμό με πολύ γρήγορους ρυθμούς.

Για παράδειγμα, έστω ένα σύνολο από κινούμενους παροχείς υπηρεσιών (ταξί, πωλητές, περιπλανώμενοι ιατροί, κλπ) που στέλνουν τη

θέση τους σε ένα σύστημα και ενδιαφέρονται αυτή η πληροφορία να γίνει γνωστή ευρύτερα. Από την άλλη, υπάρχει ένα σύνολο από καταναλωτές υπηρεσιών (χρήστες) οι οποίοι ενδιαφέρονται να εντοπίζουν παροχές σύμφωνα με κριτήρια που θέτουν (χωρικά, χρονικά, προφίλ κλπ.). Όλη αυτή η πληροφορία ανταλλάσσεται και τη διαχειρίζεται ένα σύστημα παγκόσμιου ιστού. Η ανταλλαγή πληροφορίας μπορεί να είναι αμφίδρομη μεταξύ των εμπλεκόμενων μερών ή ακόμα και μονομερής σε περίπτωση που η ανατροφοδότηση του συστήματος δεν είναι απαραίτητη.

Ευχαριστίες

Στον επιβλέποντα Καθηγητή και επιστημονικό υπεύθυνο της πτυχιακής εργασίας «Σχεδίαση και ανάπτυξη Web 2.0 mashup Διαδικτυακής Εφαρμογής Διαχείρισης Στόλου» Τσελίκια Νίκο, που μας ενθάρρυνε να επιλέξουμε και να ολοκληρώσουμε αυτήν την πτυχιακή εργασία, και μας βοήθησε σε κάθε μας απορία.

Περίληψη

Στην παρούσα πτυχιακή εργασία θα παρουσιαστούν αρχικά οι στόχοι αυτής, οι οποίοι προσπαθήσαμε να είναι ευδιάκριτοι, αφενός για να μπορούμε να εστιάσουμε στη δημιουργία και ανάπτυξη της συγκεκριμένης εφαρμογής αφετέρου για να είναι κατανοητοί και από μη ειδικευμένο κοινό.

Στη συνέχεια θα μελετηθούν οι ανάγκες και τα υπάρχοντα δεδομένα προκειμένου να παρουσιαστούν και να αναλυθούν με τη σειρά που χρησιμοποιήθηκαν οι διάφορες τεχνολογίες (Ajax, Java Script, Google Maps, Joomla!) και γλώσσες προγραμματισμού (Visual Basic, PHP), καθώς και οι λόγοι χρήσης αυτών με σκοπό να γίνει αντιληπτός τόσο ο τρόπος όσο και οι δομημένοι μέθοδοι ανάπτυξης της εφαρμογής.

Επιπλέον θα γίνει πλήρης αναφορά και περιγραφή του κώδικα αλλά και του εξοπλισμού που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής, αλλά και εκτενής ανάλυση των βάσεων δεδομένων MySQL που δημιουργήθηκαν καθώς και οι εγγραφές αυτών.

Τέλος, θα παρουσιαστεί εγχειρίδιο χρήσης της εφαρμογής αλλά και πιθανές επεκτάσεις αυτής δίνοντάς μας τη δυνατότητα προσαρμογής της σε κάθε νέα ανάγκη.

Κεφάλαιο 1

Εισαγωγή

1.1 Δομή Εργασίας

Η δομή της εργασίας έγινε με τέτοιο τρόπο ώστε μέσα από κάθε κεφάλαιο να παρέχεται η απαραίτητη γνώση αλλά και ο τρόπος μεθοδολογίας των ενεργειών αλλά και κατευθύνσεων που ακολουθηθήκαν προκειμένου η μετάβαση στο επόμενο κεφάλαιο/βήμα της εργασίας αυτής να γίνεται με τον ομαλότερο δυνατό τρόπο. Έτσι, στη συνέχεια θα γίνει μια σύντομη περιγραφή του περιεχομένου των κεφαλαίων, προκειμένου να προϋδαστεί ο αναγνώστης για τα άρθρα που θα ακολουθηθούν.

Κεφάλαιο 1.

Στόχοι Εργασίας

Σε αυτό το κεφάλαιο αναφέρονται ξεκάθαρα οι λόγοι για τους οποίους επιλέχτηκε η συγκεκριμένη πτυχιακή εργασία καθώς επίσης και στόχοι που τέθηκαν για να μπορέσουμε να τη φέρουμε εις πέρας.

Μεθοδολογία Εργασίας

Σκοπός του κειμένου είναι να μεταφέρει στον αναγνώστη τον τρόπο με τον οποίο βήμα-βήμα δημιουργήθηκε αυτή η εφαρμογή. Μοναδικός τρόπος υλοποίησής της η ακολουθία μιας συγκεκριμένης μεθόδου, η οποία περιγράφεται αναλυτικά σε αυτό το κεφάλαιο.

Κεφάλαιο 2.

Περιγραφή αναγκών

Βασική επιδίωξη είναι η περιγραφή των αναγκών για τη δημιουργία μιας τέτοιου είδους εφαρμογής και οι τρόποι ικανοποίησης αυτών. Η περιγραφή των αναγκών γίνεται σε κάθε επίπεδο, θέλοντας να τονιστεί ότι η εφαρμογή εξασφαλίζει την απλούστευση της κλήσης ταξί τόσο στον πελάτη και τελικό παραλήπτη του αιτήματος (ταξί) όσο και στον διαχειριστή αυτών.

Εργαλεία ανάπτυξης εφαρμογής και εξοπλισμός

Στο σημείο αυτό γίνεται μια εκτενής αναφορά στα εργαλεία, τις τεχνολογίες, τα προγράμματα αλλά και τον εξοπλισμό που χρησιμοποιήθηκαν για την ολοκλήρωση αυτής της πτυχιακής εργασίας. Σκοπός αυτής της αναφοράς είναι η περαιτέρω πληροφόρηση του αναγνώστη για το σχετικό εργαλείο/τεχνολογία αλλά και η ενημέρωσή του για το σημείο που εμπλέκεται στην εφαρμογή καθώς και ο λόγος που χρησιμοποιήθηκε.

Περιγραφή εφαρμογής

Οι γραμμές αυτές περιγράφουν αναλυτικά τη λειτουργία της εφαρμογής από τη στιγμή της κλήσης ενός ταξί, δηλαδή από τη στιγμή της αποστολής ενός e-mail μέχρι και τη στιγμή που το όχημα λαμβάνει και επιβεβαιώνει το αίτημα που δέχτηκε. Στο κείμενο αυτό γίνεται πλήρης αναφορά του κάθε σταδίου που περνάει το αίτημα, ενώ επίσης παρέχονται screenshots για την ευκολία της κατανόησης της όλης διαδικασίας.

Κεφάλαιο 3.

Σε ολόκληρο το τρίτο κεφάλαιο γίνεται μια πλήρης περιγραφή της εφαρμογής αλλά σε αρκετά χαμηλό επίπεδο (low lever), δηλαδή σε επίπεδο κώδικα. Μέσα από αυτό το κεφάλαιο αναγνώστες που είναι οικείοι με τον προγραμματισμό θα είναι σε θέση να αντιληφθούν τεχνικές προγραμματισμού που χρησιμοποιήθηκαν για την υλοποίηση συγκεκριμένων διεργασιών. Τέλος, γίνεται αναφορά στις βάσεις δεδομένων που δημιουργήθηκαν για να διατηρούν τις εγγραφές του συστήματος καθώς επίσης και ανάλυση κομματιών κώδικα που αφορούν τη δημιουργία και απεικόνιση χαρτών google maps με τη χρήση Ajax.

Κεφάλαιο 4.

Σε αυτό το κεφάλαιο της εργασίας μας αναφέρονται περιληπτικά οι δυνατότητες αλλά και οι δυνατές επεκτάσεις της εφαρμογής με γνώμονα τις σημερινές ανάγκες αλλά και πιθανές απαιτήσεις που υπάρχουν από ένα τέτοιο σύστημα διαχείρισης στόλου. Στόχος του κεφαλαίου αυτού είναι να αναδειχθεί η μεγάλη ευελιξία αλλά και η δυνατότητα προσαρμογής της εφαρμογής σε νέες ανάγκες που πιθανώς να υπάρξουν/προκύψουν στο μέλλον.

Κεφάλαιο 5.

Τέλος, στο κεφάλαιο αυτό γίνεται αναφορά στη βιβλιογραφία και τις πηγές που ανατρέξαμε και χρησιμοποιήθηκαν για την υλοποίηση και ολοκλήρωση της πτυχιακής εργασίας.

1.2 Στόχοι Εργασίας

Κατά την επιλογή της διπλωματικής εργασίας αλλά και κατά τη διάρκεια υλοποίησής της τέθηκαν στόχοι τους οποίους έπρεπε να πραγματοποιήσουμε για να φέρουμε εις πέρας την υλοποίησή της.

Οι στόχοι αυτοί είχαν αρχικά να κάνουν με τη σχεδίαση αλλά και την ανάπτυξη της εφαρμογής, την απλή της χρήση αλλά κυρίως με τη διεύρυνση του δικού μας γνωστικού πεδίου.

Βασισμένοι στα παραπάνω θεωρήσαμε πως οι στόχοι για τη σχεδίαση και ανάπτυξη θα έπρεπε να είναι τέτοιοι ώστε να δημιουργήσουμε μία εφαρμογή η οποία θα εξυπηρετούσε καθημερινές μας ανάγκες. Έτσι, καταλήξαμε στο να σχεδιάσουμε και να αναπτύξουμε μία εφαρμογή για τη διαχείριση στόλου ταξί.

Σαν δεύτερο στόχο θέσαμε την ευχρηστία της εφαρμογής, έχοντας πάντα στο μυαλό μας πως η εφαρμογή θα χρησιμοποιείται από μη ειδικευμένο κοινό. Ο στόχος αυτός υλοποιήθηκε με τη χρήση πολύ φιλικού περιβάλλοντος κατάθεσης των αιτημάτων παραγγελίας ταξί.

Τέλος, βασικότερη επιδίωξή μας είναι ήταν ο εμπλουτισμός των γνώσεών μας αλλά και η οικειοποίηση με νέες τεχνολογίες προκειμένου να διευρύνουμε το γνωστικό μας πεδίο.

Όλα τα παραπάνω δημιουργήθηκαν και παρέμειναν ως κύριοι στόχοι μας από την αρχή μέχρι και την ολοκλήρωση της πτυχιακής εργασίας «ΑΝΑΠΤΥΞΗ WEB 2.0 MASHUP ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΣΤΟΛΟΥ».

1.3 Μεθοδολογία Εργασίας

Η εργασία ξεκίνησε με τη βιβλιογραφική αναζήτηση. Αναζητήθηκαν επιστημονικές δημοσιεύσεις σχετικές με το γενικότερο θέμα της διαχείρισης στόλου και της διαδικτυακής εφαρμογής. Μέσα από την κριτική αξιολόγηση αναγνωρίστηκαν τα βασικά σημεία που πρέπει να έχει μια τέτοια εφαρμογή στα πλαίσια αυτά, αποφασίστηκε η δημιουργία ενός νέου συστήματος διαχείρισης στόλου που θα συνδυάζει τεχνολογίες βάσεων δεδομένων διαδίκτυο και GPS. Πριν την έναρξη ανάπτυξης της εφαρμογής πραγματοποιήθηκε έρευνα για την καταγραφή των βασικών αναγκών μιας τέτοιας εφαρμογής για τη διαχείριση των πληροφοριών και την ανάγκη των χρηστών, έτσι ώστε να είναι απλή, λειτουργική και χρηστική.

Μέσα από την παραπάνω αναζήτηση καταλήξαμε πως η εφαρμογή θα πρέπει να χωριστεί σε δύο τμήματα, στον εξοπλισμό που θα χρησιμοποιηθεί καθώς και στις τεχνολογίες που θα μας επιτρέψουν να κάνουμε την εφαρμογή απλή λειτουργική και χρηστική όπως προαναφέρθηκε.

Με γνώμονα τα παραπάνω η ανάπτυξη της εφαρμογής συμπεριέλαβε τον αναλυτικό σχεδιασμό της σχεσιακής βάσης δεδομένων, μέσα από βρόγχους μοντελοποίησης οντοτήτων και κανονικοποίησης προκειμένου να υπάρχει καταγεγραμμένη αλλά και αποθηκευμένη κάθε πληροφορία που αποστέλλεται από το χρήστη αλλά και από το κάθε επιμέρους όχημα. Οι προαναφερθείσες βάσεις δεδομένων είναι πλήρως εκμεταλλεύσιμες από την εφαρμογή που έχει δημιουργηθεί με VB προκειμένου να μπορούν να αλληλεπιδρούν κατάλληλα μεταξύ τους αλλά και ανεξάρτητα. Επιπλέον, δημιουργήθηκε online φόρμα συμπλήρωσης αιτήματος από τον χρήστη, η οποία αποστέλλεται στη κεντρική εφαρμογή και αποθηκεύεται και αυτή στη βάση δεδομένων. Το αίτημα αυτό διατίθεται στα οχήματα ταξί προκειμένου με τις κατάλληλες ενέργειες να αποδεχθούν ή όχι το αίτημα. Με την παραπάνω μεθοδολογία αλληλεπίδρασης της εφαρμογής και των βάσεων της, αλλά και

των τριών επιμέρους μερών της εφαρμογής (χρήστη, διαχειριστή, τελικό παραλήπτη αιτήματος) μεθοδεύτηκε η εφαρμογή.

Κεφάλαιο 2

Ανάλυση και Περιγραφή Εφαρμογής

2.1 Περιγραφή αναγκών

Αντιλαμβανόμενοι τις καθημερινές ανάγκες για εξοικονόμηση χρόνου και ευκολία στις μετακινήσεις, αλλά - από την άλλη - θεωρώντας ότι η ακρίβεια στην πληροφορία είναι παραπάνω από σημαντική δημιουργήσαμε την εφαρμογή παρακολούθησης στόλου θεωρώντας πως θα ικανοποιήσουμε τον «πελάτη», το κέντρο παρακολούθησης οχημάτων καθώς και τον οδηγό του οχήματος στον μέγιστο βαθμό.

Σκεφτόμενοι σαν «πελάτες» που η χρήση του ταξί είναι καθημερινή ανάγκη, παρατηρήσαμε πως δαπανάται αρκετός χρόνος για την εύρεση ελεύθερου οχήματος είτε αυτό γίνεται από τον ίδιο τον «πελάτη» είτε μέσω τηλεφωνικού κέντρου. Έτσι, η ανάγκη για μείωση τέτοιου είδους χρονικής δαπάνης μας οδήγησε στη λύση δημιουργίας ηλεκτρονικής φόρμας αίτησης ταξί, πράγμα το οποίο κάνει την παραγγελία οχήματος υπόθεση λίγων δευτερολέπτων.

Από την πλευρά του οδηγού του οχήματος, η βασική ανάγκη για λήψη του αιτήματος σε πραγματικό χρόνο η επεξεργασία αυτού αλλά και η ανταπόκριση σε αυτό μας οδήγησε στην ενσωμάτωση ειδικής συσκευής στο ταξί, η οποία επικοινωνεί με το κέντρο λήψης αιτημάτων όχι στηριζόμενη σε παρωχημένες τεχνολογίες όπως UHF, VHF αλλά σε foul IP routing μέσω της χρήσης της τεχνολογίας ασύρματων κυψελωτών συστημάτων 3G/GPRS.

Τέλος, η ανάγκη του διαχειριστή για πληροφορία ακριβή, έγκυρη σε πραγματικό χρόνο αλλά και η ενοποίηση των δύο προαναφερθεισών πλευρών (πελάτης, οδηγός) μας οδήγησε στην ανάπτυξη διαδικτυακής εφαρμογής διαχείρισης στόλου.

2.2 *Εργαλεία ανάπτυξης εφαρμογής και εξοπλισμός*

2.2.1 Εργαλεία ανάπτυξης εφαρμογής

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν αρκετά εργαλεία τα οποία έπαιξαν, το καθένα ξεχωριστά, πολύ σημαντικό ρόλο στην επίτευξη του στόχου που είχε τεθεί. Στη συνέχεια θα αναφερθούν αναλυτικά οι τεχνολογίες αλλά και οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν προκειμένου να γίνει από όλους κατανοητό ο σκοπός και ο λόγος της χρήσης αυτών.

Έτσι με τη σειρά που χρησιμοποιήθηκαν θα κάνουμε μια σύντομη αναφορά:

- Joomla!

Το Joomla! είναι ένα δωρεάν σύστημα διαχείρισης περιεχομένου (Content Management System). Χρησιμοποιείται για τη δημοσίευση περιεχομένου στον παγκόσμιο ιστό και σε τοπικά δίκτυα – intranets. Είναι γραμμένο σε PHP και αποθηκεύει τα δεδομένα στη βάση MySQL. Το βασικό χαρακτηριστικό του είναι ότι οι σελίδες που εμφανίζει είναι δυναμικές, δηλαδή δημιουργούνται τη στιγμή που ζητούνται. Ένα σύστημα διακομιστή (server) όπως είναι ο Apache λαμβάνει τις αιτήσεις των χρηστών και τις εξυπηρετεί.

Με ερωτήματα προς τη βάση λαμβάνει δεδομένα τα οποία μορφοποιεί και αποστέλλει στον εκάστοτε φυλλομετρητή (web browser) του χρήστη. Με χρήση της Joomla! δημιουργήσαμε την online φόρμα αποστολής αιτημάτων του χρήστη.

- Microsoft Visual Studio

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE, integrated development environment) από τη Microsoft που περιλαμβάνει εργαλεία για σχεδίαση φόρμας, για δημιουργία GUI για διαχείριση βάσεων δεδομένων ενώ επίσης υποστηρίζει διάφορα plugins που βελτιώνουν τη λειτουργικότητα. Με τη χρήση του Visual Studio αναπτύχθηκαν η εφαρμογή για τη διαδικτυακή διαχείριση στόλου.

- Microsoft Outlook 2010

Πρόγραμμα ηλεκτρονικής αλληλογραφίας ενσωματωμένο στη σουίτα προγραμμάτων του Microsoft Office. Με τη χρήση του outlook το οποίο λαμβάνει τα e-mails από τους υποψήφιους πελάτες μπορούμε, μέσω της εφαρμογής, να εισάγουμε όλα τα αιτήματα/e-mails στην εφαρμογή μας, να τα επεξεργαστούμε, να τα εγγράψουμε στη σχετική βάση δεδομένων και στη συνέχεια να τα αποστείλουμε στον τελικό παραλήπτη.

- XAMPP

Η ανάγκη για τη δημιουργία ενός web server με ιδιαίτερα χαμηλό κόστος μας οδήγησε στη χρήση του XAMPP, το οποίο είναι ένα χρήσιμο βοήθημα που μας δίνει την δυνατότητα να κάνουμε τον υπολογιστή μας ένα web server και όχι μόνο. Με την χρήση του XAMPP μας δόθηκε η δυνατότητα να χρησιμοποιήσουμε τα παρακάτω πακέτα (Apache 2.2.2, MySQL 5.0.21, phpMyAdmin 2.8.1) υπηρεσιών τα οποία ήταν αναγκαία για την εφαρμογή μας. Η διασύνδεση των βάσεων δεδομένων MySql γίνεται με τη χρήση της γλώσσας προγραμματισμού PHP (παρακάτω γίνεται σχετική αναφορά), η οποία διαθέτει συγκεκριμένες εντολές διασύνδεσης με τις διάφορες βάσεις δεδομένων.

- PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα διακομιστή του Παγκόσμιου Ιστού π.χ. Apache (στην δική μας περίπτωση μέσω XAMPP), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

- HTML (Hyper Text Markup Language)

Η HTML είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Με τη χρήση της HTML μέσω της Joomla! διαμορφώθηκε η online σελίδα όπου ο χρήστης μπορεί να επισκεφτεί για να κατοχυρώσει το αίτημά του.

- Google Maps API

Το Google Maps API αποτελεί μια προγραμματιστική διεπαφή για την ενσωμάτωση χαρτών Google (Google Maps) σε ιστοσελίδες με χρήση JavaScript. Πιο συγκεκριμένα, το API περιλαμβάνει ένα σύνολο λειτουργικότητων και υπηρεσιών οι οποίες παρέχουν τη δυνατότητα για τον χειρισμό χαρτών και τον εμπλουτισμό τους με περιεχόμενο, με στόχο τη δημιουργία προηγμένων και βασισμένων σε χάρτες εφαρμογών μέσω ιστοσελίδων. Με την χρήση των Google Maps απεικονίζουμε τη θέση των οχημάτων σε πραγματικό χρόνο.

Η απεικόνιση αντικειμένων στους χάρτες της Google, γενικά γίνεται είτε ορίζοντας απ' ευθείας σε σχετικό script τα στοιχεία του κάθε αντικειμένου ξεχωριστά, είτε συνδέοντας κάποιο αρχείο τύπου XML με συγκεκριμένο κώδικα στο script είτε συνδέοντας κάποιο αρχείο KML (Keyhole Markup Language) ή και με άλλους τρόπους στους οποίους δε θα αναφερθούμε στην παρούσα εργασία. Στην εφαρμογή, δε θα μπορούσαν να υπάρχουν απ' ευθείας σε script και συνεπώς δημιουργείται διαφορετικό αρχείο XML με τις απαντήσεις του κάθε ερωτήματος και στη συνέχεια τα XML αυτά διαβάζονται από τον κώδικα JavaScript και απεικονίζονται στο χάρτη. Επίσης τα αρχεία XML ανανεώνονται ανά τακτά χρονικά διαστήματα. Η γλώσσα σήμανσης KML (Keyhole Markup Language) χρησιμοποιεί το συντακτικό της XML με σκοπό τη διαμόρφωση και αποθήκευση γεωγραφικών χαρακτηριστικών, όπως σημεία, γραμμές, εικόνες, πολύγωνα και μοντέλα για εμφάνιση στο Google Earth, το Google Maps και άλλες εφαρμογές. Επίσης η KML μπορεί να χρησιμοποιηθεί για την κοινή χρήση μερών και πληροφοριών με άλλους χρήστες αυτών των εφαρμογών. Το Google Earth επεξεργάζεται ένα αρχείο KML με παρόμοιο τρόπο με την επεξεργασία των αρχείων HTML και XML από τα προγράμματα περιήγησης ιστού. Όπως το HTML, το KML διαθέτει δομή βασισμένη σε ετικέτες, χρησιμοποιώντας ονόματα και χαρακτηριστικά για συγκεκριμένους σκοπούς εμφάνισης. Επομένως, το Google Earth λειτουργεί ως πρόγραμμα περιήγησης σε σχέση με τα αρχεία KML

Αρχεία KML, μετά από σχετική διερεύνηση, αποφάνηκε ότι για να απεικονιστούν σε χάρτη GoogleMaps πρέπει να βρίσκονται σε web server της Google και συνεπώς παύει η δυνατότητα ανανέωσης του περιεχομένου ανά τακτά χρονικά διαστήματα μερικών δευτερολέπτων. Από τα παραπάνω διαφαίνεται η αναγκαιότητα χρήσης αρχείων XML σε ένα απόλυτα δυναμικό περιβάλλον όπως αυτό της παρούσας εφαρμογής.

Για την απεικόνιση του χάρτη στην εφαρμογή, την ανάγνωση αρχείων XML, την απεικόνιση σημείων και τροχιών επί του χάρτη, την αυτόματη ανανέωση των αποτελεσμάτων, τη δημιουργία πινάκων με αποτελέσματα ερωτημάτων, τον έλεγχο των πεδίων εισαγωγής πληροφοριών και την επίτευξη λοιπών δυναμικών λειτουργιών της εφαρμογής είναι απαραίτητη η χρήση της JavaScript.

- AJAX (Asynchronous JavaScript And XML)

Με τη χρήση Ajax η οποία δεν είναι μια καινούργια προγραμματιστική γλώσσα αλλά ο συνδυασμός JavaScript και XML επιτυγχάνουμε μικρότερους χρόνους απασχόλησης του server (στη δική μας περίπτωση Apache Server) λόγω του ότι η ανάκτηση των πληροφοριών γίνεται ασύγχρονα. Αποτέλεσμα αυτής της ασύγχρονης ανάκτησης πληροφοριών είναι η αποφυγή επαναφόρτισης ολόκληρης της σελίδας παρά μόνο του τμήματος που περιέχει τα updates. Εφαρμογή της συγκεκριμένης τεχνολογίας στην δική μας περίπτωση έχει γίνει για την αποφυγή της συνεχούς ανανέωσης του χάρτη (Google Maps) κάθε φορά που τα αντικείμενα (οχήματα) αλλάζουν θέση. Θέτοντας λοιπόν σε εφαρμογή την τεχνική AJAX επιτυγχάνουμε την ανανέωση των αντικειμένων και μόνο πάνω στον χάρτη.

2.2.2 Εξοπλισμός εφαρμογής

Για την πλήρη εφαρμογή της τρέχουσας πτυχιακής εργασίας μας χρησιμοποιήθηκε εξοπλισμός τόσο από την πλευρά του πελάτη και του οχήματος όσο και από την πλευρά του διαχειριστή των δύο παραπάνω. Έτσι,

στη συνέχεια θα γίνει μια σύντομη αναφορά του απαιτούμενου εξοπλισμού για την υλοποίηση της εφαρμογής.

Πελάτης

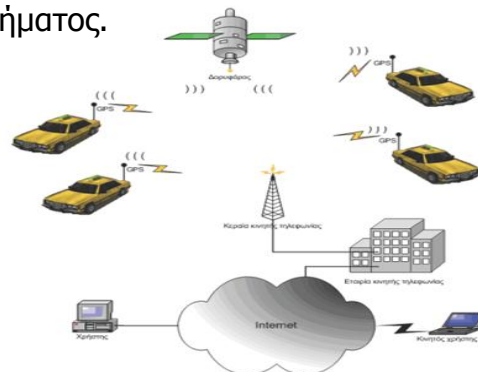
Από την πλευρά του πελάτη η μόνη απαραίτητη προϋπόθεση είναι η προσβασιμότητα στον παγκόσμιο ιστό για να μπορέσει να κάνει την κλήση του οχήματος. Την πρόσβαση σε αυτή μπορεί να την έχει είτε μέσω ενός έξυπνου κινητού τηλεφώνου με πρόσβαση στο internet, είτε μέσω ενός Η/Υ είτε μέσω οποιασδήποτε άλλης συσκευής που του δίνει την δυνατότητα να έχει πρόσβαση στο internet.

Όχημα

Από την πλευρά του οχήματος απαραίτητη προϋπόθεση για τη λήψη των αιτημάτων από τον διαχειριστή της εφαρμογής αλλά και την απάντηση αυτών είναι η κατοχή ενός ηλεκτρονικού υπολογιστή (tablet, smartphone κτλ) με πρόσβαση στο διαδίκτυο και ενσωματωμένο ή μη δέκτη GPS προκειμένου ο διαχειριστής της εφαρμογής να λαμβάνει σε πραγματικό χρόνο της ανταποκρίσεις των αιτημάτων καθώς και τις τρέχουσες συντεταγμένες των οχημάτων του στόλου.

Διαχειριστής

Τέλος, από την πλευρά του διαχειριστή χρειάζεται ένας ηλεκτρονικός υπολογιστής με εγκατεστημένη την εφαρμογή διαχείρισης στόλου, με πρόσβαση στο internet προκειμένου να λαμβάνει τα αιτήματα των πελατών, να τα επεξεργάζεται, να τα αποστέλλει στα οχήματα, να λαμβάνει τις απαντήσεις αυτών, να απεικονίζει την ακριβή θέση των οχημάτων σε Google Maps και τέλος να επιστρέφει στον τελικό πελάτη την επιβεβαίωση για την «κράτηση» του οχήματος.



2.3 Περιγραφή εφαρμογής

Η εφαρμογή της οποίας η ανάλυση θα γίνει τμηματικά (πελάτης, διαχειριστής, όχημα) σε αυτό το κεφάλαιο, έχει ως σκοπό την εξυπηρέτηση ενός αιτούμενου για ταξί πελάτη. Η εφαρμογή διαχωρίζεται σε τρία μέρη, πρώτο μέρος η αίτηση για κλήση του ταξί, δεύτερο μέρος η διαχείριση και παρακολούθηση του αιτήματος από τον διαχειριστή και τρίτο μέρος η αποδοχή ή απόρριψη του αιτήματος από τα οχήματα. Σε όλη αυτήν τη διαδικασία η μόνη ενέργεια που απαιτείται από τον αιτούμενο είναι η συμπλήρωση μιας ηλεκτρονικής online φόρμας. Από εκεί και έπειτα η εφαρμογή διαχείρισης αναλαμβάνει τη λήψη του αιτήματος, την επιβεβαίωση αυτού, την αναζήτηση του πλησιέστερου οχήματος, τη δέσμευσή του και τέλος την απεικόνιση του οχήματος σε Google maps με τη χρήση Ajax τεχνολογίας.

Η εφαρμογή έχει σαν σκοπό τη διαχείριση πληροφοριών και τη παρακολούθηση συγκεκριμένης ομάδας κινούμενων αντικειμένων σε πραγματικό χρόνο από οποιονδήποτε υπολογιστή που έχει εγκατεστημένη την εφαρμογή. Με βάση τα παραπάνω θα ακολουθηθεί αναλυτική περιγραφή των μερών της εφαρμογής.

A. Αίτημα για ταξί.

Ο πελάτης θέλοντας να καλέσει ταξί, η μοναδική ενεργεία που έχει να κάνει

Ονοματεπώνυμο:

Αναχώρηση από:

Προορισμός:

Ωρα αναχώρησης:

Διεύθυνση Ηλεκτρονικού Ταχυδρομείου:

Τηλέφωνο Επικοινωνίας:

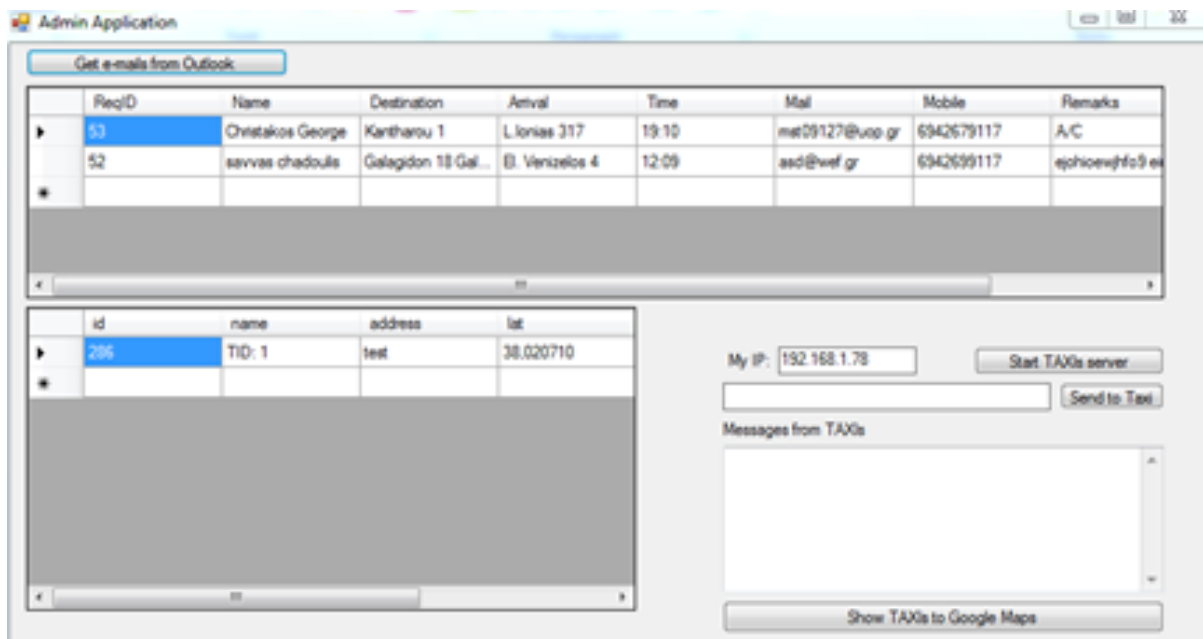
Παρατηρήσεις:

E-mail a copy of this message to your own address.

είναι να επισκεφτεί την ιστοσελίδα που έχει δημιουργηθεί για το σκοπό αυτό, www.kotronas.gr, να συμπληρώσει τα ανάλογα πεδία, - δηλαδή το ονοματεπώνυμο του αιτούντος, τη διεύθυνση αναχώρησης αλλά και τη διεύθυνση προορισμού, την ώρα άφιξης του ταξί στη διεύθυνση αναχώρησης, ένα e-mail επικοινωνίας προκειμένου να γίνει η επιβεβαίωση της λήψης του αιτήματος, ένα τηλέφωνο επικοινωνίας καθώς και επιπλέον πληροφορίες ή απαιτήσεις που έχει – και να κάνει την αποστολή του αιτήματος πατώντας το κουμπί «send». Στη συνέχεια, το αίτημα παραλαμβάνεται από το διαχειριστή του οποίου οι ενέργειες περιγράφονται στη συνέχεια.

B. Διαχείριση και παρακολούθηση του αιτήματος

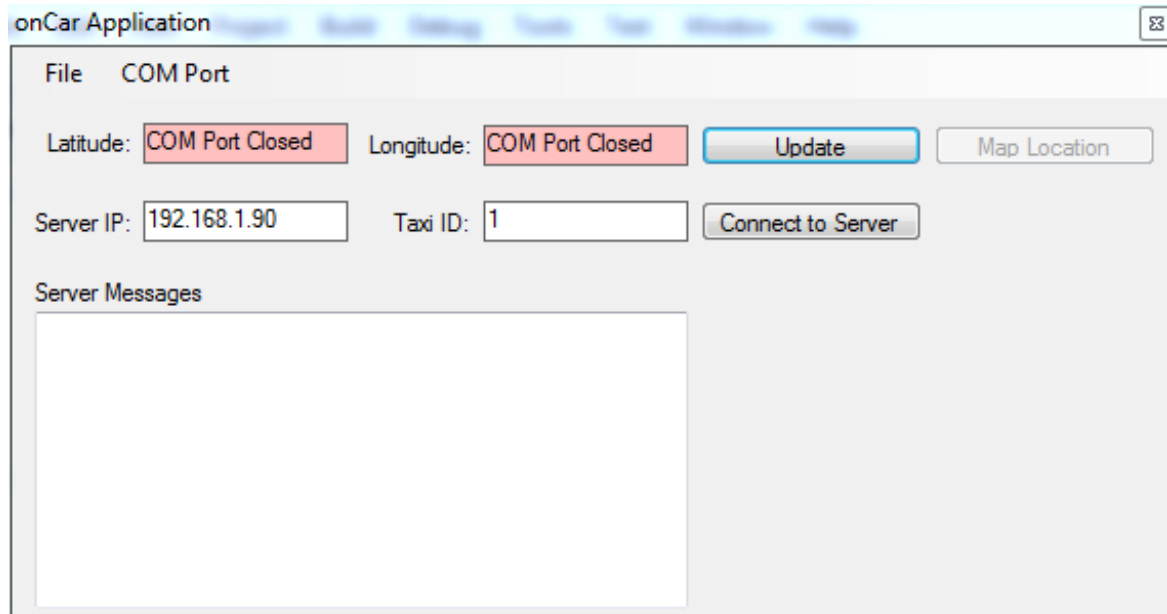
Μετά την αποστολή του αιτήματος από τον πελάτη μέσω της ηλεκτρονικής online φόρμας όπου συμπλήρωσε, ο διαχειριστής λαμβάνει ένα e-mail στο πρόγραμμα διαχείρισης του ηλεκτρονικού ταχυδρομείου. Μέσα από την εφαρμογή και κατόπιν σύνδεσης του προγράμματος ηλεκτρονικού ταχυδρομείου με αυτή, το αίτημα εισέρχεται στην εφαρμογή, (εισαγωγή μέσω ειδικού πλήκτρου στην εφαρμογή «Get e-mails from outlook») δέχεται την κατάλληλη επεξεργασία, εγγράφεται στη σχετική βάση δεδομένων (MySQL) και αποστέλλεται κατευθείαν σε όλα τα οχήματα που είναι συνδεδεμένα με την εφαρμογή του διαχειριστή αφού πρωτίστως έχει ενεργοποιηθεί το σχετικό



Service, το οποίο επιτρέπει την επικοινωνία μεταξύ διαχειριστή και οχήματος σε layer 3 επίπεδο (IP). Από εκεί και ύστερα υπάρχει ανοικτό κανάλι επικοινωνίας του διαχειριστή και του οχήματος είτε για να λαμβάνει το όχημα καινούρια αιτήματα είτε για να μπορεί ο διαχειριστής να του περάσει σε κείμενο οποιαδήποτε πληροφορία χρειαστεί οποιαδήποτε στιγμή, αλλά κυρίως για να μπορεί ο τελευταίος να λαμβάνει σε πραγματικό χρόνο τις συντεταγμένες του οχήματος όπου η εφαρμογή αρχικά τις αποθηκεύει σε μια νέα βάση δεδομένων (MySQL) και στη συνέχεια τις απεικονίζει σε χάρτες (Google maps) για την παρακολούθηση της κίνησης των οχημάτων. Τέλος, η εφαρμογή διατηρεί μία ακόμα βάση δεδομένων (MySQL) με τον αριθμό, το χαρακτηριστικό του κάθε ταξί, τις συντεταγμένες αλλά και γενικότερες πληροφορίες των οχημάτων που βρίσκονται συνδεδεμένα με αυτήν, ώστε να υπάρχει ανά πάσα στιγμή η δυνατότητα ανάκτησης περαιτέρω πληροφοριών.

Αποδοχή ή απόρριψη του αιτήματος από τα οχήματα

Για να υπάρχει επικοινωνία μεταξύ του οχήματος και του διαχειριστή της εφαρμογής πέρα από την ενεργοποίηση του service που προαναφέρθηκε στην πλευρά του διαχειριστή, χρειάζεται και σύνδεση του οχήματος με την εφαρμογή. Έτσι, κατά την ενεργοποίηση της εφαρμογής στο ταξί το μόνο που πρέπει να συμβεί προκειμένου το κάθε όχημα να είναι σε θέση να λάβει τα διάφορα αιτήματα ώστε να μπορεί να τα διαχειριστεί, είναι να δηλώσει την IP του ηλεκτρονικού υπολογιστή του διαχειριστή και να πατήσει το κουμπί «connect to server». Από εκεί και έπειτα υπάρχει ένα μόνιμο ανοικτό κανάλι επικοινωνίας το οποίο επιτρέπει αμφίδρομη μετάδοση πληροφορίας.



Κάθε φορά που το όχημα λαμβάνει ένα νέο αίτημα, αυτόματα εμφανίζεται στη οθόνη της ηλεκτρονικής συσκευής ένα παράθυρο αποδοχής ή απόρριψης του αιτήματος. Αν ο οδηγός του ταξί επιλέξει αποδοχή τότε σε λίγα δευτερόλεπτα λαμβάνει ένα μήνυμα επιβεβαίωσης από τον διαχειριστή ώστε να είναι ξεκάθαρο πως η ανάθεση του αιτήματος έχει δοθεί στο συγκεκριμένο όχημα. Στη συνέχεια η συσκευή του ταξί αρχίζει να στέλνει περιοδικά και ανά πολύ μικρά χρονικά διαστήματα τις τρέχουσες συντεταγμένες του προκειμένου αυτές να λαμβάνονται από την εφαρμογή του διαχειριστή και να απεικονίζονται στους χάρτες (Google maps.).

Σε αντίθετη περίπτωση που ο οδηγός του ταξί απορρίψει το αίτημα, δεν υπάρχει κανένα συμβάν, το νέο αίτημα θα εμφανιστεί και πάλι στην οθόνη της ηλεκτρονικής συσκευής του ταξί και το δικαίωμα επιλογής ή απόρριψης είναι και πάλι θέμα επιλογής.

Τέλος, κατά την πρώτη και μόνο φορά της εγκατάστασης της εφαρμογής στη συσκευή του οχήματος και λόγω του ότι η συσκευή GPS που χρησιμοποιούμε είναι εξωτερική θα πρέπει να δηλώσουμε τη θύρα σύνδεσης αυτής. (ComPort 1, ComPort2 etc.)

Κεφάλαιο 3

Περιγραφή της Βάσης Δεδομένων και ανάλυση Κώδικα

3.1 Βάσεις Δεδομένων

Για την εγγραφή διάφορων απαραίτητων για την εφαρμογή πληροφοριών καθώς και για τη διαχείριση αλλά και την ανεύρεση αυτών, απαραίτητη ήταν η χρήση βάσης δεδομένων MySQL την οποία ονομάσαμε google, λόγω του ότι περιέχει και πληροφορίες σχετικά με την εγγραφή δεδομένων που έχουν να κάνουν με την απεικόνιση των οχημάτων σε google maps.

Η βάση αυτή είναι πλήρως προσπελάσιμη, διαχειρίσιμη αλλά και εκμεταλλεύσιμη από την εφαρμογή, αφού μπορούμε μέσω «Data sources» να κάνουμε τη σύνδεση με αυτή. Η βάση αυτή περιέχει δύο πίνακες, οι οποίοι περιέχουν δυναμικές πληροφορίες.

Ο πρώτος πίνακας ο οποίος ονομάζεται "markers" περιέχει πληροφορίες σχετικά με τα οχήματα, δηλαδή τα ταξί, τα οποία συνδέονται με την εφαρμογή του διαχειριστή. Η βάση αυτή εμπλουτίζεται κάθε φορά που ένα ταξί συνδέεται με την εφαρμογή και έτσι δίνει τη δυνατότητα στον διαχειριστή μέσα από ένα γραφικό περιβάλλον να μπορεί να επικοινωνεί με συγκεκριμένο το όχημα. Επιπλέον, σε αυτόν τον πίνακα πέρα από μοναδικά χαρακτηριστικά του οχήματος αποθηκεύονται και οι συντεταγμένες της θέσης του οχήματος προκειμένου να γίνει αργότερα η απεικόνιση στα google maps. Τέλος, στη συνέχεια παραθέεται ο κώδικας που χρησιμοποιήθηκε για τη δημιουργία του πίνακα ο οποίος θα μπορούσε να δημιουργηθεί επίσης και μέσω του xampp.

```
CREATE TABLE IF NOT EXISTS `markers` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(60) NOT NULL,  
  `address` varchar(80) NOT NULL,  
  `lat` decimal(10,6) NOT NULL,  
  `lng` decimal(10,6) NOT NULL,  
  `type` varchar(30) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=287 ;
```

Ο δεύτερος πίνακας ο οποίος ονομάζεται "requests" περιέχει πληροφορίες οι οποίες σχετίζονται με τα αιτήματα των πελατών. Στον πίνακα αυτόν εγγράφονται όλα τα στοιχεία τα οποία εμπεριέχονται στη φόρμα που συμπληρώνει ο πελάτης online. Τα στοιχεία που εγγράφονται παραμετροποιούνται ανάλογα και στέλνονται σε κάθε όχημα που είναι συνδεδεμένο με την εφαρμογή του διαχειριστή. Μέσω της βάσης αυτής μπορούν σε μεταγενέστερες εκδόσεις της εφαρμογής, να εξάγονται στατιστικά για τους πιο συχνούς προορισμούς, πελάτες κτλ. Στη συνέχεια παραθέτεται ο κώδικας που χρησιμοποιήθηκε για τη δημιουργία του πίνακα ο οποίος και πάλι θα μπορούσε να δημιουργηθεί μέσω χαμπρ από ένα εύχρηστο και γραφικό περιβάλλον.

```
CREATE TABLE IF NOT EXISTS `requests` (  
  `RequestID` int(11) NOT NULL AUTO_INCREMENT,  
  `Name` text NOT NULL,  
  `Destination` text NOT NULL,  
  `Arrival` text NOT NULL,  
  `Time` text NOT NULL,  
  `Mail` text NOT NULL,  
  `Mobile` text NOT NULL,  
  `Remarks` text NOT NULL,  
  `AcceptedTaxiID` int(11) DEFAULT NULL,  
  PRIMARY KEY (`RequestID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=56 ;
```

3.2 Πελάτης

Για τον υποψήφιο πελάτη η μόνη απαραίτητη από την πλευρά του ενέργεια είναι να επισκεφθεί τον σύνδεσμο www.kotronas.gr και από εκεί επιλέγοντας το πεδίο επικοινωνία να συμπληρώσει τη φόρμα με τα απαραίτητα πεδία προκειμένου να αρχίσει η διαδικασία παραγγελίας και παρακολούθησης του ταξί από την πλευρά του διαχειριστή.

Όνοματεπώνυμο:

Αναχώρηση από:

Προορισμός:

Ώρα αναχώρησης:

Διεύθυνση Ηλεκτρονικού Ταχυδρομείου:

Τηλέφωνο Επικοινωνίας:

Παρατηρήσεις:

E-mail a copy of this message to your own address.

Η παραπάνω φόρμα καθώς και η αποστολή αυτής στο e-mail του διαχειριστή έχει δημιουργηθεί χρησιμοποιώντας Joomla και παραμετροποιώντας τον παρακάτω κώδικα ανάλογα, προκειμένου να δημιουργήσουμε τα πεδία που χρειαζόμασταν για τη ανάγκη της εφαρμογής μας.

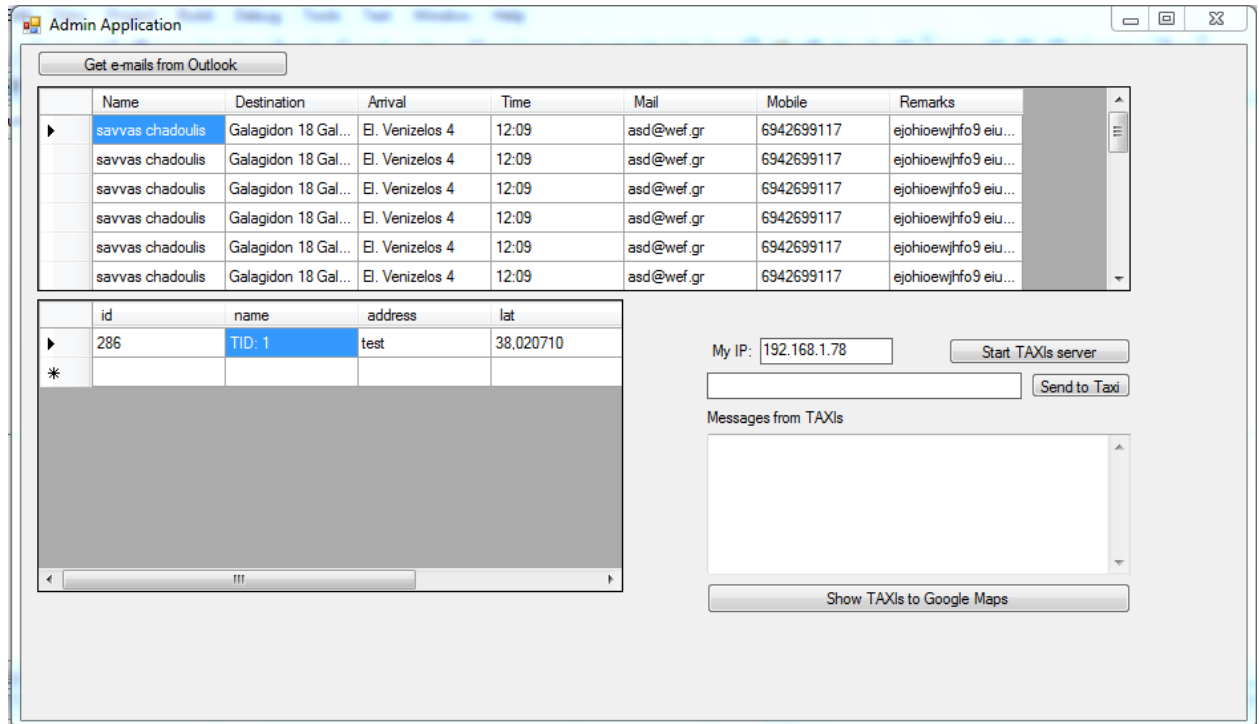
```
1 <?php
2 /** $Id: default_form.php 11917 2009-05-29 19:37:05Z ian $ */
3 defined( '_JEXEC' ) or die( 'Restricted access' );
4
5 $script = '<!--
6     function validateForm( frm ) {
7         var valid = document.formvalidator.isValid(frm);
8         if (valid == false) {
9             // do field validation
10            if (frm.email.invalid) {
11                alert( "" . JText::_ ( 'Please enter a valid e-mail address.', true ) . "" );
12            } else if (frm.text.invalid) {
13                alert( "" . JText::_ ( 'CONTACT_FORM_NC', true ) . "" );
14            }
15            return false;
16        } else {
17            frm.submit();
18        }
19    }
20    // -->';
21 $document =& JFactory::getDocument();
22 $document->addScriptDeclaration($script);
23
24 if(isset($this->error)) : ?>
25 <tr>
26     <td><?php echo $this->error; ?></td>
27 </tr>
28 <?php endif; ?>
29 <tr>
30     <td colspan="2">
31         <br /><br />
32         <form action="<?php echo JRoute::_ ( 'index.php' ); ?>" method="post" name="emailForm" id="emailForm" class="form-validate">
33             <div class="contact_email"<?php echo $this->escape($this->params->get('pageclass_sfx')); ?>">
34
35                 <label for="contact_name">
36                     &nbsp;<?php echo JText::_ ( 'Όνοματεπώνυμο' ); ?>:
37                 </label>
38                 <br />
39                 <input type="text" name="name" id="contact_name" size="30" value="" class="inputbox required" maxlength="100" />
40                 <br />
41
42                 <label id="dep_add" for="departure_address">
43                     &nbsp;<?php echo JText::_ ( 'Αναχώρηση από' ); ?>:
44                 </label>
45                 <br />
46                 <input type="text" id="departure_address" name="daddress" size="30" value="" class="inputbox required" maxlength="100" />
47                 <br />
```

```

49 <label id="arr_add" for="arrival_address">
50    <?php echo JText::_('Προορισμός');?>:
51 </label>
52 <br />
53 <input type="text" id="arrival_address" name="raddress" size="30" value="" class="inputbox required " maxlength="100" />
54 <br />
55
56 <label id="dep_time" for="departure_time">
57    <?php echo JText::_('Ωρα αναχώρησης');?>:
58 </label>
59 <br />
60 <input type="text" id="departure_time" name="dtime" size="30" value="" class="inputbox required" maxlength="5" />
61 <br />
62
63 <label id="contact_emailmsg" for="contact_email">
64    <?php echo JText::_('Διεύθυνση Ηλεκτρονικού Ταχυδρομείου');?>:
65 </label>
66 <br />
67 <input type="text" id="contact_email" name="email" size="30" value="" class="inputbox required validate-email" maxlength="100" />
68 <br />
69
70 <label id="contact_mob" for="contact_mobile">
71    <?php echo JText::_('Τηλέφωνο Επικοινωνίας');?>:
72 </label>
73 <br />
74 <input type="text" id="contact_mobile" name="cmobile" size="30" value="" class="inputbox required " maxlength="10" />
75 <br />
76
77
78 <label id="contact_textmsg" for="contact_text">
79    <?php echo JText::_('Παρατηρήσεις');?>:
80 </label>
81 <br />
82 <textarea cols="50" rows="10" name="text" id="contact_text" class="inputbox "></textarea>
83
84
85
86 <?php if ($this->contact->params->get( 'show_email_copy' ) ) : ?>
87 <br />
88 <input type="checkbox" name="email_copy" id="contact_email_copy" value="1" />
89 <label for="contact_email_copy">
90    <?php echo JText::_('EMAIL_A_COPY'); ?>
91 </label>
92 <?php endif; ?>
93 <br />
94 <br />
95 <button class="button validate" type="submit"><?php echo JText::_('Send'); ?></button>
96 </div>
97
98 <input type="hidden" name="option" value="com_contact" />
99 <input type="hidden" name="view" value="contact" />
100 <input type="hidden" name="id" value="<?php echo $this->contact->id; ?>" />
101 <input type="hidden" name="task" value="submit" />
102 <?php echo JHTML::_('form.token'); ?>
103 </form>
104 <br />
105 </td>
106 </tr>

```

3.3 Διαχειριστής



Με το παρακάτω κομμάτι κώδικα, λαμβάνουμε και διαμορφώνουμε ανάλογα τα αιτήματα που έχουν αποσταλεί με e-mail και τα αποθηκεύουμε μέσω της εφαρμογής του διαχειριστή στη βάση δεδομένων. Με αυτόν τον τρόπο ο διαχειριστής είναι σε θέση να προωθήσει τα αιτήματα σε όλα τα οχήματα, αναμένοντας από αυτά την επιβεβαίωση. Στη συνέχεια, ο διαχειριστής στέλνοντας ένα μήνυμα επιβεβαίωσης στο πλησιέστερο για τη διαδρομή όχημα, του αναθέτει με αυτόν τον τρόπο το δρομολόγιο.

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    ' Create Outlook application.
    Dim oApp As Microsoft.Office.Interop.Outlook.Application =
New Microsoft.Office.Interop.Outlook.Application()

    ' Get Mapi Namespace.
    Dim oNS As Outlook.NameSpace = oApp.GetNamespace("mapi")
    oNS.Logon("YourValidProfile", Missing.Value, False, True) '
TODO:

    ' Get Messages collection of Inbox.
    Dim oInbox As Outlook.MAPIFolder =
oNS.GetDefaultFolder(Outlook.OlDefaultFolders.olFolderInbox)

```

```
Dim oItems As Outlook.Items = oInbox.Items
'MsgBox("Total : " & oItems.Count)

' Get unread e-mail messages.
oItems = oItems.Restrict("[Unread] = true")
'MsgBox("Total Unread : " & oItems.Count)

' Loop each unread message.
Dim oMsg As Outlook.MailItem
Dim i As Integer

For i = 1 To oItems.Count
    oMsg = oItems.Item(i)
    Dim tmpBody() As String

    tmpBody = oMsg.Body.Substring(60).Split("#")
    AddNewRequest(tmpBody)

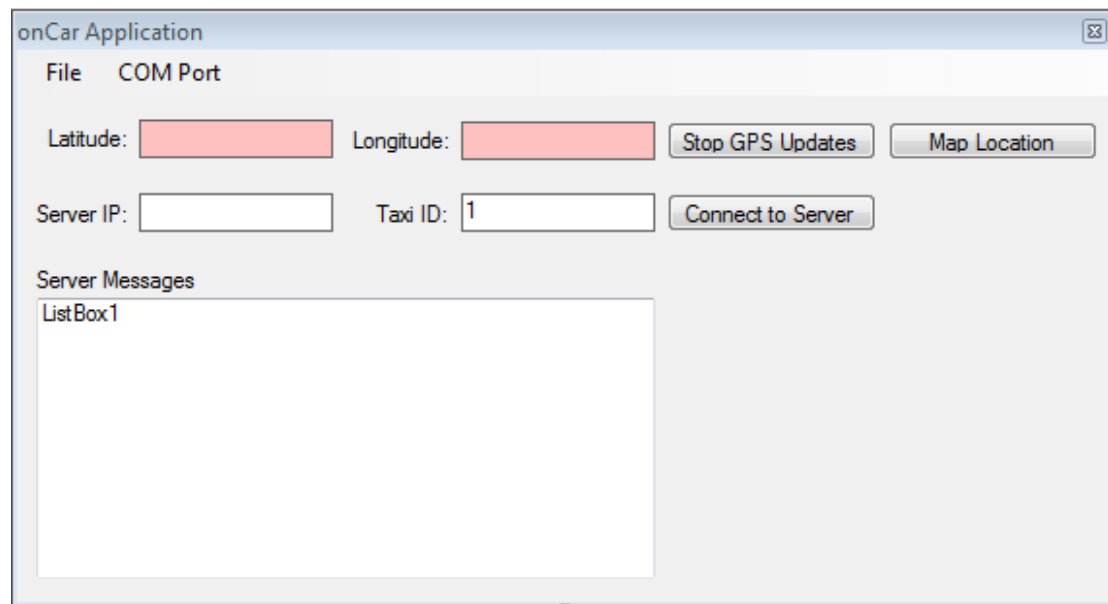
    Dim tmpMessage As String
    tmpMessage = "Request: "
    tmpMessage = tmpMessage & tmpBody(3) & "#" & tmpBody(2) &
    "#" & tmpBody(4)

    SendToAllTaxis(tmpMessage)
Next

' Log off.
oNS.Logoff()

' Clean up.
oApp = Nothing
oNS = Nothing
oItems = Nothing
oMsg = Nothing
End Sub
```

3.4 Ταξί



Ορισμός της IP του διαχειριστή και σύνδεση του ταξί με αυτόν ώστε να είναι εφικτή η αποστολή των δεδομένων σε αυτόν για την απεικόνισή τους στον χάρτη.

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    My.Settings.ServerIP = TextBox1.Text
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    'Try
    If tcpClient.Client.Connected Then
        Timer2.Enabled = False
        tcpClient.Client.Close()
    End If
    myIP = Net.IPAddress.Parse(My.Settings.ServerIP)
    tcpClient = New System.Net.Sockets.TcpClient()
    tcpClient.Client.Connect(myIP, 8002)
    Timer2.Enabled = True
    'Catch
    'Exit Sub
    'End Try
End Sub
```

Διαβάζει το usb GPS από συγκεκριμένη πόρτα που το έχουμε θέσει, μετατρέπει τις συντεταγμένες σε μορφή τέτοια ώστε να μπορούν να αποθηκευτούν στη βάση δεδομένων ως lat & long.

```
Private Sub timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles timer1.Tick

    If SerialPort1.IsOpen Then
```

```
Dim data As String = SerialPort1.ReadExisting()
Dim strArr() As String = data.Split("$")
Dim i As Integer = 0

If strArr.Length > 1 Then

    Try

        For i = 0 To strArr.Length

            Dim strTemp As String = strArr(i)
            Dim lineArr() As String = strTemp.Split(",")

            If (lineArr(0) = "GPGGA") Then

                Try

                    ' Latitude
                    Dim dLat As String =
Convert.ToDouble(lineArr(2)).ToString

                    Latitude = dLat.Substring(0, 2)
                    Latitude = Latitude & "."
                    Latitude = Latitude &
(Convert.ToDecimal(dLat.Substring(2, 3) & "." & dLat.Substring(5, 3))
/ 60 * 100).ToString("000000")

                    ' Longitude
                    Dim dLon As String =
Convert.ToDouble(lineArr(4))

                    Longitude = dLon.Substring(0, 2)
                    Longitude = Longitude & "."
                    Longitude = Longitude &
(Convert.ToDecimal(dLon.Substring(2, 3) & "." & dLon.Substring(5, 3))
/ 60 * 100).ToString("000000")

                    ' Display
                    txtLat.Text = Latitude
                    txtLong.Text = Longitude

                    btnMapIt.Enabled = True

                Catch

                    ' Can't Read GPS values
                    txtLat.Text = "GPS Unavailable"
                    txtLong.Text = "GPS Unavailable"
                    btnMapIt.Enabled = False

                End Try

            End If

        Next

    Catch
        'do nothing
    End Try

End Try
```

```
        End If

    Else

        txtLat.Text = "COM Port Closed"
        txtLong.Text = "COM Port Closed"
        btnMapIt.Enabled = False

    End If

    If Not tcpClient.Connected Then Exit Sub
    If PrevLat <> txtLat.Text Or PrevLon <> txtLong.Text Then
        Dim newMessage As String
        Dim TaxiID As String = TextBox2.Text
        newMessage = "TID: " & TaxiID & "-LAT: " & txtLat.Text &
"-LON: " & txtLong.Text
        SendToAdmin(newMessage)
        PrevLat = txtLat.Text
        PrevLon = txtLong.Text
    End If
End Sub
```

Timer με τον οποίο καθορίζουμε το ρυθμό ανανέωσης των στιγμάτων του GPS.

```
Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUpdate.Click

    ' cycle timer
    If timer1.Enabled = True Then
        timer1.Enabled = False
    Else
        timer1.Enabled = True
    End If

    ' update button label
    If btnUpdate.Text = "Update" Then
        btnUpdate.Text = "Stop Updates"
    Else
        btnUpdate.Text = "Update"
    End If

End Sub
```

3.5 Google Maps

Για την ακριβή απεικόνιση των οχημάτων πάνω στο χάρτη αλλά και την συνεχή ανανέωση αυτών έγινε χρήση "google maps" αλλά και χρήση της τεχνολογίας "ajax" των οποίων ο κώδικας παρατίθεται και επεξηγεται στη συνέχεια.

Η ανάγκη για τη χρήση της τεχνολογίας ajax προέκυψε από την ανάγκη της ανανέωσης της θέσης των οχημάτων χωρίς την επαναφόρτωση ολόκληρου του χάρτη αλλά μόνο των σημείων που παριστάνουν οχήματα. Έτσι, κάθε φορά που ένας συγκεκριμένος χρονιστής (timer) ολοκληρώνει τον κύκλο του, μια συγκεκριμένη ρουτίνα ενεργοποιείται στον κώδικα και έτσι διαβάζονται τα νέα δεδομένα, δηλαδή οι συντεταγμένες των οχημάτων και απεικονίζονται στον χάρτη χωρίς να υπάρχει ανάγκη για επαναφόρτωση όλης της σελίδας.

Στον παρακάτω κώδικα φαίνεται η φόρτωση του Google maps API με χρήση του tag `<script>`, καθώς επίσης στη συνέχεια της ίδιας γραμμής το κλειδί που πήραμε για την παραχώρηση των google maps αλλά και ο ορισμός της γλώσσας που προτιμάμε να εμφανίζονται τα controls.

Στη συνέχεια δημιουργούμε τις πινέζες όπου θα χρησιμοποιηθούν για την απεικόνιση των οχημάτων στο χάρτη θέτοντάς τους τις ιδιότητές τους, μέγεθος εικόνας, σκιάς κτλ. Στη συνέχεια τρέχει ο κώδικας που βρίσκεται μέσα στο body καλώντας τη ρουτίνα load. Το επόμενο βήμα είναι η απεικόνιση του χάρτη δημιουργώντας το αντικείμενο map. Επιπλέον, μέσα σε αυτή τη ρουτίνα θέτουμε τα controls που θέλουμε να υπάρχουν στον χάρτη καθώς και το σημείο όπου θα κεντράρει ο χάρτης κατά την εκκίνησή του. Στο τέλος αυτής της ρουτίνας καλείται η ρουτίνα refresh markers, όπου στην πραγματικότητα είναι μια ρουτίνα η οποία διαβάζει τα δεδομένα ενός .xml αρχείου, (θα γίνει αναφορά στη συνέχεια), αρχείου που παίρνει δεδομένα σε πραγματικό χρόνο από μια βάση δεδομένων, και με το κάλεσμα της ρουτίνας create marker τα απεικονίζει με τη μορφή πινεζών στο χάρτη.


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Google Maps AJAX + MySQL/PHP Example</title>
    <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAR0TWF73ru1OP_SnETQPFKxRj5djmSsmVAgDaRb1psFcJ1ThRhXsJxifqM96NjrBsBn2XrZWSE-QQqQ"
      type="text/javascript"></script>
    <script type="text/javascript">
      //
var iconBlue = new GIcon();
iconBlue.image = 'http://labs.google.com/ridefinder/images/mm_20_blue.png';
iconBlue.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
iconBlue.iconSize = new GSize(12, 20);
iconBlue.shadowSize = new GSize(22, 20);
iconBlue.iconAnchor = new GPoint(6, 20);
iconBlue.infoWindowAnchor = new GPoint(5, 1);

var iconRed = new GIcon();
iconRed.image = 'http://labs.google.com/ridefinder/images/mm_20_red.png';
iconRed.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
iconRed.iconSize = new GSize(12, 20);
iconRed.shadowSize = new GSize(22, 20);
iconRed.iconAnchor = new GPoint(6, 20);
iconRed.infoWindowAnchor = new GPoint(5, 1);

var customIcons = [];
customIcons["vehicle1"] = iconBlue;
customIcons["vehicle2"] = iconRed;

var map = null;
var timeOut=null;

function load() {
  if (GBrowserIsCompatible()) {
    map = new GMap2(document.getElementById("map"));
    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());
    map.setCenter(new GLatLng(37.949159, 23.638787), 7);
    map.enableScrollWheelZoom();
  }
}</pre></div><div data-bbox="144 895 387 912" data-label="Page-Footer">Χρηστάκος Γεώργιος, Οκτώβριος 2011</div><div data-bbox="565 895 800 912" data-label="Page-Footer">ΠΜΣ Πανεπιστημίου Πελοποννήσου</div><div data-bbox="818 923 853 942" data-label="Page-Footer">38</div>
```

```

        refreshMarkers();
    }
}

function refreshMarkers()
{
    clearTimeout(timeOut);
    GDownloadUrl("phpsqlajax_genxml3.php", function (data) {

        var xml = new GXml.parse(data);
        var markers = xml.documentElement.getElementsByTagName("marker");

        map.clearOverlays();

        for (var i = 0; i < markers.length; i++) {
            var name = markers[i].getAttribute("name");
            var address = markers[i].getAttribute("address");
            var type = markers[i].getAttribute("type");
            var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
                parseFloat(markers[i].getAttribute("lng")));
            var marker = createMarker(point, name, address, type);
            map.addOverlay(marker);
        }
        xml=null;
    } );

    timeOut=setTimeout("refreshMarkers()",1000);
}

function createMarker(point, name, address, type) {
    var marker = new GMarker(point, customIcons[type]);
    var html = "<b>" + name + "</b> <br/>" + address;
    GEvent.addListener(marker, 'click', function() {
        marker.openInfoWindowHtml(html);
    });
    return marker;
}
//]]>
</script>
</head>

<body onload="load()" onunload="GUnload()">
    <div id="map" style="width: 100%; height: 600px" ></div>

```

Όπως προαναφέρθηκε τα δεδομένα τα οποία χρησιμοποιούνται για την απεικόνιση των πινεζών στο χάρτη διαβάζονται κατά το κάλεσμα της ρουτίνας «refresh markers» από ένα .xml αρχείο, το οποίο συγκεντρώνει τα δεδομένα από τη βάση mysql όπου εγγράφονται. Στη συνέχεια παρατίθεται ο κώδικας ο οποίος χρησιμοποιείται για να πάρει τα δεδομένα από τη βάση, και να δημιουργήσει ένα .xml αρχείο το οποίο θα περιέχει τις απαραίτητες πληροφορίες για την απεικόνιση των πινεζών.

```
<?php
$username = "root";
$password = "";
$database = "google";

// Start XML file, create parent node
$dom = new DOMDocument("1.0");
$node = $dom->createElement("markers");
$parnode = $dom->appendChild($node);

// Opens a connection to a MySQL server
$connection=mysql_connect (localhost, $username, $password);
if (!$connection) { die('Not connected : ' . mysql_error());}

// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

// Select all the rows in the markers table
$query = "SELECT * FROM markers WHERE 1";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");

// Iterate through the rows, adding XML nodes for each
while ($row = @mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    $node = $dom->createElement("marker");
    $newnode = $parnode->appendChild($node);
    $newnode->setAttribute("name", $row['name']);
    $newnode->setAttribute("address", $row['address']);
    $newnode->setAttribute("lat", $row['lat']);
    $newnode->setAttribute("lng", $row['lng']);
    $newnode->setAttribute("type", $row['type']);
}
echo $dom->saveXML();
?>
```


Έτσι, στον παραπάνω κώδικα αφού γίνεται εισαγωγή των χαρακτηριστικών της βάσης, ξεκινά η διαδικασία δημιουργίας ενός .xml (eXtensible Markup Language) dom. Το xml dom προσδιορίζει τα αντικείμενα και τις ιδιότητες όλων των στοιχείων του, καθώς και τους τρόπους όπου θα ανακτούνται αυτά.

Στις γραμμές κώδικα που ακολουθούν γίνεται το άνοιγμα της βάσης δεδομένων καθώς και ο ορισμός της σύνδεσης αυτής ως ενεργή. Στην ενεργή και συνδεδεμένη βάση δεδομένων κάνουμε ένα ερώτημα ώστε να διαβαστούν τα δεδομένα όλων των εγγράφων, προκειμένου στη συνέχεια να δημιουργηθούν nodes, δεδομένου ότι, σύμφωνα με το DOM, οτιδήποτε σε ένα XML αρχείο είναι node (document node, element node, text nodes, attribute node, comment nodes). Τέλος, το xml αυτό αρχείο αποθηκεύεται προκειμένου να μπορεί να διαβαστεί. Η ανάκτηση του αρχείου γίνεται από το προηγούμενο κομμάτι κώδικα που σχολιάστηκε μέσω της εντολής

```
var xml = new GXml.parse(data);
```

Κεφάλαιο 4

Μελλοντικές Επεκτάσεις

4.1 Ιστορικό κινήσεων

Μπορεί να προστεθεί διαδικασία κατά την οποία θα αποθηκεύονται και οι παλαιότερες θέσεις των ταξί και όχι μόνο η τρέχουσα θέση καθώς και παραπάνω στοιχεία όπως η ταχύτητα αυτών. Έτσι θα μπορούμε να έχουμε ιστορικό για τις θέσεις και τις καταστάσεις των ταξί και ο διαχειριστής θα μπορεί να προβάλλει περισσότερα στατιστικά δεδομένα. Όταν τα δεδομένα αυτά ξεπερνούν μια συγκεκριμένη διάρκεια περνούν σε ιστορικό υποσύστημα και διαγράφονται κατά βούληση.

Π.χ.(1) για ορισμένο χρονικό διάστημα και για ορισμένη περιοχή πόσες φορές βρέθηκε το ταξί σε συγκεκριμένη κατάσταση και ποια ήταν η μέση ταχύτητα που είχε.

Π.χ.(2) σε ποια περιοχή και πότε βρέθηκαν πολλά ταξί που είχαν συγκεκριμένη κατάσταση.

Π.χ. (3) Πόσες φορές σε ένα συγκεκριμένο χρονικό διάστημα κλήθηκαν ταξί με προορισμό συγκεκριμένη περιοχή.

4.2 Ιστορικό διαδρομών

Όπως προαναφέρεται, τα δεδομένα για τις καταχωρημένες διαδρομές διατηρούνται σε μία βάση δεδομένων. Σαν επέκταση θα μπορούσαμε να δημιουργήσουμε φίλτρα ώστε να μπορούμε να παίρνουμε συγκεκριμένα δεδομένα από τη βάση αυτή.

Π.χ.(1) πόσο διαρκεί κατά μέσο όρο η μετάβαση ενός ή περισσότερων ταξί από την τρέχουσα θέση τους, στη θέση της διεύθυνσης του χρήστη.

Π.χ.(2) σε ποιο χρονικό διάστημα παρουσιάστηκαν οι περισσότερες ή οι λιγότερες κλήσεις από τους πελάτες.

Π.χ.(3) πόσες κλήσεις έχει καταχωρήσει κάθε πελάτης.

4.3 Περιοχές εξαίρεσης / συναγερμού

Για την καλύτερη λειτουργία της προσομοίωσης της κίνησης των ταξί θα είναι δυνατό να καταχωρηθούν περιοχές από τις οποίες να μην διέρχονται ταξί.

Π.χ. περιοχή με έκτακτα καιρικά φαινόμενα. Επίσης, περιοχές στις οποίες μόλις διέρχεται ένα ταξί να προκαλείται συναγερμός με αποστολή e-mail στον διαχειριστή του συστήματος.

Π.χ. περιοχή στην οποία πραγματοποιείται πορεία διαμαρτυρίας και είναι αποκλεισμένη.

4.4 Περιορισμένη κίνηση των ταξί

Σημαντική βελτίωση στην εφαρμογή θα αποτελούσε η υλοποίηση περιορισμένης κίνησης σε τροχιές για τα ταξί, και όχι ελεύθερης κίνησης. Δηλαδή, θα καταχωρούνται οι συντεταγμένες των δρόμων του συστήματος στους οποίους μπορούν να κινούνται τα ταξί και η κίνησή τους θα μπορεί να γίνεται μονάχα σε αυτούς.

4.5 Διευθύνσεις πελάτη

Για καλύτερη εξυπηρέτηση του πελάτη να μπορεί να καταχωρεί περισσότερες από μία διευθύνσεις αναχώρησης στις οποίες υπάρχει ενδεχόμενο να καλέσει ταξί. Π.χ. διεύθυνση εργασίας. Επίσης να δοθεί η δυνατότητα να καταχωρεί διευθύνσεις οι οποίες θα αποτελούν πιθανό προορισμό μελλοντικά. Π.χ. διεύθυνση πλατείας Αριστοτέλους κ.ο.κ.

Επομένως, μετά την υλοποίηση των παραπάνω, κατά την κλήση ενός ταξί, ο πελάτης θα μπορεί να επιλέγει τις διευθύνσεις αναχώρησης και προορισμού από μία λίστα, γεγονός που διευκολύνει την κλήση ενός ταξί, και

μειώνει κατά πολύ τη συνολική διαδικασία κλήσης, αφού με μόνο μερικά «κλικ» θα ήταν σε θέση να καλέσει ταξί.

4.6 Κατάσταση κλήσεων πελάτη

Ο πελάτης να μπορεί να ενημερώνεται μέσω της ιστοσελίδας για την κατάσταση της κλήσης που έχει πραγματοποιήσει πριν από λίγο. Π.χ. σε πόσα λεπτά περίπου θα βρίσκεται το ταξί στην διεύθυνσή του.

Επίσης, εφόσον υπάρχει το ιστορικό διαδρομών (υποκεφάλαιο 7.3), να μπορεί να βλέπει στοιχεία από παλαιότερες κλήσεις του πελάτη καθώς και στατιστικά και να λαμβάνει έναν μέσο όρο χρόνου αναμονής για την άφιξη του ταξί.

4.7 Διάφορες εξειδικεύσεις

Η εφαρμογή αυτή μπορεί να τροποποιηθεί ώστε να αποκτήσει εξειδίκευση στη διαχείριση και άλλων κινούμενων αντικειμένων εκτός των ταξί.

Π.χ.(1) διαχείριση κινούμενων οχημάτων από μεταφορική εταιρεία (courier), η οποία ενδιαφέρεται για την εξέλιξη των μεταφορών της καθώς και στατιστικά στοιχεία για αυτές.

Π.χ.(2) παρακολούθηση της κίνησης των επίγειων οχημάτων που κινούνται σε δρόμους (αυτοκίνητα, φορτηγά, μηχανάκια), ώστε να παρουσιάζεται συνολική εικόνα της κίνησης αυτών με δυνατότητες προβολής στατιστικών δεδομένων και πρόβλεψης ενδεχόμενου μποτιλιαρίσματος.

Π.χ.(3) παρακολούθηση κινούμενων αεροπλάνων και πλοίων, ώστε να προβάλλονται στοιχεία για την κίνηση αυτών.

Π.χ.(4) παρακολούθηση της κίνησης των σύννεφων και των καιρικών φαινομένων, ώστε να δημιουργούνται στατιστικά στοιχεία και να προβλέπονται ορισμένα καιρικά φαινόμενα.

Τέτοια συστήματα ήδη εφαρμόζονται από εταιρίες και αποδεικνύονται πολύ σημαντικά.

Κεφάλαιο 5

Βιβλιογραφία

- ([http://en.wikipedia.org/wiki/Galileo %28satellite navigation%29](http://en.wikipedia.org/wiki/Galileo_%28satellite_navigation%29))
- <http://www.myjoomla.gr/content/section/4/83/>
- <http://help.joomla.org/>
- <http://www.joomlatutorials.com/>
- http://www.java2s.com/Tutorial/VB/0020_Language-Basics/Readkeyboardinput.htm
- http://en.wikipedia.org/wiki/List_of_CLI_languages
- http://en.wikipedia.org/wiki/Visual_Basic_.NET
- http://www.java2s.com/Tutorial/CSharp/0280_Development/Readinginputfromconsole.htm
- Google maps Σημειώσεις μαθήματος
- <http://www.codefrenzy.net/2008/03/15/code-for-gps-tracking-with-google-maps-ajax/>
- <http://www.phpeveryday.com/articles/AJAX-MySQL-Creating-XML-Data-use-DOM-P406.html>
- <http://code.google.com/apis/maps/articles/phpsqlajax.html>
- <http://www.hscripts.com/tutorials/javascript/clear-timeout.php>
- <http://www.templatezine.com/2009/06/how-to-customize-joomla-default-contact-form/>
- http://www.w3schools.com/php/php_ref_mail.asp
- <http://forum.joomla.org/viewtopic.php?f=231&t=213543>
- http://www.reference.joomlademo.de/nav.html?components/com_contact/controller.php.source.html
- http://www.experts-exchange.com/Programming/Languages/Scripting/JavaScript/Q_23901817.html