

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ
ΕΛΛΑΔΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**«ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΠΛΗΡΟΦΟΡΙΑΚΟΥ
ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΕΤΑΙΡΙΑ ΕΝΟΙΚΙΑΣΕΩΣ
ΑΥΤΟΚΙΝΗΤΩΝ»**

**ΜΙΧΕΛΗ ΕΥΘΥΜΙΑ 1237
ΔΟΥΒΗΣ ΑΝΤΩΝΗΣ 1182**

ΕΠΙΒΛΕΠΩΝ: ΦΑΛΙΑΓΚΑ ΕΥΑΝΘΙΑ, ΔΙΔΑΚΤΩΡ

ΑΝΤΙΡΡΙΟ

ΝΟΕΜΒΡΙΟΣ 2017

ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες.....	3
Περίληψη.....	3
Abstract.....	4
Κεφάλαιο 1 Εισαγωγή	5
1.1 Σκοπός της Πτυχιακής Εργασίας.....	5
1.2 Έννοια Πληροφοριακών Συστημάτων.....	5
1.3 Εξέλιξη Πληροφοριακών Συστημάτων	6
1.4 Ανάλυση και Σχεδιασμός Πληροφοριακών Συστημάτων.....	8
1.5 Κατηγορίες Πληροφοριακών Συστημάτων.....	8
Κεφάλαιο 2 Σκοπός και Στόχοι.....	10
2.1 Σχεδίαση πληροφοριακού συστήματος.....	10
2.1.1 Λίστα προδιαγραφών.....	10
2.1.2 UML Unified Modeling Language.....	11
2.1.3 Πιθανοί Χρήστες (Actors) του συστήματος.....	12
2.1.4 Συνολικό μοντέλο περιπτώσεων χρήσης.....	12
Περίπτωση χρήσης δημιουργία κράτησης από πελάτη.....	13
Περίπτωση χρήσης διαγραφή κράτησης από τον πελάτη.....	14
Περίπτωση χρήσης σύνδεση διαχειριστή στο σύστημα.....	15
Περίπτωση χρήσης διαχείριση αυτοκινήτων.....	16
Περίπτωση χρήσης κρατήσεων και μετατροπή σε συμβόλαιο..	17
2.2 Sql Structured Query Language.....	18
2.2.1 Αναπαράσταση διαγράμματος οντοτήτων - συσχετίσεων.....	18
2.2.2 Αναπαράσταση σχεσιακού μοντέλου.....	19
2.2.3 Οι πίνακες της βάσης δεδομένων.....	19

Κεφάλαιο 3 Γλώσσες Ανάπτυξης Συστημάτων Παγκοσμίου Ιστού.....	27
3.1 PHP.....	27
3.2 HTML.....	28
3.3 CSS.....	30
Κεφάλαιο 4 Χρήση Προγραμμάτων.....	31
4.1 Adobe Dreamweaver.....	31
4.2 Apache.....	34
4.3 Xampp.....	35
Κεφάλαιο 5 Αρχιτεκτονική του συστήματος.....	36
5.1 Η Αρχιτεκτονική του Συστήματος - Επίπεδα (Tiers) Λειτουργίας..	36
Κεφάλαιο 6 Επίλογος.....	42
Βιβλιογραφία.....	43
Παράρτημα 1 - Διεπαφές συστήματος.....	44
Παράρτημα 2 - Κώδικας βάσης δεδομένων.....	48
Παράρτημα 3 - Κώδικες (Html - Php - Css) διεπαφών πελάτη.....	57
Παράρτημα 4 - Κώδικες (Html - Php - Css) διεπαφών διαχειριστή.....	69

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε θερμά την επιβλέποντα καθηγήτρια κυρία Ευανθία Φαλιάγκα για τη καθοδήγηση της καθ' όλη τη διάρκεια της εκπόνησης αυτής της πτυχιακής.

Τους καθηγητές, καθώς και το προσωπικό της σχολής, για όλη τη βοήθεια και τις γνώσεις που μας προσέφεραν όλα αυτά τα χρόνια των σπουδών μας.

Και τέλος θα θέλαμε να ευχαριστήσουμε τους γονείς μας για την εμπιστοσύνη τους και τη στήριξη που μας έχουν προσφέρει σε όλες μας τις αποφάσεις μέχρι σήμερα.

ΠΕΡΙΛΗΨΗ

Τα πληροφοριακά συστήματα αποτελούν έναν σημαντικό παράγοντα στις μέρες μας, ώστε να επιτυγχάνεται η ανάλυση της πληροφορίας από το ανθρώπινο δυναμικό στο έπακρο. Η πλειοψηφία των επιχειρήσεων διαχειρίζεται πληροφοριακά συστήματα, που έχουν ως αποτέλεσμα καλύτερη οργάνωση και καλύτερα αποτελέσματα στην εργασία τους, ανάλογα τις ανάγκες που παρουσιάζονται

Στη συγκεκριμένη πτυχιακή εργασία, ασχολούμαστε με τη δημιουργία και την ανάπτυξη πληροφοριακού συστήματος στηριζόμενο σε βάση δεδομένων, καθώς και με τη δημιουργία ιστοσελίδας ενοικίασης αυτοκινήτων, με στόχο την παροχή όσο το δυνατόν περισσότερων στοιχείων που θα εξυπηρετούν τις ανάγκες των πελατών.

Επίσης, γίνεται αναλυτική περιγραφή της ανάπτυξης της βάσης δεδομένων, με γλώσσα προγραμματισμού SQL, ενώ παρακάτω συναντάμε τις γλώσσες ανάπτυξης συστημάτων παγκοσμίου ιστού, όπως την PHP, HTML και CSS με τις οποίες στήνουμε την ιστοσελίδα μας.

Τέλος, καταλυτικό ρόλο στην εκπόνηση αυτής της εργασίας κατέχει η χρήση ορισμένων προγραμμάτων όπως το Apache 2.4.17, το MySQL 5.7.9, το PHP 5.6.16, το Adobe Dreamweaver CS5, που χωρίς αυτά θα ήταν αδύνατον να φτάσουμε στα επιθυμητά αποτελέσματα και συμπεράσματα

ABSTRACT

Nowadays, information systems consist of an important factor, in order to achieve analysis of information by the human resource. The majority of companies manage information systems, as a result of better organization and results in their work, depending on the needs presented.

In this project, we deal with the creation and development of a database based on information system as well as the creation of a car rental website, aiming to provide as much information as possible to customer needs.

A detailed description of the database's development is provided with the use of the SQL language program while we see web development languages such as PHP, HTML and CSS which we use for setting up our web site.

Finally, an important thing of this project is the use of some programs such as Apache 2.4.17, MySQL 5.7.9, PHP 5.6.16, Adobe Dreamweaver CS5. Without those, it would be impossible to achieve the desired results and conclusions.

Κεφάλαιο 1 Εισαγωγή

Ο ανεκδιήγητος ρυθμός ανταλλαγής πληροφοριών, αλλά και η αναγκαιότητα αποθήκευσης αυτών των πληροφοριών, οδήγησε στην δημιουργία ενός συστήματος όπου θα καθιστούσε ικανό τον ομαλό συνδυασμό αυτών. Ένα τέτοιο σύστημα είναι το πληροφοριακό και στην έρευνά μας θα πραγματοποιήσουμε ακριβή ανάλυση της έννοιας αυτής.

Καθώς κάθε άνθρωπος έχει διαφορετικές ανάγκες, οι επιχειρήσεις άρχισαν να εξειδικεύονται με βάση αυτές. Άλλωστε, έγινε άμεσα αντιληπτό πως η συλλογή, η επεξεργασία και η αποθήκευση δεδομένων για κάθε περίπτωση, ήταν απαραίτητο γεγονός. Έτσι η ανάπτυξη ενός πληροφοριακού συστήματος έγινε κύριο μέλημα, με σκοπό την εξυπηρέτηση των αναγκών.

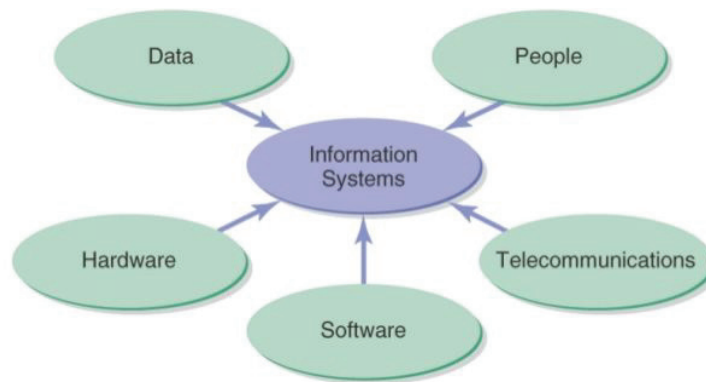
Παράλληλα, ο εκσυγχρονισμός της τεχνολογίας βοήθησε στην εξέλιξη της διαμόρφωσης ενός πληροφοριακού συστήματος.

1.1 Σκοπός της Πτυχιακής εργασίας

Ο σκοπός αυτής της πτυχιακής εργασίας είναι η μελέτη, ο σχεδιασμός και η ανάπτυξη ενός πληροφοριακού συστήματος για τη διαχείριση κρατήσεων για εταιρεία ενοικιάσεως αυτοκινήτων καθώς και η ιστοσελίδα στην οποία ο πελάτης μπορεί να κάνει μια κράτηση.

1.2 Έννοια Πληροφοριακών Συστημάτων

Με τον όρο πληροφοριακά συστήματα ονομάζουμε τις ενέργειες αυτές που ασκούνται είτε από τον άνθρωπο, είτε από ένα αυτοματοποιημένο υπολογιστικό σύστημα, οι οποίες λειτουργούν ως μέσον συλλογής, εγγραφής, ανάκτησης, επεξεργασίας, αποθήκευσης και ανάλυσης πληροφοριών. Τα συστήματα αυτά έχουν τη δυνατότητα να αποτελούνται από τρία μέρη όπως λογισμικό, υλικό και τηλεπικοινωνιακό



Εικόνα 1.1

Τα πληροφοριακά συστήματα αποτελούν το μέσο για την αρμονική συνεργασία ανθρώπινου δυναμικού, δεδομένων, διαδικασιών, και τεχνολογιών πληροφορίας και επικοινωνιών. Μάλιστα, η χρήση τους βοήθησε στην ύπαρξη μίας αλληλένδετης σχέσης μεταξύ της επιστήμης των υπολογιστών και των επιχειρήσεων.

Η επινόηση των πληροφοριακών συστημάτων είχε ως σκοπό την παροχή βοήθειας στην εκάστοτε επιχείρηση, καθώς με τη λειτουργία τους καθιστούν τη λήψη και τη διαχείριση των αποφάσεων πιο εύκολη.

Γενικότερα, με τον όρο πληροφοριακά συστήματα, αντιλαμβανόμαστε τον τρόπο με τον οποίο οι άνθρωποι χρησιμοποιούν την τεχνολογία της πληροφορίας και της επικοινωνίας, με στόχο την υποστήριξη των επιχειρηματικών διαδικασιών. Έτσι, η σχέση μεταξύ των συστημάτων διαχείρισης βάσης δεδομένων, αλλά και των συστημάτων δραστηριότητας είναι άμεση.

Επίσης, η αποθήκευση και η επεξεργασία των δεδομένων διαμορφώνουν το πληροφοριακό σύστημα ως μία μορφή επικοινωνίας του συστήματος με τον άνθρωπο, ενώ μπορεί να χαρακτηριστεί και ως ένα είδος ημιεπίσημης γλώσσας που ενισχύει τις ανθρώπινες αποφάσεις και ενέργειες.

1.3 Εξέλιξη Πληροφοριακών Συστημάτων

Από τα παλαιά χρόνια εμφανίστηκε μία έμφυτη προδιάθεση σχηματισμού πληροφοριακών συστημάτων, καθώς η ανάγκη μετάδοσης και επεξεργασίας της πληροφορίας ήταν βασικό μέλημα για τον άνθρωπο. Κάπως έτσι, μέσα από τα έτη παρατηρείται η δημιουργία και η εξέλιξη τους.

Από τα αρχαία κίβλας χρόνια, υπήρχαν ενδείξεις για την προσπάθεια μετάδοσης της πληροφορίας ανάμεσα στους ανθρώπους, όπως για παράδειγμα με τις φρυκτωρίες με

τις οποίες μεταδίδονταν μηνύματα είτε σε καιρό πολέμου, είτε σε καιρό ειρήνης. Με την πάροδο του χρόνου βέβαια, ο άνθρωπος συστηματοποίησε την μετάδοση των πληροφοριών ανακαλύπτοντας νέα μέσα όπως το χαρτί και εν συνεχεία το τηλέφωνο, μέχρι να φτάσουμε στη διάδοση των υπολογιστών στο ευρύ κοινό. Όλοι αυτοί οι παράγοντες βέβαια, έπαιξαν σπουδαίο ρόλο στη διαμόρφωση και στη βελτίωση των πληροφοριακών συστημάτων.



Εικόνα 1.2

Ξεκινώντας μία ιστορική αναδρομή από την περίοδο όπου η επιστήμη των υπολογιστών άνθισε, θα παρατηρήσουμε και την ανάπτυξη των πληροφοριακών συστημάτων. Πιο αναλυτικά:

Δεκαετία '70: Ξεκινάει η διάδοση των υπολογιστών, ειδικότερα στις επιχειρήσεις και στους οργανισμούς. Βασική γλώσσα προγραμματισμού ήταν η COBOL, ενώ γινόταν αποθήκευση όλων των δεδομένων στους δίσκους του υπολογιστή. Η χρήση των παραπάνω δεδομένων εφαρμόζονταν σε περιπτώσεις μισθοδοσίας, καταγραφής εμπορευμάτων, έκδοσης τιμολογίων κλπ

Δεκαετία '80: Αρχίζει η ανάπτυξη των προσωπικών υπολογιστών και οι πρώτες απόπειρες δημιουργίας τοπικών δικτύων. Οι προσωπικοί υπολογιστές εγκαθίστανται στις εταιρείες με αποτέλεσμα διάφορα τμήματα των εταιρειών να αποκτούν μια ανεξαρτησία από τα δεδομένα του κεντρικού υπολογιστή. Υπάρχει ραγδαία ανάπτυξη προγραμμάτων επεξεργασίας κειμένου και λογιστικών φύλλων.

Δεκαετία '90: Παρατηρείται περαιτέρω ανάπτυξη των δικτύων και ανάγκη τεχνικής υποστήριξης αυτών, ενώ ξεκινάει η δημιουργία εκτεταμένων δικτύων σε εταιρείες. Ο κύριος σκοπός των πληροφοριακών συστημάτων, καθίσταται η αυτοματοποίηση των υφισταμένων διαδικασιών επεξεργασίας δεδομένων και η δημιουργία ολοκληρωμένων πληροφοριακών συστημάτων.

Δεκαετία '00: Τη δεκαετία αυτή, το κύριο μέλημα είναι η κεντρική διαχείριση των πληροφοριακών συστημάτων και η καλή γνώση των επιμέρους τμημάτων του πληροφοριακού συστήματος. Τα εκτεταμένα δίκτυα επεκτείνονται ακόμη περισσότερο με τη χρήση του Internet, έτσι ώστε να περιλαμβάνονται στα πληροφοριακά συστήματα παγκόσμιες επιχειρήσεις και επιχειρησιακοί εταίροι πχ. αλυσίδες τροφοδοσίας και διανομής.

Σύγχρονη εποχή: Τα σύγχρονα πληροφοριακά συστήματα αναπτύσσονται με κύριο μέλημα την αποδοτικότητα και την ταχύτητα στην καταγραφή εμπορευμάτων, στην κατασκευή, στη διανομή, στην αλληλεπίδραση των συστημάτων, στην υποστήριξη λήψης αποφάσεων κλπ.

1.4 Ανάλυση και Σχεδιασμός Πληροφοριακών Συστημάτων

Οι αναλυτές συστημάτων υποστηρίζουν πως η ανάλυση και η σχεδίαση ενός συστήματος εξαρτάται από κάποιους παράγοντες. Για παράδειγμα η κατανόηση της οργάνωσης του σκοπού, της δομής και των διεργασιών ενός πληροφοριακού συστήματος, αλλά και η αναγκαιότητα της τεχνικής κατάρτισης, ώστε να είναι κάποιος σε θέση να εκμεταλλευτεί τα οφέλη της τεχνολογία της πληροφορίας, κατέχουν σημαντικό ρόλο.

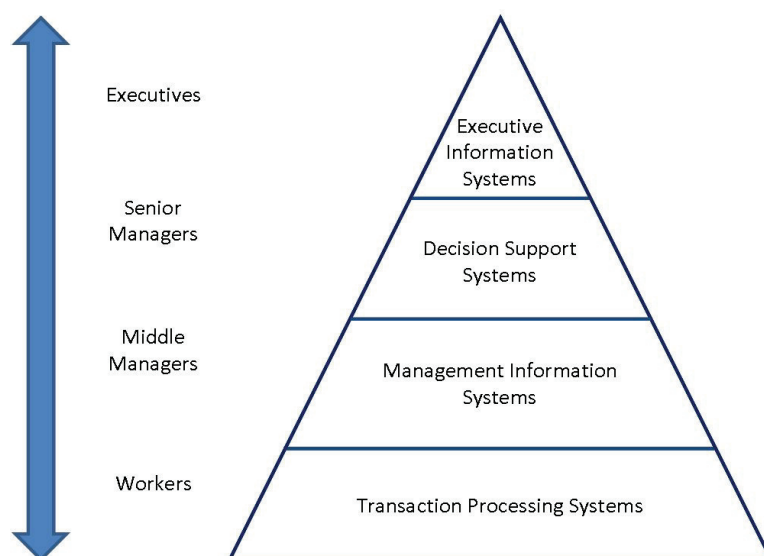
Ο κύριος στόχος για τη βελτίωση του σχεδιασμού ενός πληροφοριακού συστήματος, ώστε να δημιουργηθεί όσο το δυνατόν πιο αποτελεσματικά είναι η απόκτηση και η ανάπτυξη υπολογιστικών προγραμμάτων, πάνω στα οποία οι εργαζόμενοι πρέπει να εξοικειώνονται για να είναι σε θέση να τα χρησιμοποιούν. Τέτοια προγράμματα είναι αυτά που υποστηρίζουν οργανωτικές λειτουργίες ή διεργασίες.

Έτσι, ένα σύστημα μπορεί να μετατρέψει τα δεδομένα σε πληροφορία εφόσον περιλαμβάνει λογισμικά, υλικά συστήματα, εκπαιδευτικό υλικό για την τεχνική κατάρτιση των εργαζομένων, κατανομή ρόλων ώστε να γίνεται σύνδεση με το σύστημα, έλεγχος για την πρόληψη κλοπών, ενώ επίσης βασικός παράγοντας είναι οι υπάλληλοι να είναι γνώστες των προγραμμάτων ώστε να εκτελείται σωστά η εργασία.

1.5 Κατηγορίες Πληροφοριακών Συστημάτων

Τα πληροφοριακά συστήματα διαχωρίζονται σε διάφορες κατηγορίες ανάλογα τους τύπους που τα χαρακτηρίζουν. Βιβλία της δεκαετίας του 1980, αναφέρουν την ιεραρχία της οργάνωσης των πληροφοριακών συστημάτων μέσω μίας πυραμίδας. Στο κάτω μέρος της πυραμίδας καταγράφονται τα συστήματα επεξεργασίας συναλλαγών, ενώ στη συνέχεια αναφέρονται τα συστήματα διαχείρισης

πληροφοριών. Σε υψηλότερη θέση βρίσκονται τα συστήματα υποστήριξης αποφάσεων, ενώ τέλος στην κορυφή εδραιώνονται τα συστήματα υποστήριξης διοίκησης.



Εικόνα 1.3

Παρόλο που το μοντέλο της πυραμίδας παραμένει χρήσιμο, η εμφάνιση νέων τεχνολογιών ανέπτυξαν νέες κατηγορίες πληροφοριακών συστημάτων, μερικές από τις οποίες δεν ταιριάζουν με το μοντέλο της πυραμίδας. Παραδείγματα τέτοιων περιπτώσεων είναι:

- Συστήματα αποθήκευσης δεδομένων
- Συστήματα προγραμματισμού παραγωγής και υλικών
- Συστήματα επιχειρήσεων
- Έμπειρα συστήματα
- Συστήματα μηχανών αναζήτησης
- Γεωγραφικά συστήματα πληροφοριών
- Παγκόσμια συστήματα πληροφοριών
- Συστήματα αυτοματισμού γραφείο

Ένα υπολογιστικό πληροφοριακό σύστημα χρησιμοποιεί την τεχνολογία των υπολογιστών με σκοπό να εκτελούνται ορισμένες προγραμματισμένες εργασίες που απαιτούνται. Τα πληροφοριακά συστήματα που βασίζονται σε υπολογιστή, αποτελούνται από κάποια μέρη εκ των οποίων τα βασικότερα είναι:

- **Hardware:** είναι συσκευές όπως η οθόνη, ο επεξεργαστής, ο εκτυπωτής, το πληκτρολόγιο, όλα αυτά δουλεύουν μαζί ώστε να δεχθούν, να επεξεργαστούν και να παρουσιάσουν τα δεδομένα και τις πληροφορίες.
- **Software:** είναι τα προγράμματα που επιτρέπουν στο hardware

να επεξεργαστεί την πληροφορία.

- Βάσεις δεδομένων: είναι η συλλογή αρχείων ή πινάκων που περιέχουν τα ανάλογα δεδομένα.
- Δίκτυα: είναι σύστημα σύνδεσης που επιτρέπει πολλούς διαφορετικούς υπολογιστές να διανέμουν τους πόρους.
- Διαδικασίες: είναι οι εντολές για τον συνδυασμό όλων των παραπάνω στοιχείων, έτσι ώστε να γίνεται επεξεργασία της πληροφορίας και να λαμβάνουμε το τελικό αποτέλεσμα.

Κεφάλαιο 2 Σκοπός και Στόχοι

2.1 Σχεδίαση πληροφοριακού συστήματος

2.1.1 Λίστα προδιαγραφών

ΔΥΝΑΤΟΤΗΤΕΣ ΠΕΛΑΤΗ

1.Ο πελάτης έχει τη δυνατότητα να κάνει κράτηση για ενοικίαση ενός αυτοκινήτου.

α).Για να κάνει τη κράτηση δίνει τα παρακάτω στοιχεία:

- Όνομα και Επώνυμο
- Email
- Τηλέφωνο

β) Για τη κράτηση επιλέγει τα παρακάτω:

- Ημερομηνία και ώρα παραλαβής αυτοκινήτου
- Μέρος παραλαβής αυτοκινήτου
- Κατηγορία αυτοκινήτου (αυτόματα, μικρό, τζίπ κτλ.)
- Ημερομηνία και ώρα παράδοσης αυτοκινήτου
- Μέρος παράδοσης αυτοκινήτου

2.Ο πελάτης έχει τη δυνατότητα να ακυρώσει τη κράτηση που έκανε.

ΔΥΝΑΤΟΤΗΤΕΣ ΔΙΑΧΕΙΡΗΣΤΗ

1.Ο διαχειριστής δέχεται τα αιτήματα για τις κρατήσεις.

2.Αναθέτει κάποιο αυτοκίνητο έτσι ώστε να έχει γνώση για το πόσα αυτοκίνητα έχει ακόμα ελεύθερα.

(Στη πορεία μπορεί να αλλάξει το αυτοκίνητο που έχει αναθέσει αν π.χ το αυτοκίνητο έχει κάποιο τρακάρισμα ή πρόβλημα και πρέπει να πάει στο συνεργείο)

3.Ο διαχειριστής περνάει τα στοιχεία των αιτήματα που έχει λάβει στη βάση με όλες τις λεπτομέρειες

4.Ο διαχειριστής στέλνει mail στους πελάτες για να τους ενημερώση//επιβεβαίωσης ότι έλαβε τη κράτηση τους μαζί με τα στοιχεία της κρατήσεις (αυτόματα)

5.Ο διαχειριστής μπορεί να αλλάζει τη κατάσταση των αυτοκινήτων στις εξής:

- Ελεύθερο
- Νοικιασμένο
- Στο συνεργείο

Επίσης μπορεί ανανεώνει το στόλο προσθέτοντας ή αφαιρώντας αυτοκίνητα

6.Ο διαχειριστής μπορεί να επεξεργαστεί τα στοιχεία των περασμένων κρατήσεων αν ζητηθεί κάποια αλλαγή από κάποιο πελάτη ή σε περίπτωση ακύρωσης.

7.Ο διαχειριστής μπορεί να μετατρέψει τη κράτηση σε συμβόλαιο όταν έρθει ο πελάτης να παραλάβει το συμβόλαιο και να κλείσει το συμβόλαιο κατά την επιστροφή του αυτοκινήτου.

2.1.2 UML Unified Modeling Language

Η Uml είναι μια γλώσσα μοντελοποίησης που σκοπός της είναι να μας βοηθήσει με την γραφική απεικόνιση και τη κατασκευή του λογισμικού συστήματος που θέλουμε να δημιουργήσουμε

Η γλώσσα Uml χρησιμοποιεί διαγράμματα, τα οποία χωρίζονται σε 2 κατηγορίες σε αυτά που περιγράφουν τη στατική δομή του συστήματος και αυτά που περιγράφουν τη δυναμική δομή του συστήματος είναι τα εξής:

- Διάγραμμα Κλάσεων
- Διάγραμμα Περιπτώσεων Χρήσης
- Διάγραμμα Ακολουθίας
- Διάγραμμα Συνεργασίας
- Διάγραμμα Δραστηριότητας
- Διάγραμμα Καταστάσεων
- Διάγραμμα Συστατικών
- Διάγραμμα Ανάπτυξης

Στη συγκεκριμένη εργασία θα χρησιμοποιήσουμε τα διαγράμματα περίπτωσης χρήσης τα οποία περιγράφουν όλες τις δυνατότητες που κάθε δίνει το σύστημα στο

χρήστη καθώς και τα διαγράμματα δραστηριοτήτων που

Η γλώσσα Uml δημιουργήθηκε από τους Grady Booch, Ivar Jacobson και James Rumbaugh το 1994.

Μέχρι σήμερα έχουν βγει οι εξής εκδόσεις : Uml 1.x και Uml 2.x

2.1.3 Πιθανοί Χρήστες(Actors) του Συστήματος

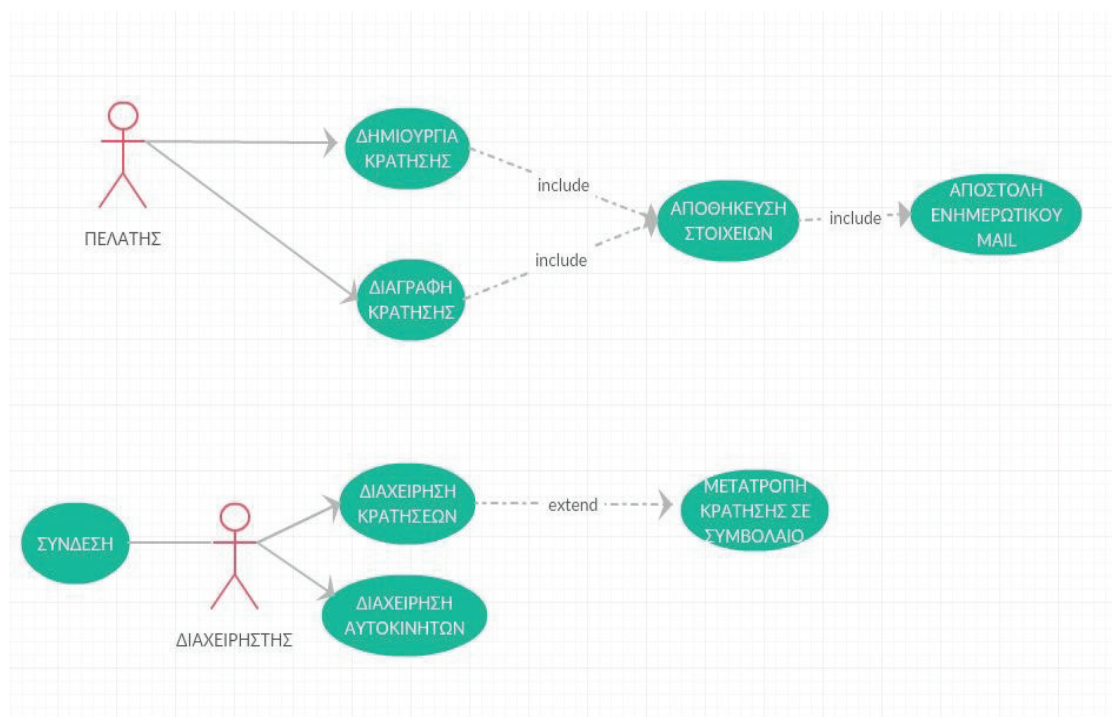
Κάθε χρήστης αντιπροσωπεύει έναν ρόλο ή μια οντότητα που μπορεί να βρίσκεται εκτός από το σύστημα μας αλλά να αλληλεπιδρά με αυτό π.χ ένα άλλο σύστημα η μια βάση δεδομένων.

Στη δική μας περίπτωση οι χρήστες του συστήματος είναι 2:

- Ο πελάτης : Ο οποίος είναι ο χρήστης που μπαίνει στο internet και κάνει τη κράτηση
- Ο διαχειριστής: Ο οποίος είναι ο χρήστης που διαχειρίζεται το σύστημα.

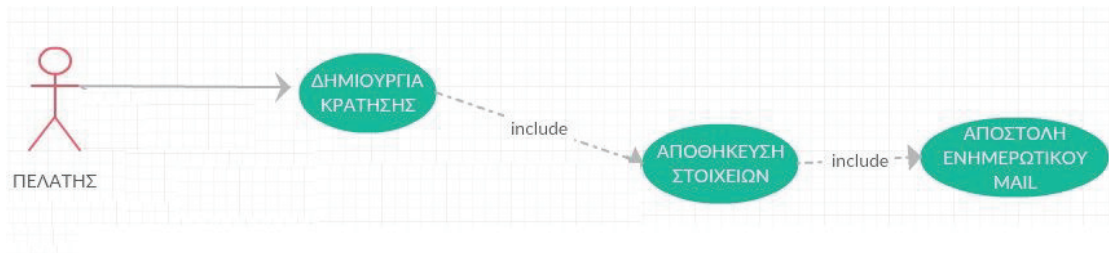
2.1.4 Συνολικό μοντέλο περιπτώσεων χρήσης

Σε αυτή την ενότητα παρουσιάζεται το συνολικό μοντέλο χρήσης. Στο παρακάτω διάγραμμα παρουσιάζονται όλες οι λειτουργίες που μπορεί να πραγματοποιήσει ο κάθε χρήστης.



Διάγραμμα 2.1

Περίπτωση χρήσης δημιουργία κράτησης από τον πελάτη

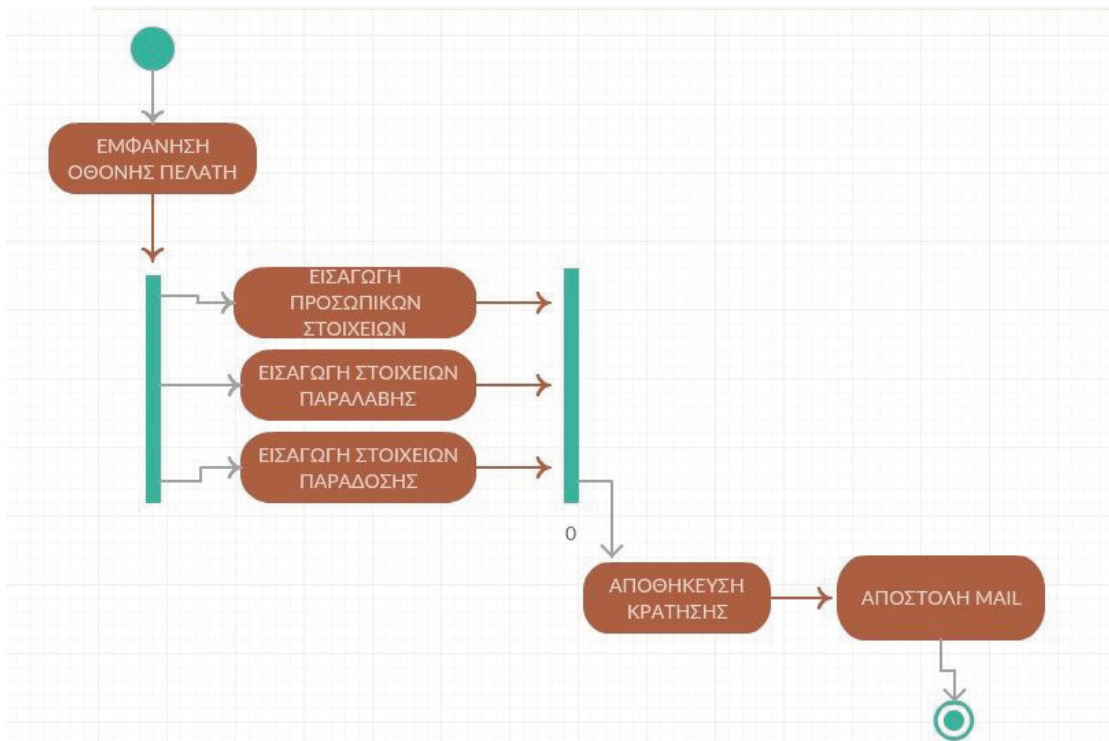


Διάγραμμα 2.2

Βασική Ροή

- Ο χρήστης μπαίνει στην ιστοσελίδα
- Ο χρήστης δίνει τα παρακάτω στοιχεία:
 - όνομα, επώνυμο , τηλέφωνο, mail
 - Ημερομηνία παραλαβής, ώρα και τόπος
 - Ημερομηνία παράδοσης, ώρα και τόπος
 - Κατηγορία αυτοκινήτου
- Ο χρήστης επιβεβαιώνει τη κράτηση πατώντας το κουμπί “Submit”
- Ο χρήστης λαμβάνει ένα mail με τον κωδικό της κράτησης του.

Διάγραμμα Δραστηριοτήτων



Διάγραμμα 2.3

Περίπτωση χρήσης διαγραφή κράτησης από τον πελάτη



Διάγραμμα 2.4

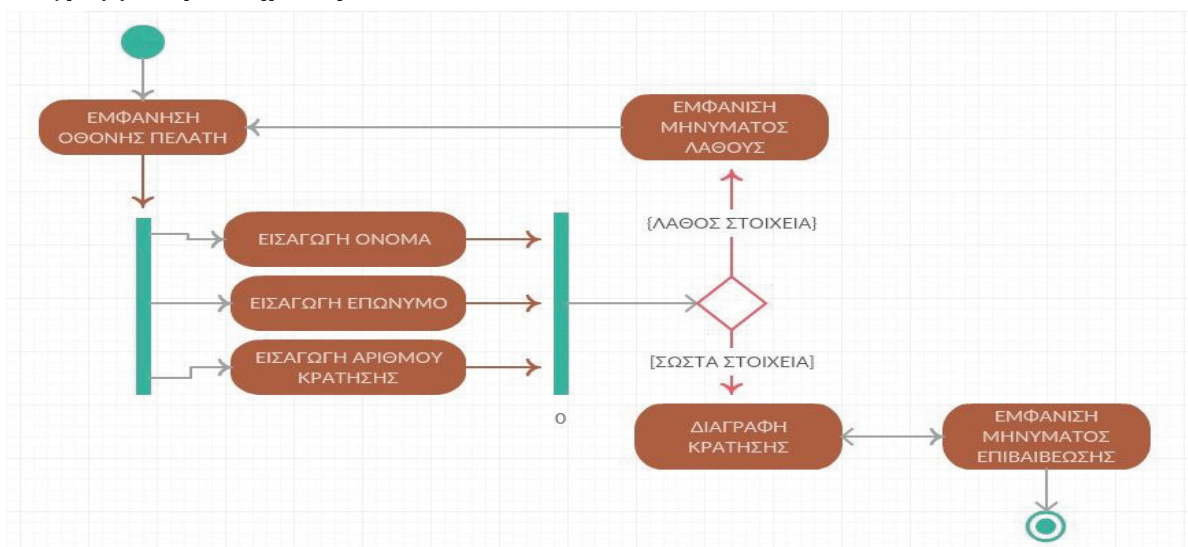
Βασική Ροή

- Ο χρήστης μπαίνει στην ιστοσελίδα
- Ο χρήστης δίνει τα παρακάτω στοιχεία:
 - Όνομα
 - Επώνυμο
 - Αριθμό κράτησης
- Ο χρήστης επιβεβαιώνει τη διαγραφή της κράτησης πατώντας το κουμπί “Submit”
- Η κράτηση διαγράφεται
- Εμφάνιση μηνύματος επιβεβαίωσης

Εναλλακτική Ροή

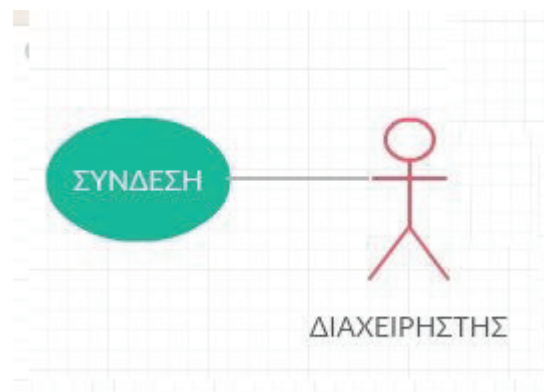
- Τα στοιχεία που έχει δώσει ο χρήστης είναι λάθος
- Εμφάνιση μηνύματος λάθους

Διάγραμμα Δραστηριοτήτων



Διάγραμμα 2.5

Περίπτωση χρήσης σύνδεση διαχειριστή στο σύστημα



Διάγραμμα 2.5

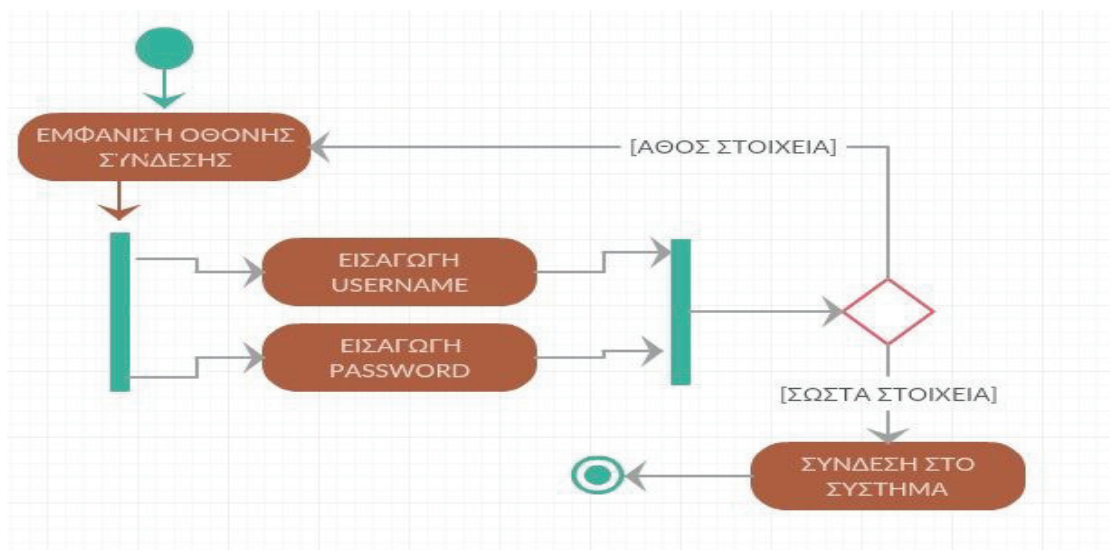
Βασική Ροή

- Εμφανίζεται η οθόνη σύνδεσης
- Ο διαχειριστής εισάγει
 - Username
 - Password
- Ο διαχειριστής συνδέεται στο σύστημα

Εναλλακτική Ροή

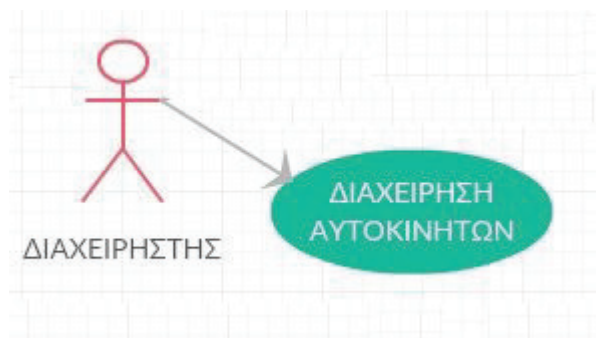
- Τα στοιχεία που έχει δώσει ο διαχειριστής είναι λάθος
- Εμφανίζεται η οθόνη σύνδεσης

Διάγραμμα Δραστηριοτήτων



Διάγραμμα 2.6

Περίπτωση χρήσης Διαχείριση αυτοκινήτων



Διάγραμμα 2.7

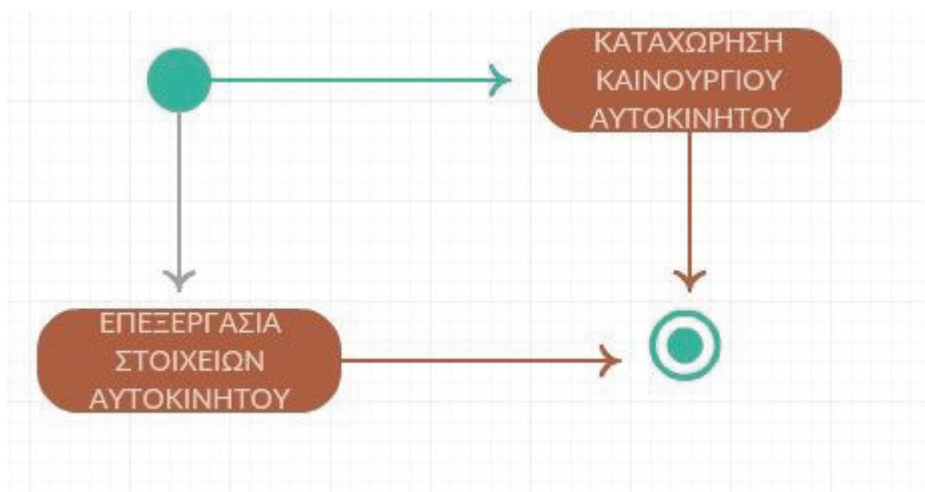
Βασική Ροή 1

- Ο διαχειριστής προσθέτει ένα καινούργιο αυτοκίνητο στο στόλο της εταιρείας δίνοντας τα κατάλληλα στοιχεία

Βασική Ροή 2

- Ο διαχειριστής αλλάζει κάποια στοιχεία ενός ήδη υπάρχων αυτοκινήτου του στόλου της εταιρείας

Διάγραμμα Δραστηριοτήτων



Διάγραμμα 2.8

Περίπτωση χρήσης διαχείριση κρατήσεων και μετατροπή σε συμβόλαιο



Διάγραμμα 2.9

Βασική Ροή 1

- Ο διαχειριστής δημιουργεί μια καινούργια κράτηση στο σύστημα δίνοντας τα κατάλληλα στοιχεία

Βασική Ροή 2

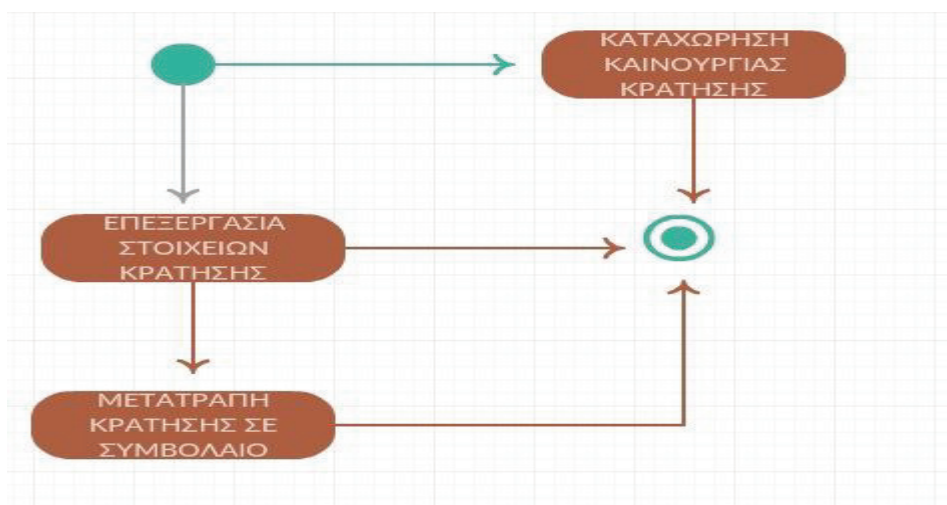
- Ο διαχειριστής αλλάζει κάποια στοιχεία σε κάποια ήσ

δη υπάρχουσα κράτηση.

Βασική Ροή 3

- Ο διαχειριστής μετατρέπει τη κράτηση σε συμβόλαιο όταν έρθει η ώρα της παράδοσης του αυτοκινήτου στο πελάτη.

Διάγραμμα Δραστηριοτήτων



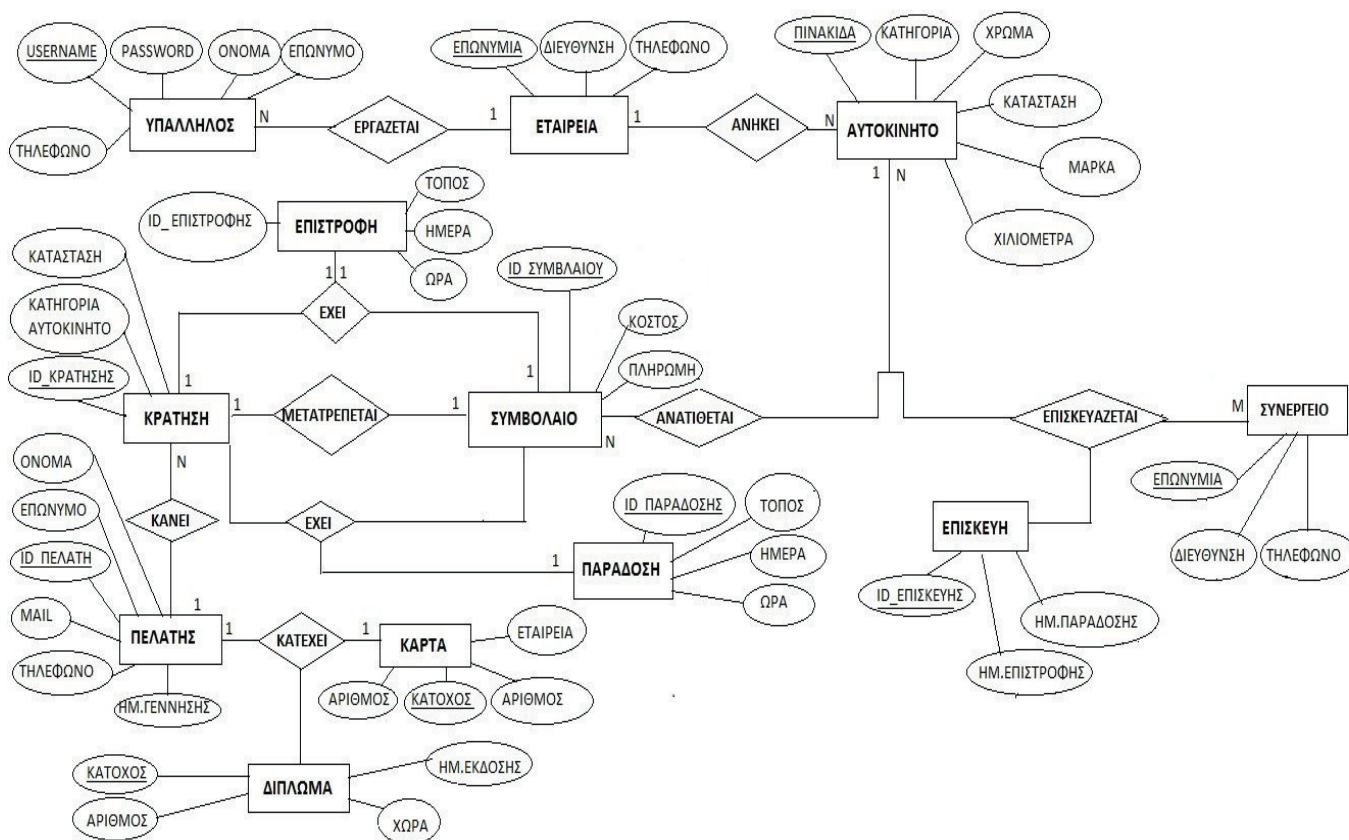
Διάγραμμα 2.10

2.2 Sql Structured Query Language.

Η Sql είναι μια γλώσσα που χρησιμοποιείται για τη δημιουργία βάσεων δεδομένων και δίνει τη δυνατότητα στο προγραμματιστή όχι μόνο να δημιουργήσει μια βάση δεδομένων αλλά και να τη τροποποιήσει να ενημερώσει τα δεδομένα της και να ελέγξει το ποιος θα μπορεί να έχει πρόσβαση σε αυτά

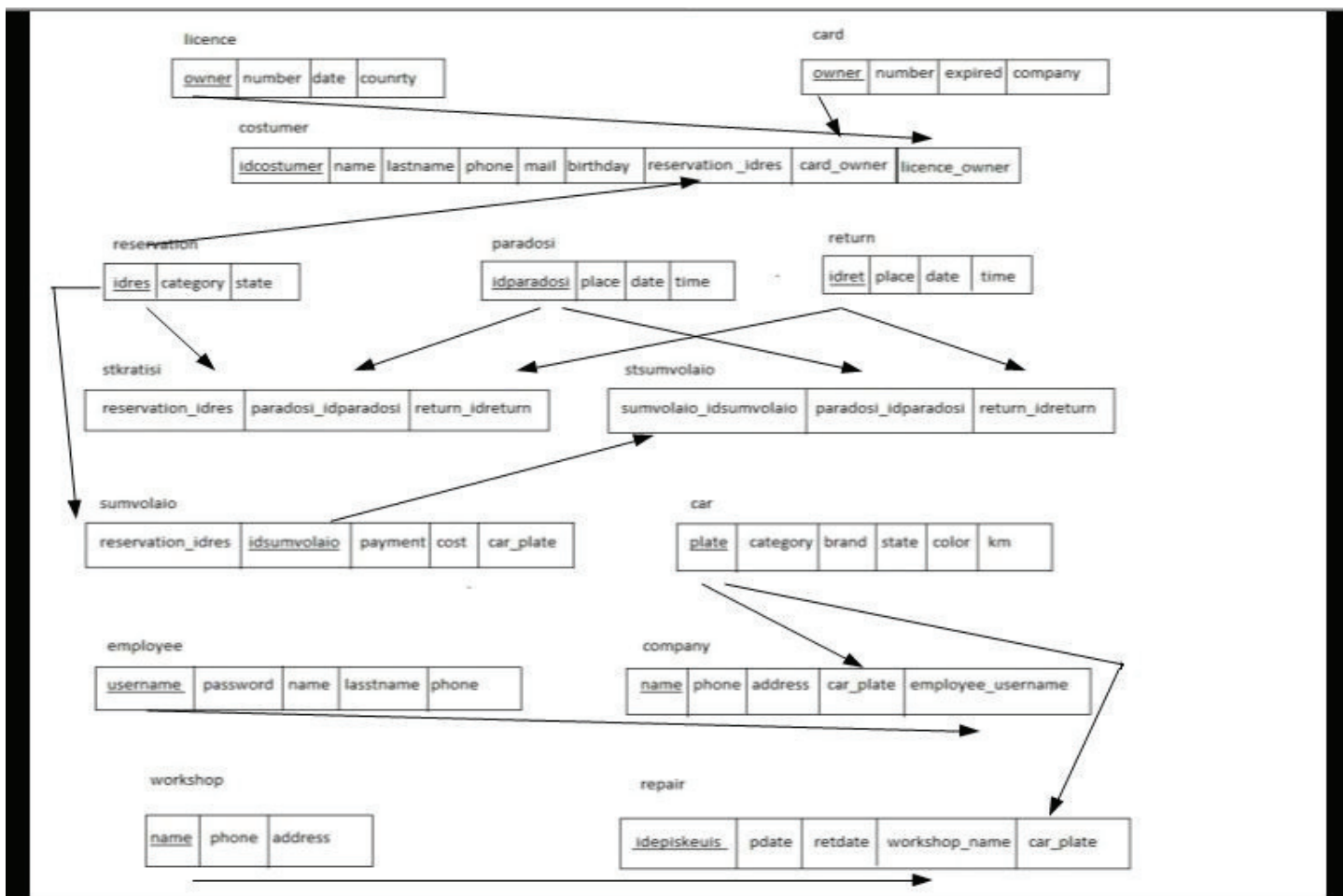
Η Sql δημιουργήθηκε τη δεκαετία του 1970 από τους Andrew Richardson, Donald C. Messely και τον Raymond F.Boynce

2.2.1 Αναπαράσταση διαγράμματος Οντοτήτων - Συσχετίσεων



Διάγραμμα 2.11

2.2.2 Αναπαράσταση Σχεσιακού μοντέλου



Διάγραμμα 2.12

2.2.3 Οι πίνακες της Βάσης Δεδομένων

Με βάση το διάγραμμα οντοτήτων - συσχέτισης και του σχεσιακού μοντέλου παρακάτω παρουσιάζονται οι πίνακες της βάσης μας.

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΟ ΑΥΤΟΚΙΝΗΤΟ

Όνομα πίνακα: car

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
Plate	VARCHAR (45)	OXI	Πινακίδα	
Category	VARCHAR (45)	OXI	Κατηγορία	
brand	VARCHAR (45)	OXI	Μάρκα	
state	VARCHAR (45)	NAI	Κατάσταση(Ελεύθερο,νοικιασμένο)	
colour	VARCHAR (45)	NAI	Χρώμα	
Km	INT	NAI	Πόσα χιλιόμετρα έχει κάνει	

Πίνακας 2.1

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΟΝ ΠΕΛΑΤΗ

Όνομα πίνακα: costumer

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
idcostomer	INT/ AUTO_INCREMENT	OXI	Id πελάτη	
Name	VARCHAR (45)	OXI	όνομα	
Lastname	VARCHAR (45)	OXI	επώνυμο	
phone	VARCHAR (45)	OXI	τηλέφωνο	
mail	VARCHAR (45)	OXI	mail	
birthday	DATE	NAI	γενέθλια	
card_owner	VARCHAR (45)	NAI	Ένωση με πιστωτική κάρτα	Πίνακα card Πεδίο Owner

reservation_idres	INT	ΝΑΙ	Ένωση με Id κράτησης	Πίνακα reservation Πεδίο idres
licence_owner	VARCHAR (45)	ΝΑΙ	Ένωση με δίπλωμα οδήγησης	Πίνακα licens Πεδίο owner

Πίνακας 2.2

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΟ ΔΙΠΛΩΜΑ ΟΔΗΓΗΣΗΣ

Όνομα πίνακα: licence

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
owner	VARCHAR (45)	ΟΧΙ	Όνομα ιδιοκτήτη	
number	VARCHAR (45)	ΟΧΙ	Αριθμός διπλώματος	
date	DATE	ΟΧΙ	Ημερομηνία έκδοσης	
country	VARCHAR (45)	ΟΧΙ	Χώρα έκδοσης	

Πίνακας 2.3

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΗ ΠΙΣΤΩΤΙΚΗ ΚΑΡΤΑ

Όνομα πίνακα: card

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
Owner	VARCHAR (45)	ΟΧΙ	Όνομα ιδιοκτήτη	
number	VARCHAR (45)	ΟΧΙ	Αριθμός	
expired	DATE	ΟΧΙ	Ημερομηνία λήξης	

company	VARCHAR (45)	OXI	Τύπος κάρτας(VISA, AMEX, κτλ)	
---------	-----------------	-----	-------------------------------------	--

Πίνακας 2.4

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΗΝ ΕΤΑΙΡΕΙΑ

Όνομα πίνακα: company

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
name	VARCHAR (45)	OXI	Επωνυμία	
phone	VARCHAR (45)	OXI	Τηλέφωνο	
address	VARCHAR (45)	OXI	Διεύθυνση	
employee_username	VARCHAR (45)	NAI	Σύνδεση με τα στοιχεία του υπαλλήλου	Πίνακας employee Πεδίο username
car_plate	VARCHAR (45)	NAI	Σύνδεση με αυτοκίνητο που ανήκει στην εταιρεία	Πίνακας car Πεδίο plate

Πίνακας 2.5

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΟΝ ΥΠΑΛΛΗΛΟ

Όνομα πίνακα: employee

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
username	VARCHAR (45)	OXI	Όνομα χρήστη που χρησιμοποιεί ο υπάλληλος για να	

			συνδεθεί	
password	VARCHAR (45)	OXI	Κωδικό που χρησιμοποιεί ο υπάλληλος για να συνδεθεί	
name	VARCHAR (45)	OXI	Όνομα	
lastname	VARCHAR (45)	OXI	Επώνυμο	
phone	VARCHAR (45)	OXI	Τηλέφωνο	

Πίνακας 2.6

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΟ ΣΥΝΕΡΓΕΙΟ

Όνομα πίνακα: workshop

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝ Ο	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
name	VARCHAR (45)	OXI	Όνομα	
phone	VARCHAR (45)	OXI	Τηλέφωνο	
Address	VARCHAR (45)	OXI	Διεύθυνση	

Πίνακας 2.7

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΕΠΙΣΚΕΥΗΣ ΤΟΥ ΑΥΤΟΚΙΝΗΤΟΥ

Όνομα πίνακα: repair

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
idepiskeuis	INT/ AUTO_INCREMENT	OXI	Id επισκευής	

pdate	DATE	NAI	Ημέρα που πήγε το αυτοκίνητο στο συνεργείο	
retdate	DATE	NAI	Ημέρα που το αυτοκίνητο επισκευάστηκε	
workshop_name	VARCHAR (45)	NAI	Σύνδεση με το συνεργείο	Πίνακας workshop Πεδίο name
car_plate	VARCHAR (45)	NAI	Σύνδεση με πινακίδα αυτοκινήτου που είναι στο συνεργείο	Πίνακας car Πεδίο plate

Πίνακας 2.8

● ΠΙΝΑΚΑΣ ΓΙΑ ΤΟ ΣΥΜΒΟΛΑΙΟ

Όνομα πίνακα: sumvolaio

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
idsumvolaio	INT/ AUTO_INCREMENT	OXI	Id συμβολαίου	
payment	VARCHAR (45)	NAI	πληρωμή	
cost	INT	OXI	κόστος	
car_plate	INT	OXI	Σύνδεση με αυτοκίνητο που ανατέθηκε	Πίνακας car Πεδίο plate
reservation_idres	INT	OXI	Σύνδεση με αρχική κράτηση	Πίνακας reservation Πεδίο idres

Πίνακας 2.9

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΗ ΚΡΑΤΗΣΗ

Όνομα πίνακα: reservation

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
idres	INT/ AUTO_INCREMENT	OXI	Id κράτησης	
category	VARCHAR (45)	OXI	Κατηγορία αυτοκινήτου	
state	VARCHAR (45)	OXI	Κατάσταση κράτησης(π.χ αναμονή,ακύρωση)	

Πίνακας 2.10

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΠΑΡΑΔΟΣΗΣ

Όνομα πίνακα: paradosi

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
idparadosi	INT/ AUTO_INCREMENT	OXI	Id παράδοσης	
place	VARCHAR (45)	OXI	Τόπος παράδοσης	
date	DATE	OXI	Ημέρα παράδοσης	
time	TIME	OXI	Ωρα παράδοσης	

Πίνακας 2.11

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΕΠΙΣΤΡΟΦΗΣ

Όνομα πίνακα: return

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
Idret	INT/ AUTO_INCREMENT	ΟΧΙ	Id επιστροφής	
place	VARCHAR (45)	ΟΧΙ	Τόπος επιστροφής	
date	DATE	ΟΧΙ	Ημέρα επιστροφής	
time	TIME	ΟΧΙ	Ώρα επιστροφής	

Πίνακας 2.12

- ΠΙΝΑΚΑΣ ΓΙΑ ΤΗΝ ΕΝΩΣΗ ΤΗΣ ΚΡΑΤΗΣΗΣ ΜΕ ΤΑ ΣΤΟΙΧΕΙΑ ΕΠΙΣΤΡΟΦΗΣ ΚΑΙ ΠΑΡΑΔΟΣΗΣ

Όνομα πίνακα: stkratisi

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
paradosi_idparadosi	INT	ΟΧΙ	Ένωση με id παράδοσης	Πίνακας paradosi Πεδίο idparadosi
return_idret	INT	ΟΧΙ	Ένωση με id επιστροφής	Πίνακας return Πεδίο idreturn
reservation_idres	INT	ΟΧΙ	Ένωση με id κράτησης	Πίνακας reservation Πεδίο idres

Πίνακας 2.13

- **ΠΙΝΑΚΑΣ ΓΙΑ ΤΗΝ ΕΝΩΣΗ ΤΟΥ ΣΥΜΒΟΛΑΙΟΥ ΜΕ ΤΑ ΣΤΟΙΧΕΙΑ ΕΠΙΣΤΡΟΦΗΣ ΚΑΙ ΠΑΡΑΔΟΣΗΣ**

Όνομα πίνακα: stsumvolaio

ΠΕΔΙΟ	ΤΥΠΟΣ	ΚΕΝΟ	ΣΧΟΛΙΟ	ΣΥΝΔΕΣΗ
sumvolaio_idsumvolaio	INT	OXI	Ένωση με id κράτησης	Πίνακας sumvolaio Πεδίο idsumvolaio
paradosi_idparadosi	INT	OXI	Ένωση με id παράδοσης	Πίνακας paradosi Πεδίο idparadosi
return_idreturn	INT	OXI	Ένωση με id επιστροφής	Πίνακας return Πεδίο idreturn

Πίνακας 2.14

Με αυτό το χρώμα συμβολίζονται τα πεδία που είναι primary key

Με αυτό το χρώμα συμβολίζονται τα πεδία που ενώνουν 2 πίνακες μεταξύ τους

Πίνακας 2.15

Κεφάλαιο 3 Γλώσσες Ανάπτυξης Συστημάτων Παγκοσμίου Ιστού

3.1 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για διακομιστές που έχει σχεδιαστεί κυρίως για την ανάπτυξη ιστού, αλλά χρησιμοποιείται επίσης ως γενική γλώσσα προγραμματισμού. Δημιουργήθηκε από τον Rasmus Lerdorf το 1994, ενώ τα αρχικά της σημαίνουν Hypertext Preprocessor.

Ο κώδικας της PHP μπορεί να ενσωματωθεί σε HTML ή HTML5 ή μπορεί να χρησιμοποιηθεί σε συνδυασμό με διάφορα συστήματα πρότυπων ιστού, συστήματα

διαχείρισης περιεχομένου ιστού και πλαίσια ιστού. Ο κώδικας PHP επεξεργάζεται συνήθως από έναν μεταφραστή PHP που υλοποιείται ως ενότητα στον διακομιστή ιστού ή ως μία κοινή εκτελέσιμη πύλη διεπαφής. Το λογισμικό του διακομιστή ιστού, συνδυάζει τα αποτελέσματα του μεταφρασμένου και του εκτελεσμένου κώδικα PHP, που μπορεί να είναι οποιοσδήποτε τύπος δεδομένων, περιλαμβανομένων εικόνων, με την παραγόμενη ιστοσελίδα. Ο κώδικας PHP μπορεί επίσης να εκτελεστεί με μία διεπαφή γραμμής εντολών και μπορεί να χρησιμοποιηθεί για να υλοποιήσει αυτόνομες εφαρμογές γραφικών .

Ο standard διερμηνέας PHP, με την υποστήριξη της Zend Engine, είναι ένα δωρεάν λογισμικό που εκδίδεται με την άδεια PHP. Η γλώσσα PHP έχει ευρέως διαδοθεί και μπορεί να εφαρμοστεί στους περισσότερους διακομιστές ιστού, σε σχεδόν όλα τα λειτουργικά συστήματα και τις πλατφόρμες χωρίς χρέωση.

Η PHP είναι μία γλώσσα στην οποία μπορούμε να συναντήσουμε πολλά θετικά χαρακτηριστικά, που βοηθούν στην εύκολη και άμεση υλοποίηση ιστοσελίδων. Κάποια από αυτά αναφέρονται παρακάτω:

Ταχύτητα: Είναι σχετικά γρήγορη αφού χρησιμοποιεί πολλούς πόρους του συστήματος.

Σταθερότητα: Είναι σταθερή γλώσσα, εννοώντας ότι συντηρείται από πολλούς προγραμματιστές κι έτσι σε περίπτωση εμφάνισης σφαλμάτων, η διόρθωση είναι άμεση.

Ευκολία χρήσης: Χρησιμοποιεί σύνταξη τύπου γλώσσας C. Έτσι, ιδιαίτερα για άτομα που είναι εξοικειωμένα με τη C, είναι πολύ εύκολο να δημιουργήσουν και ανεβάσουν κείμενα ιστοσελίδων.

Ανοιχτή πηγή: Η PHP έχει αναπτυχθεί και έχει συντηρηθεί από μία μεγάλη ομάδα προγραμματιστών, διαμορφώνοντας μία ομάδα υποστήριξης.

Ισχυρή βιβλιοθήκη υποστήριξης: Μπορεί εύκολα κανείς να βρει λειτουργικές ενότητες όπως pdf, γραφήματα κλπ.

Ενσωματωμένες μονάδες σύνδεσης βάσης δεδομένων: Πραγματοποιείται εύκολα η δυνατότητα σύνδεσης σε βάση δεδομένων, καθώς πολλές ιστοσελίδες αποτελούνται από δεδομένα. Έτσι χρησιμοποιείται συχνά η βάση δεδομένων, γεγονός που μειώνει σημαντικά το χρόνο ανάπτυξης εφαρμογών του ιστού.

Εκτελέσιμη σε πολλά λειτουργικά: Τα Windows, Linux και Mac είναι κάποια από τα διάφορα λειτουργικά συστήματα πάνω στα οποία μπορεί να τρέξει η PHP.

3.2 HTML

Η HTML είναι η τυπική γλώσσα σήμανσης για τη δημιουργία ιστοσελίδων και εφαρμογών ιστού. Μαζί με τη CSS και τη JavaScript αποτελούν μια βασική τριάδα τεχνολογιών για τον παγκόσμιο ιστό. Τα Web προγράμματα περιήγησης λαμβάνουν έγγραφα HTML από έναν διακομιστή ιστού ή από έναν τοπικό αποθηκευτικό χώρο και τα αποδίδουν σε ιστοσελίδες πολυμέσων. Η HTML περιγράφει τη σημασιολογική δομή μιας ιστοσελίδας, ενώ στην αρχή περιελάμβανε στοιχεία για την εμφάνιση του εγγράφου.

Τα στοιχεία HTML είναι τα δομικά στοιχεία των σελίδων HTML. Με τη χρήση της HTML, μπορεί κανείς να ενσωματώσει στην ιστοσελίδα του εικόνες, βίντεο και πολλές άλλες εφαρμογές. Παρέχουν ένα μέσο για τη δημιουργία δομημένων εγγράφων με την ένδειξη διαρθρωτικής σημασιολογίας για κείμενο όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους, εισαγωγικά και άλλα στοιχεία. Τα στοιχεία HTML οριοθετούνται από ετικέτες, γραμμένα με γωνιακές παρενθέσεις. Ετικέτες όπως `` και `<input />` εισάγουν άμεσα περιεχόμενο στη σελίδα. Άλλοι, όπως το `<p> ... </p>`, περιβάλλουν και παρέχουν πληροφορίες σχετικά με το κείμενο του εγγράφου και ενδέχεται να περιλαμβάνουν και άλλες ετικέτες ως επιμέρους στοιχεία. Τα προγράμματα περιήγησης δεν εμφανίζουν τις ετικέτες HTML, αλλά τα χρησιμοποιούν για την ερμηνεία του περιεχομένου της σελίδας.

Η HTML μπορεί να ενσωματώσει προγράμματα γραμμένα σε μια γλώσσα δέσμης ενεργειών, όπως JavaScript, που επηρεάζουν τη συμπεριφορά και το περιεχόμενο των ιστοσελίδων. Τέλος η ενσωμάτωση της CSS, την οποία θα αναλύσουμε πιο κάτω, ορίζει την εμφάνιση και τη διάταξη του περιεχομένου

Η HTML είναι αρκετά διαδεδομένη γλώσσα ώστε να μας οδηγήσει στην επιλογή να την προτιμήσουμε για τη δημιουργία του site μας. Ας δούμε κάποια από τα πλεονεκτήματά της:

Συνδυασμός:

Ευέλικτη: Διαθέτει ένα πλήθος επιλογών για τη χρήση πολλών χαρακτηριστικών που θα ήθελε ένας προγραμματιστής να προσθέσει στην ιστοσελίδα του.

Υποστήριξη από πολλούς φυλλομετρητές: Είναι βέβαιο ότι η υποστήριξη από σχεδόν όλους τους φυλλομετρητές, μόνο ικανοποίηση μπορεί να προσφέρει στον προγραμματιστή

Εύκολα κατανοητή: Η HTML είναι μία γλώσσα η οποία είναι εύκολα κατανοητή, γι αυτό μπορεί κανείς να τη διδαχτεί από μόνος του. Επίσης, είναι τόσο διαδεδομένη που υπάρχουν πολλές εφαρμογές προσομοίωσης, ώστε να είναι δυνατή η εκμάθηση και η εξάσκηση.

Συνεπής και αποτελεσματική: Οι κανόνες της είναι πολύ συγκεκριμένοι ούτως ώστε εάν οποιοσδήποτε τους ακολουθήσει κατά γράμμα, να έχει και το κατάλληλο

αποτέλεσμα.

Μικρός χρόνος φόρτωσης: Οι ιστοσελίδες που έχουν δημιουργηθεί με HTML δεν απαιτούν μεγάλο χρόνο φόρτωσης.

3.3 CSS

Η CSS) είναι μια γλώσσα που χρησιμοποιείται για την περιγραφή της παρουσίασης ενός εγγράφου γραμμένου σε μια γλώσσα σήμανσης. Παρόλο που χρησιμοποιείται συνήθως για τη ρύθμιση του οπτικού στυλ των ιστοσελίδων και των διεπαφών χρήστη που είναι γραμμένα σε HTML και XHTML, η γλώσσα μπορεί να εφαρμοστεί σε οποιοδήποτε έγγραφο XML, συμπεριλαμβανομένων των απλών XML, SVG και XUL, και εφαρμόζεται στην απόδοση σε ομιλία ή σε άλλη μέσο. Μαζί με την HTML και τη JavaScript, η CSS αποτελεί βασική τεχνολογία που χρησιμοποιείται από τους περισσότερους ιστότοπους για τη δημιουργία οπτικών σελίδων, διεπαφών του χρήστη για εφαρμογές ιστού και διεπαφών του χρήστη για πολλές εφαρμογές κινητής τηλεφωνίας.

Η CSS σχεδιάστηκε κυρίως για να επιτρέψει τον διαχωρισμό των εγγράφων ενός περιεχομένου από την παρουσίαση των εγγράφων, όπως η διάταξη, τα χρώματα και οι γραμματοσειρές [Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, να δώσει μεγαλύτερη ευελιξία και έλεγχο στις προδιαγραφές των χαρακτηριστικών παρουσίασης, να επιτρέψει σε πολλαπλές σελίδες HTML να μοιράζονται μορφοποίηση, καθορίζοντας τη σχετική CSS σε ξεχωριστό αρχείο με κατάληξη .css και μειώνοντας την πολυπλοκότητα και την επανάληψη στο δομικό περιεχόμενο.

Ο διαχωρισμός της μορφοποίησης και του περιεχομένου καθιστά δυνατή την παρουσίαση της ίδιας σελίδας σήμανσης σε διαφορετικά στυλ για διαφορετικές μεθόδους απόδοσης, όπως επί της οθόνης, σε έντυπη μορφή, με φωνή (μέσω προγράμματος περιήγησης ομιλίας ή προγράμματος ανάγνωσης οθόνης) συσκευές. Μπορεί επίσης να εμφανίσει διαφορετικά την ιστοσελίδα ανάλογα με το μέγεθος της οθόνης ή τη συσκευή προβολής. Οι αναγνώστες μπορούν επίσης να καθορίσουν ένα διαφορετικό φύλλο στυλ, όπως ένα αρχείο CSS που είναι αποθηκευμένο στον δίσκο του υπολογιστή, για να αντικαταστήσει αυτό που έχει καθορίσει ο δημιουργός.

Αλλαγές στο γραφικό σχέδιο ενός εγγράφου (ή εκατοντάδων εγγράφων) μπορούν να εφαρμοστούν γρήγορα και εύκολα, με την επεξεργασία μερικών γραμμών στο αρχείο CSS που χρησιμοποιούν, αντί να αλλάξουν σήμανση στα έγγραφα.

Το πρότυπο CSS περιγράφει ένα σχήμα προτεραιότητας για να καθορίσει ποιι κανόνες στυλ ισχύουν, εάν περισσότεροι από ένας κανόνες συμφωνούν με ένα συγκεκριμένο στοιχείο. Σε αυτή τη διαδοχή, οι προτεραιότητες υπολογίζονται και

εκχωρούνται σε κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλέψιμα.

Τα πλεονεκτήματα της CSS είναι σημαντικά και είναι τα ακόλουθα:

Εξοικονόμηση χρόνου: Εφόσον εισάγουμε στη σελίδα μας μία φορά CSS, τότε ο ίδιος ο κώδικας μπορεί να εφαρμοστεί στις ομάδες των στοιχείων της CSS και μπορεί να επαναχρησιμοποιηθεί σε πολλές σελίδες HTML. Επίσης, παρέχει μεγάλη ευελιξία στον ορισμό ιδιοτήτων ενός στοιχείου.

Εύκολη συντήρηση: Παρέχει ένα μέσο ενημέρωσης για τη μορφοποίηση των εγγράφων αλλά και τη διατήρηση της συνοχής τους σε πολλά έγγραφα. Πραγματοποιώντας μία αλλαγή στη CSS του ιστότοπου, τα στοιχεία σε όλες τις ιστοσελίδες θα ενημερώνονται αυτόματα.

Γρήγορη φόρτωση σελίδων: Η CSS επιτρέπει σε πολλαπλές σελίδες, να μοιράζονται τη μορφοποίηση, να μειώνουν την πολυπλοκότητα και την επανάληψη στο δομικό περιεχόμενο. Μειώνει σημαντικά το μέγεθος μεταφοράς αρχείων, με αποτέλεσμα την επίτευξη της ταχύτερης φόρτωσης των σελίδων.

Συμβατότητα πολλαπλών συσκευών: Η CSS επιτρέπει τη βελτιστοποίηση ενός εγγράφου HTML για περισσότερους από έναν τύπο συσκευών ή μέσων. Δηλαδή με τη χρήση της, το ίδιο έγγραφο HTML μπορεί να παρουσιαστεί διαφορετικά σε οθόνη υπολογιστή, σε κινητά τηλέφωνα, εκτυπωτές κλπ.

Κεφάλαιο 4 Χρήση Προγραμμάτων

4.1 Adobe Dreamweaver

Το Adobe Dreamweaver είναι ένα πρόγραμμα λογισμικού για το σχεδιασμό ιστοσελίδων. Ουσιαστικά είναι ένα πρόγραμμα προγραμματισμού και επεξεργασίας της HTML. Το πρόγραμμα παρέχει μια διασύνδεση, με την οποία ο χρήστης έχει τη δυνατότητα να βλέπει το αποτέλεσμα την ώρα που προγραμματίζει, έτσι ώστε να είναι ικανός να δημιουργεί και να επεξεργάζεται ιστοσελίδες σε ένα πιο φιλικό περιβάλλον.

Το Dreamweaver υποστηρίζει πολλές γλώσσες όπως για παράδειγμα HTML, XML, CSS και JavaScript. Όσο για τις γλώσσες λειτουργίας, το πρόγραμμα υποστηρίζει αγγλικά, ισπανικά, γαλλικά, γερμανικά, ιαπωνικά, κινέζικα, ιταλικά, ρωσικά και πολλά άλλα.

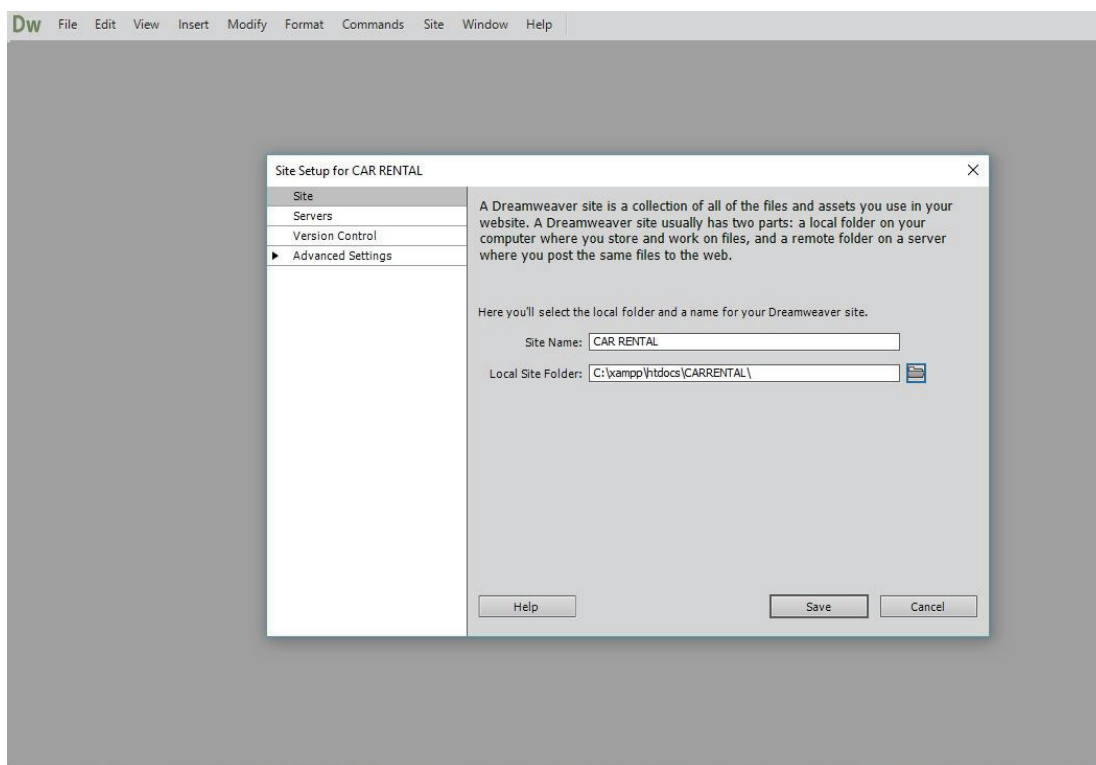
Αναπτύχθηκε και κυκλοφόρησε το 1997 από τη Macromedia. Το 2005 η Adobe αγόρασε την Macromedia, η οποία είχε στην κατοχή της τα δικαιώματα του

Dreamweaver, και συνέχισε την ανάπτυξη του προγράμματος Η ύπαρξη πολλών εργαλείων καθιστούν το πρόγραμμα ευέλικτο για την δημιουργία και την επεξεργασία απλών ή πολύπλοκων ιστοσελίδων.

Το πρόγραμμα διαθέτει πολλά πλεονεκτήματα, καθιστώντας το πολύ χρήσιμο με σημαντικότερο την **αναγνώριση κώδικα (Code highlighting)** που βοηθάει στην γρήγορη σάρωση του κώδικα καθώς και στον εντοπισμό τυχόν σφαλμάτων. Με κάθε είδος κώδικα υπάρχει μία διαφορετική σειρά από highlights που βοηθούν στην αναγνώριση των γλωσσών όπως HTML, CSS, PHP, JavaScript κλπ. Αυτό είναι πραγματικά χρήσιμο, εφόσον έχουμε πολύ κώδικα σε μία σελίδα που είναι για παράδειγμα μείγμα HTML και δυναμικών γλωσσών όπως PHP ή ASP. Ιδιαίτερα για τους αρχάριους, ο προσδιορισμός του κώδικα βοηθάει στην πιο αποτελεσματική κατανόηση του κάθε μέρους του, καθώς και στον εντοπισμό των λαθών εάν αυτά υπάρχουν. Κατά συνέπεια, στην περίπτωση που συναντήσουμε μπλε χρώμα σε μία γραμμή κώδικα, μας είναι πολύ εύκολο να εντοπίσουμε ότι υπάρχει λάθος.

Στη παρακάτω εικόνα βλέπουμε τα πρώτα βήματα της δημιουργίας της ιστοσελίδας στο Dreamweaver.

Η ονομασία είναι Car rental και την αποθηκεύουμε στην εξής τοποθεσία C:\xampp\htdocs\CARRENTAL έτσι ώστε να υπάρχει η επικοινωνία μεταξύ της ιστοσελίδας, της βάσης δεδομένων και του server.

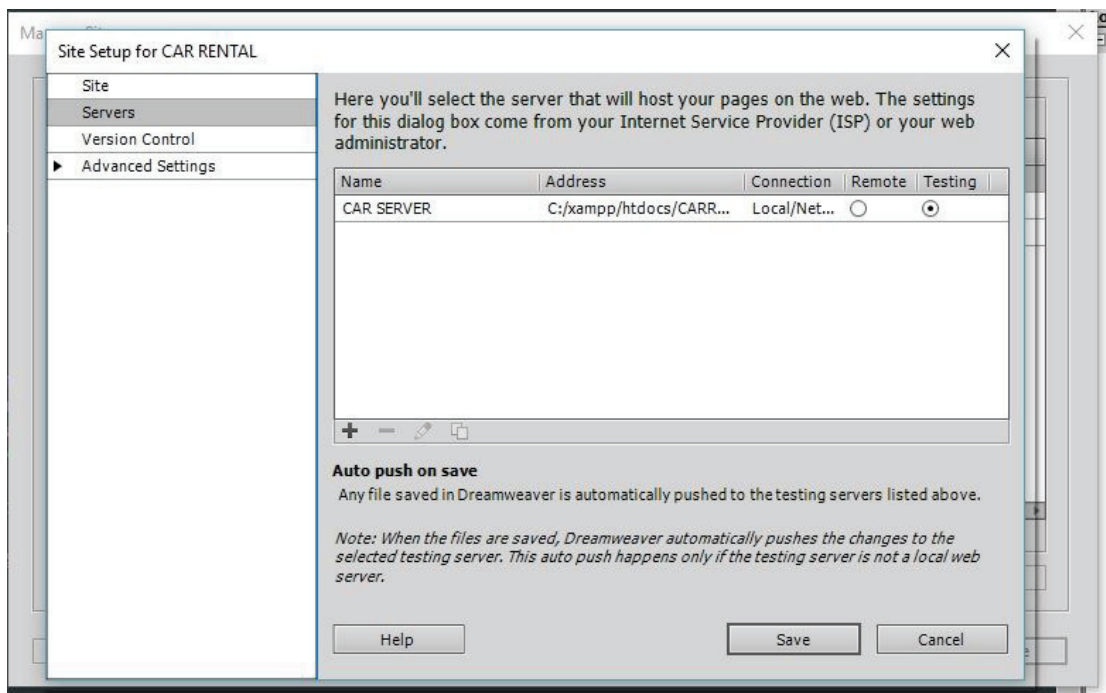


Εικόνα 4.1

Ρύθμιση SERVER

Στη παρακάτω εικόνα βλέπουμε τη διαδικασία με την οποία συνδέουμε ένα server στην ιστοσελίδα μας.

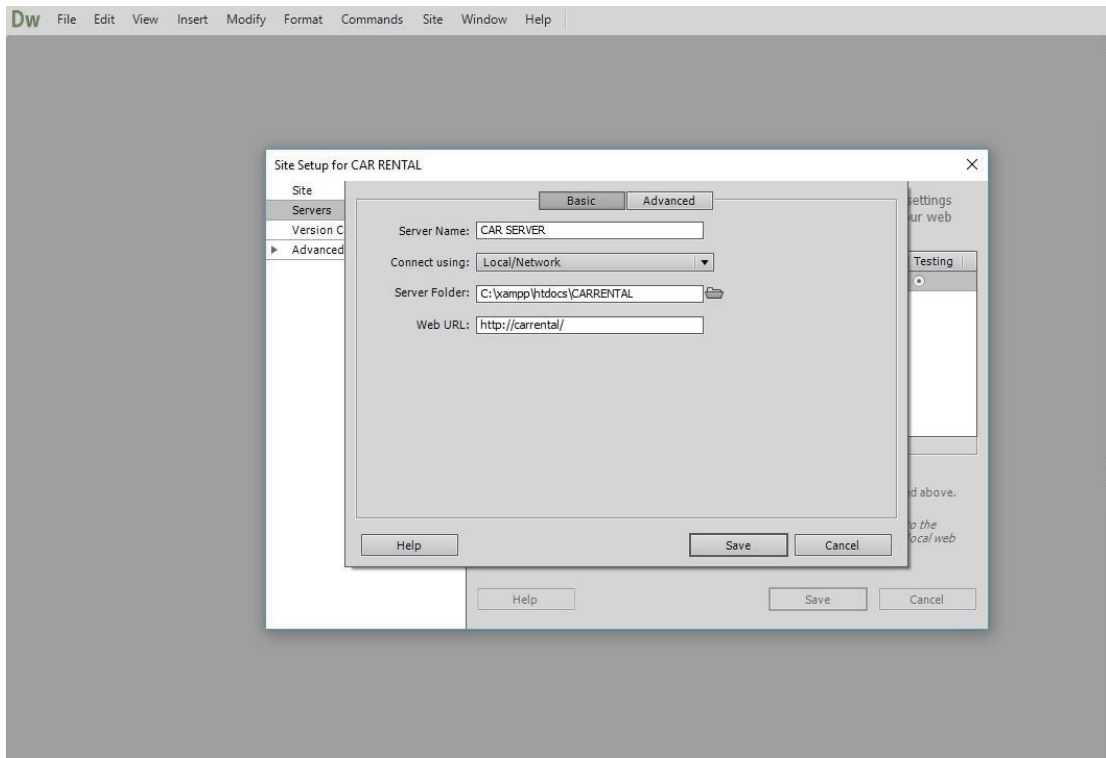
Για να λειτουργήσει πρέπει να έχουμε επιλέξει την επιλογή testing.



Εικόνα 4.2

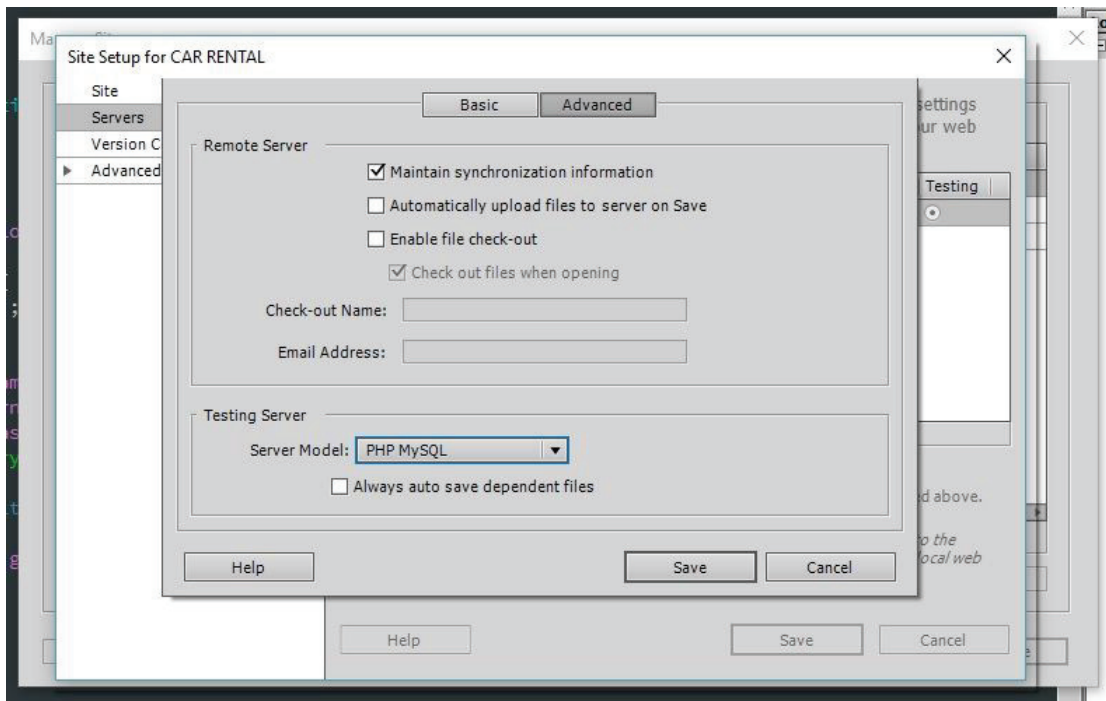
Στη συνέχεια ρυθμίζουμε το server, του δίνουμε ένα όνομα και επιλέγουμε στην επιλογή Connect using : Local/network.

Για να συνδεθεί ο server με την ιστοσελίδα μας θα πρέπει να αποθηκευτεί στον ίδιο φάκελο όπως βλέπουμε και στη παρακάτω σελίδα.



Εικόνα 4.3

Τέλος πηγαίνοντας στην καρτέλα Advanced επιλέγουμε στο Server model την επιλογή Php MySQL



Εικόνα 4.4

4.2 Apache

Το Apache HTTP Server είναι το πιο δημοφιλές web server software παγκοσμίως, βασισμένο πάνω στο NCSA server. Η ανάπτυξη του ξεκίνησε το 1995, ενώ είχε βασικό ρόλο και στην ανάπτυξη του παγκόσμιου διαδικτύου καθώς ανακηρύχθηκε ως ο πιο δημοφιλής αλλά και κύριος HTTP Server μέχρι σήμερα. Μάλιστα το 2009, έγινε ο πρώτος web server που εξυπηρέτησε πάνω από εκατό εκατομμύρια ιστοσελίδες. Αναπτύσσεται και διατηρείται από μία ομάδα από developers που χρησιμοποιούν συνήθως λειτουργικό σύστημα Linux, ενώ είναι διαθέσιμο για Microsoft Windows και άλλα λειτουργικά συστήματα. Επίσης είναι ελεύθερο και ανοιχτό πρόγραμμα, για όποιον θέλει να το χρησιμοποιήσει

Κάποιες από τις γλώσσες που υποστηρίζει το πρόγραμμα είναι η PHP, η Perl, η Python, η Tcl. Επίσης, διαθέτει ένδειξη μηνύματος σε περίπτωση λαθών, τα οποία αναγνωρίζονται με βάση τη DBMS βάση δεδομένων. Η ομάδα υποστήριξης του Apache αποτελείται από GUIs (graphical user interfaces). Τέλος, επειδή η πηγή του κώδικα είναι ελεύθερη, υπάρχει η δυνατότητα σύνδεσης στον server από οποιονδήποτε για ότι χρειάζεται.

Ένας τόσο διαδεδομένος web server δε θα μπορούσε να μην έχει πολλά οφέλη κάποια εκ των οποίων είναι η διάθεση πλούσιων χαρακτηριστικών στον server, η ελεύθερη παροχή του, η μεγάλη ιστορία από αξιοπιστία και παρουσιάσεις, το πλήθος τεκμηριωμένων αρχείων βοηθώντας στην επίλυση προβλημάτων, η συμβατότητα σχεδόν σε όλα τα λειτουργικά συστήματα (Linux, Windows, MacOS) και η συχνή ενημέρωση του server.

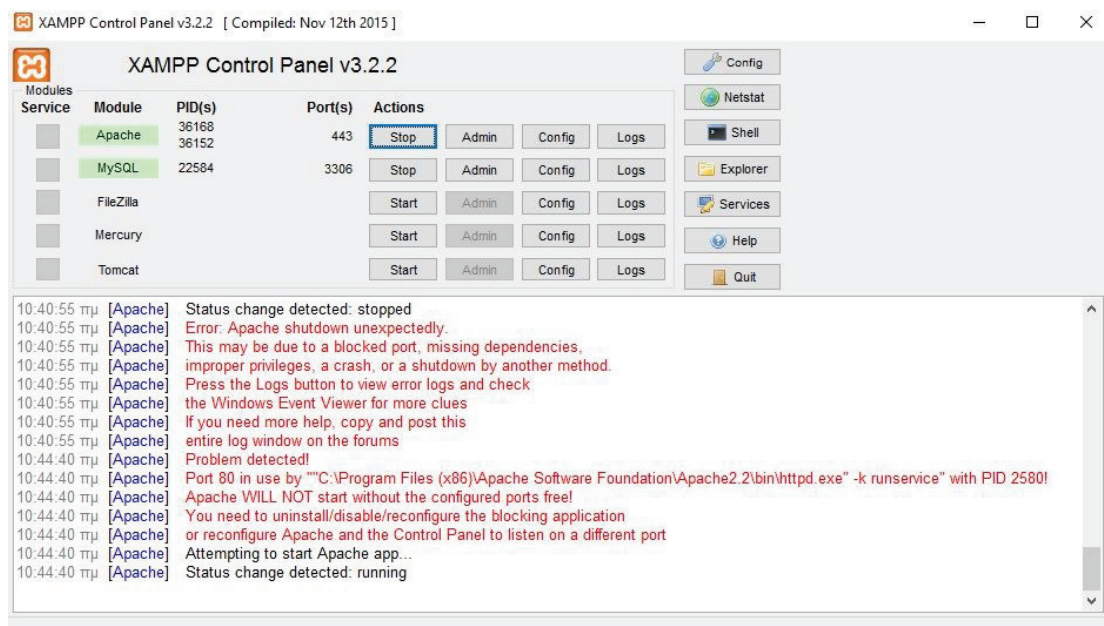
4.3 Xampp

Είναι ένα δωρεάν ανοιχτό πρόγραμμα που αποτελείται κυρίως από τον Apache HTTP Server, το database και από προγραμματιστές της PHP και της Perl. Η ονομασία προέρχεται από το Cross platform (X), το Apache (A), το MariaDB (M), την PHP (P) και από την Perl (P). Η ύπαρξη του Apache στο Xampp, βοηθάει στην διευκόλυνση των προγραμματιστών για δημιουργία ενός τοπικού web server, για αναπτυξιακούς λόγους αλλά και για εξέταση. Το Xampp λειτουργεί εξίσου καλά σε Linux, Mac και Windows. Οι ενημερώσεις του προγράμματος είναι τακτικές και γίνονται ανάλογα με τις τελευταίες εκδόσεις των Apache, MariaDB, PHP και Perl. Επιπροσθέτως, πολλά αποτελέσματά του, μπορούν να αντιγραφούν από τον έναν υπολογιστή στον άλλον.

Αρχικά, ο σκοπός των σχεδιαστών του Xampp, ήταν να το χρησιμοποιήσουν ως αναπτυξιακό εργαλείο, ούτως ώστε να επιτρέπουν στους σχεδιαστές και στους προγραμματιστές ιστοσελίδων να εργάζονται στους υπολογιστές τους χωρίς να

χρειάζεται πρόσβαση στο διαδίκτυο. Για το λόγο αυτό, πολλά σημαντικά στοιχεία του προγράμματος είναι ασφαλισμένα και δεν είναι διαθέσιμα. Ωστόσο, υπάρχει η ικανότητα εξυπηρέτησης ιστοσελίδων που ανήκουν στον παγκόσμιο διαδίκτυο. Ακόμα, εφόσον πραγματοποιηθεί η εγκατάσταση σε έναν υπολογιστή, μπορούμε να φορτώσουμε το πρόγραμμα σε ένα usb, ώστε να υπάρχει η δυνατότητα να χρησιμοποιήσουμε το πρόγραμμα σε όποιο άλλο υπολογιστή θέλουμε, χωρίς να χρειάζεται νέα εγκατάσταση.

Στη παρακάτω εικόνα φαίνεται ότι οι λειτουργίες του Apache server και της βάσης δεδομένων MySQL έχουν ενεργοποιηθεί στο xampp



Εικόνα 4.5

Κεφάλαιο 5 Αρχιτεκτονική του Συστήματος

5.1 Η Αρχιτεκτονική του Συστήματος – Επίπεδα (Tiers) Λειτουργίας

Το σύστημα αποτελείται από τρία διακριτά επίπεδα:

- Database Server
- Application Server
- Client

Πρώτο επίπεδο (First Tier) – Database Server

Ο Database server είναι το κυριότερο μέρος του συστήματος, καθώς κατέχει όλες τις αναγκαίες ενέργειες προκειμένου τα δεδομένα του συστήματος να μπορούν να αποθηκευθούν, να ανακτηθούν, να ενημερωθούν και να συντηρηθούν. Παράλληλα, διαθέτει τους κατάλληλους μηχανισμούς για την ακεραιότητα των δεδομένων (Data Integrity).

Δεύτερο επίπεδο (Second Tier) – Application Server

Σε αυτό το επίπεδο πραγματοποιούνται οι περισσότερες λειτουργίες, καθώς ο Application Server αποτελεί βασικό κομμάτι του λογισμικού. Εκμεταλλευόμενοι το γεγονός ότι τα Application Servers μπορούν να εγκατασταθούν σε πολλά μηχανήματα, υπάρχει η δυνατότητα αξιοποίησης της υπολογιστικής ισχύς, καθώς και η δυνατότητα εξασφάλισης εξαιρετικών αποτελεσμάτων ανταπόκρισης, αξιοπιστίας και επεκτασιμότητας

Επιπροσθέτως, το σύστημα λειτουργεί αρμονικά καθώς κάποια από τα Remote Clients υποστηρίζονται από τον εκάστοτε Application Server.

Τρίτο Επίπεδο (Third Tier) – Client

Σ' αυτό το επίπεδο συναντάμε την επαφή του χρήστη με το σύστημα (User Interface). Πραγματοποιείται έλεγχος των οθονών εργασίας (User Screens), όπως επίσης και η μορφοποίηση των δεδομένων που εμφανίζονται. Η επικοινωνία μεταξύ Client και Application Server επιτυγχάνεται με τη χρησιμοποίηση ενός πακέτου δεδομένων. Έτσι λαμβάνουμε τον βέλτιστο χρόνο απόκρισης, αφού με τη χρήση μιας τηλεπικοινωνιακής γραμμής (LAN, Internet Connection), το σύστημα αποκτά μικρούς χρόνους απόκρισης

Η συγκεκριμένη αρχιτεκτονική (Three Tier) έχει αναγνωριστεί ως η πιο κατάλληλη για εφαρμογές διαδικτύου.

Η λειτουργία του συστήματος σε τρία επίπεδα διασφαλίζει:

Την ελάφρυνση του δικτύου από τη μεταφορά μεγάλης ποσότητας δεδομένων π.χ. η εκτέλεση ενός Query για την ανάκτηση μερικών εγγραφών από έναν πίνακα με πολλά έγγραφα γίνεται στο διακομιστή εφαρμογής (Application Server), από τον οποίο μεταφέρεται στο χρήστη μόνο το αποτέλεσμα

Την ευχέρεια κατηγοριοποίησης του διακομιστή δεδομένων (Database Server) από τους διακομιστές εφαρμογής (Application Servers), ούτως ώστε να εκτελούνται σε διαφορετικά μηχανήματα. Έτσι, η απόδοση των μηχανών είναι ανεξάρτητη, ενώ ταυτόχρονα διασφαλίζεται επεκτασιμότητα του λογισμικού, χωρίς να χρειάζεται η ανακατασκευή του.

Την προσαρμοστικότητα στην επιλογή του Database Server, καθώς υπάρχει η δυνατότητα χρήσης διάφορων μηχανημάτων με οποιοδήποτε λειτουργικό σύστημα (π.χ. Windows), εφόσον υπάρχει δίαυλος επικοινωνίας μέσω του TCP/IP πρωτοκόλλου. Κατά συνέπεια, η αναβάθμιση του μηχανήματος επηρεάζει την αναβάθμιση στη βάση δεδομένων, αλλά όχι το υπόλοιπο σύστημα.

Ασφάλεια Δεδομένων του Συστήματος

Η ακεραιότητα (integrity), διαθεσιμότητα (availability), εμπιστευτικότητα (confidentiality), είναι τρία βασικά συστατικά που διαμορφώνουν την ασφάλεια στα πληροφοριακά συστήματα.

Η **ακεραιότητα** περιγράφει τη συντήρηση των δεδομένων του πληροφοριακού συστήματος σε γνωστές περιπτώσεις, χωρίς ανεπιθύμητες αλλαγές από άτομα χωρίς εξουσιοδότηση. Παράλληλα τους αποτρέπεται η πρόσβαση και η χρήση των υπολογιστών και των δικτύων του συστήματος

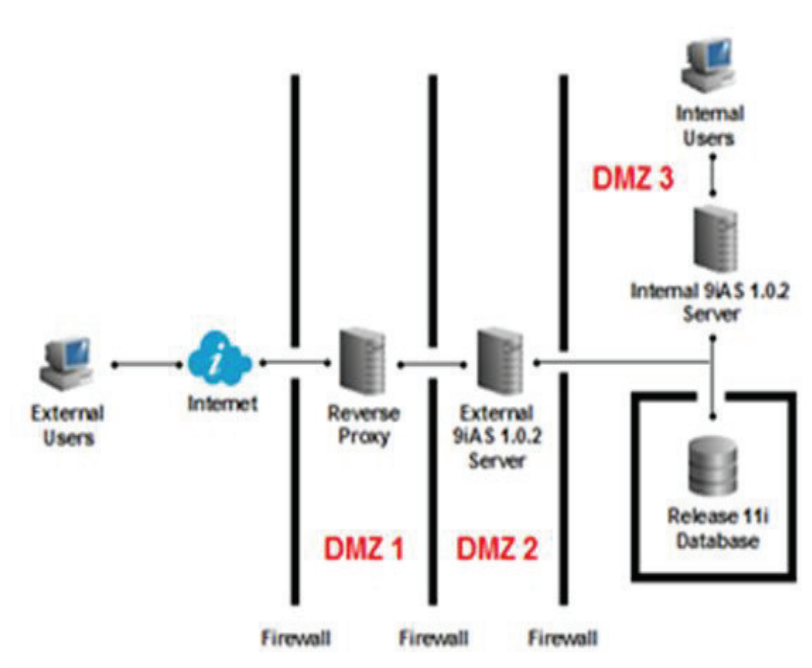
Η **διαθεσιμότητα** εξασφαλίζει ότι τα δεδομένα των υπολογιστών θα είναι ορατά όποτε αυτά ζητηθούν από τους χρήστες.

Η **εμπιστευτικότητα** αντικατοπτρίζει την ιδιωτικότητα που υπάρχει στις πληροφορίες στα άτομα που έχουν εξουσιοδότηση. Τα μη εξουσιοδοτημένα άτομα δεν έχουν πρόσβαση σε αυτές.

Ακεραιότητα

Reserve proxy με firewall τριών ζωνών

Με την ακεραιότητα καταπολεμάτε η αλλοίωση των δεδομένων και αυτό συμβαίνει με πολλές λύσεις, μία εκ των οποίων είναι η χρήση ενός reserve proxy, δύο web server ενός external και ενός internal και ενός database server. Το συγκεκριμένο σύστημα αποτελείται από τρεις ζώνες κίνησης, με τρία firewall όπως θα δούμε και στην εικόνα που ακολουθεί.



Εικόνα 5.1

Ο reserve proxy είναι ένας server που επιδιώκει την βέλτιστη ταχύτητα πλοήγησης στο διαδίκτυο (proxy server) και βρίσκεται σε ένα διακομιστή δικτύου μπροστά από τον web server. Από τον reserve proxy περνάνε στον web server, όλες οι συνδέσεις του Internet.

Το τείχος προστασίας (firewall) είναι η πρώτη μορφή άμυνας στις κακόβουλες επιθέσεις, ενώ μπορεί να βρεθεί είτε σε software είτε σε hardware. Επεξεργάζεται την κίνηση των δεδομένων μεταξύ του ίδιου και του διαδικτύου, ενώ διέπεται από κανόνες προκειμένου να ληφθεί απόφαση για το αν θα δώσει την άδεια στη διέλευση των δεδομένων στο σύστημα. Σε περίπτωση που ένα πακέτο δεδομένων παραβεί τους κανόνες απορρίπτεται

Reserve proxy με firewall δύο ζωνών

Εάν το σύστημα δημιουργηθεί μόνο με έναν web-application server, τότε μαζί με την εισαγωγή του username και του password για τις καταχωρήσεις, είναι καλό να χρησιμοποιηθεί η υποδομή Δημόσιου Κλειδιού ή η υποδομή e-token.

Χρήση Δημόσιου Κλειδιού (Public Key Infrastructure, PKI)

Είναι ένας συνδυασμός λογισμικού, τεχνολογιών, διαδικασιών καθώς και ασύμμετρης κρυπτογραφίας, που αναγνωρίζει την εγκυρότητα οποιασδήποτε

ενέργειας πραγματοποιείται σε ψηφιακή συναλλαγή.

Η υποδομή δημόσιου κλειδιού εγγυάται τον έλεγχο της ταυτότητας μιας οντότητας, υπογράφοντας το δημόσιο κλειδί της δημοσιεύοντας το με επιμέρους πληροφορίες, σε ένα πιστοποιητικό Επιπλέον, έχει στην κατοχή του λίστες με έγκυρα, ληγμένα και ανακληθέντα πιστοποιητικά

Οι βασικότεροι μηχανισμοί ασφάλειας της υποδομής δημόσιου κλειδιού είναι:

- Απόρρητο της επικοινωνίας (Confidentiality):

Ειδικεύεται στην προστασία των δεδομένων πραγματοποιώντας ελέγχους σε περίπτωση που γίνει προσπάθεια πρόσβασης σε αυτά από μη εξουσιοδοτημένα άτομα.

- Ακεραιότητα (Integrity):

Τα δεδομένα παραμένουν ακέραια καθώς οι μηχανισμοί κρυπτογράφησης προστατεύουν από τυχόν παραποιήσεις που επιδιώκονται .

- Πιστοποίηση (Authentication):

Επιβεβαιώνονται τα στοιχεία των ατόμων ή γενικότερα της προέλευσης των δεδομένων.

Χρήση E-Token

Η υποδομή e-token παράγει κωδικούς μίας χρήσης για την εισαγωγή δεδομένων. Το σύστημα θα μπορούσε να παράγει token ίδιου αριθμού με τα τερματικά που θα εισάγουν τα στοιχεία στο σύστημα.

Διαθεσιμότητα

Τα πληροφοριακά συστήματα έρχονται αντιμέτωπα συχνά με τον κίνδυνο της επίθεσης άρνησης υπηρεσιών (DOS attack), κάτι το οποίο έχει ως αποτέλεσμα οι πόροι του συστήματος να βρίσκονται εκτός λειτουργίας για προσωρινό ή μόνιμο διάστημα. Ωστόσο, το φαινόμενο αυτό δεν είναι αναγκαίο να προέρχεται από κακοπροαίρετες προθέσεις. Μία ιστοσελίδα έχει τη δυνατότητα να διατηρεί στην κατοχή της συνδέσμους των οποίων η σύνδεση είναι χαμηλής χωρητικότητας Έτσι, σε περίπτωση υπερφόρτωσης του συνδέσμου από τους χρήστες, υπερφορτώνεται και η σύνδεση της αντίστοιχης ιστοσελίδας Το φαινόμενο αυτό ονομάζεται Slashdot. Για την αποφυγή μίας τέτοιας περίπτωσης η χρήση ενός reserve proxy ή ενός router, ώστε να μοιράζεται το φορτίο σε άλλους reserve proxy που θα επικοινωνούν με τον web server, αποτελούν μία πολύ καλή λύση.

Εμπιστευτικότητα

Η υποχρέωση και η ευθύνη τα δεδομένα να παραμένουν αναλλοίωτα και κρυφά όταν αποσταλούν είναι μέρος της εμπιστευτικότητας. Με τη χρήση της Three-tier αρχιτεκτονικής μόνο ο application server επικοινωνεί με τη βάση δεδομένων.

Γνωστές απειλές ασφάλειας – SQL injection

Ταυτολογίες

Όταν μία επίθεση στηρίζεται σε ταυτολογία, σημαίνει ότι γίνεται απόπειρα εισαγωγής κώδικα με τέτοιο τρόπο ώστε οι συνθήκες να είναι πάντα αληθείς, ούτως ώστε να γίνει εισχώρηση στη βάση δεδομένων. Αυτό βέβαια εξαρτάται από τον τρόπο με τον οποίο διατυπώνεται το ερώτημα μέσα στην εφαρμογή. Οι πιο συνηθισμένες χρήσεις είναι η παράκαμψη διαδικασίας πιστοποίησης και η εξαγωγή δεδομένων.

Μηνύματα λάθους

Είναι μία κατηγορία επίθεσης που στοχεύει στην φανέρωση σημαντικών πληροφοριών για το περιεχόμενο και την μορφή της βάσης δεδομένων, μέσω μηνυμάτων σφάλματος που επιστρέφονται από την εφαρμογή. Κατά συνέπεια, ο επιτιθέμενος είναι σε θέση να διαπιστώσει τις ονομασίες των πινάκων, τα ονόματα και τους τύπους δεδομένων των πεδίων ενός πίνακα και να αποτυπώσει την σύνθεση της βάσης δεδομένων.

Ερωτήματα ένωσης

Οι επιθέσεις που πραγματοποιούνται με ερωτήματα ένωσης (UNION), εκμεταλλεύονται τις ασταθής παραμέτρους του συστήματος, ούτως ώστε να μεταβληθεί το άθροισμα των δεδομένων που επιστρέφονται από ένα συγκεκριμένο SQL ερώτημα. Ακολουθώντας αυτή την επίθεση, μπορεί κάποιος να παραπλανήσει την εφαρμογή υποχρεώνοντας την να επιστρέψει δεδομένα από έναν διαφορετικό πίνακα από αυτόν που εισήγαγε ο υπεύθυνος για την ανάπτυξη της εφαρμογής.

Κεφάλαιο 6 Επίλογος

Στο πλαίσιο αυτής της πτυχιακής εργασίας αναλύσαμε την έννοια του πληροφοριακού συστήματος, τη δομή του καθώς και το πόσο σημαντικό ρόλο κατέχει σήμερα. Αναφερθήκαμε στην ασφάλεια του συστήματος καθώς και στην αντιμετώπιση τυχόν απειλών που μπορεί να προκύψουν.

Στόχος μας ήταν να δημιουργήσουμε ένα πληροφοριακό σύστημα για εταιρεία ενοικίασης αυτοκινήτων και ακολουθήθηκε η εξής διαδικασία : αρχικά προσδιορίστηκε ο στόχος καθώς και οι προδιαγραφές του συστήματος με τη βοήθεια της uml και στη συνέχεια σχεδιάστηκαν η βάση δεδομένων και οι σελίδες των χρηστών του συστήματος μας.

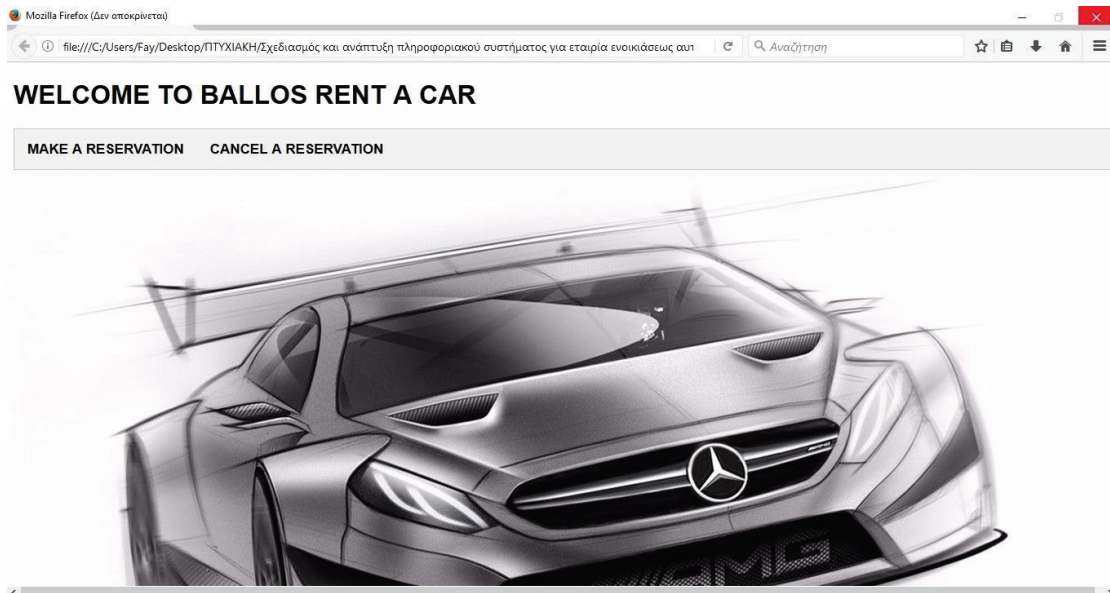
Για την υλοποίηση του συστήματος χρησιμοποιήθηκε το πρόγραμμα DreamWeaver, ως web server χρησιμοποιήθηκε ο Apache και η εφαρμογή υλοποιήθηκε με τη χρήση php, html, css και JavaScript.

Βιβλιογραφία

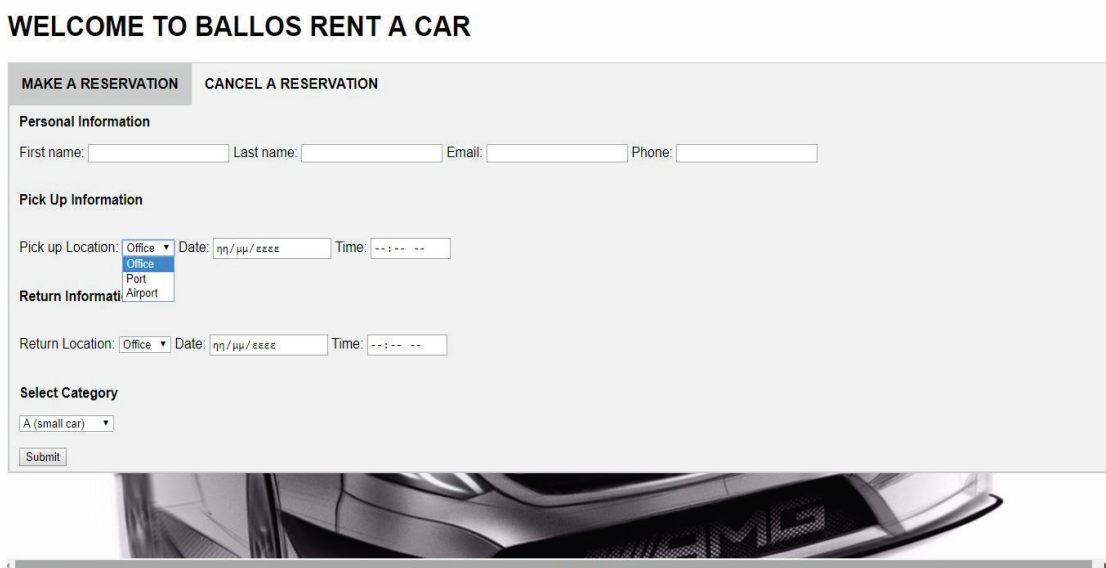
- 1) https://el.wikipedia.org/wiki/Πληροφοριακά_συστήματα
- 2) <https://www.slideshare.net/inam12/gr-1-34253962>
- 3) <https://www.slideshare.net/hiddensoul/information-system-2999864>
- 4) Εισαγωγή στις Php 6 & MySQL με εικόνες, εκδόσεις Κλειδάριθμος
- 5) <https://www.computerhope.com/jargon/d/dreamweaver.htm>
- 6) <http://www.wpbeginner.com/glossary/apache/>
- 7) https://en.wikipedia.org/wiki/Apache_HTTP_Server
- 8) <https://www.quora.com/What-are-the-benefits-advantages-of-using-an-Apache-server-over-other-servers>
- 9) <https://en.wikipedia.org/wiki/XAMPP>
- 10) <https://oncore.net/xampp-web-server.htm>
- 11) <https://stackoverflow.com/questions/8188336/benefits-of-running-apache-as-a-service>
- 12) <https://en.wikipedia.org/wiki/HTML>
- 13) https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- 14) <https://en.wikipedia.org/wiki/PHP>
- 15) <http://www.vtech-seo.com/web-design-articles/advantages-of-html.html>
- 16) <http://www.rapidprogramming.com/questions-answers/What-are-the-advantages-of-HTML--508>
- 17) <http://www.techstrikers.com/PHP/php-advantages-disadvantages.php>
- 18) <http://vanseodesign.com/css/benefits-of-cascading-style-sheets/>
- 19) <https://www.w3schools.com/>
- 20) <https://creately.com/Draw-UML-and-Class-Diagrams-Online>
- 21) Unifield Modeling Language Βασικές Αρχές Αντικειμενοστραφούς Σχεδίασης Συστημάτων και Εφαρμογών, εκδόσεις Νέων Τεχνολογιών

ΠΑΡΑΡΤΗΜΑ 1 - Διεπαφές συστήματος

Οθόνες Πελάτη



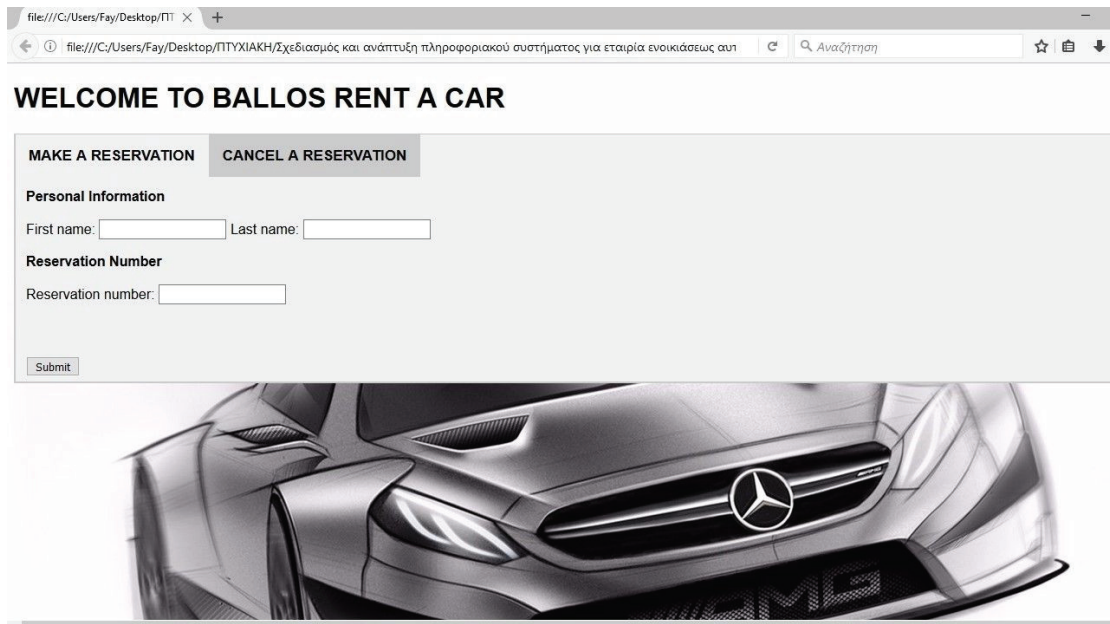
Στη παραπάνω οθόνη βλέπουμε την αρχική σελίδα του οθόνη του πελάτη. Ο πελάτης έχει 2 επιλογές “Make a reservation” σε περίπτωση που θέλει να πραγματοποιήσει μια κράτηση και “Delete a reservation” σε περίπτωση που θέλει να ακυρώσει μια υπάρχουσα κράτηση.



Παραπάνω βλέπουμε την οθόνη που εμφανίζεται στη περίπτωση που ο πελάτης διαλέξει την επιλογή “Make a reservation”, αφού συμπληρώσει τα στοιχεία που του

ζητούνται πατάει το κουμπί submit.

Όταν ο χρήστης πατήσει το κουμπί submit γίνεται έλεγχος για το αν έχει συμπληρώσει όλα τα πεδία, σε περίπτωση που κάποιο πεδίο είναι κενό τότε εμφανίζεται το ανάλογο μήνυμα



file:///C:/Users/Fay/Desktop/ΠΤ X +

file:///C:/Users/Fay/Desktop/ΠΤΥΧΙΑΚΗ/Σχεδιασμός και ανάπτυξη πληροφοριακού συστήματος για εταιρία ενοικιάσεως αυτ Αναζήτηση ☆ | 🗑 | ⬇

WELCOME TO BALLOS RENT A CAR


MAKE A RESERVATION **CANCEL A RESERVATION**

Personal Information

First name: Last name:

Reservation Number

Reservation number:



Παραπάνω βλέπουμε την οθόνη που εμφανίζεται στη περίπτωση που ο πελάτης διαλέξει την επιλογή “Cancel a reservation”, αφού συμπληρώσει τα στοιχεία που του ζητούνται πατάει το κουμπί submit. Αφού ταυτοποιηθούντα στοιχεία και είναι σωστά η κράτηση ακυρώνεται.

Οθόνες Διαχειριστή



Login Form

Username:

Password:

Ο υπάλληλος πληκτρολογεί το username και το password, εάν ο συνδυασμός είναι σωστός ο υπάλληλος συνδέεται στο σύστημα, εάν είναι λάθος το πρόγραμμα ξανά ζητάει τα στοιχεία σύνδεσης.

Οθόνες Διαχειριστή

The screenshot shows a web application interface with a dark green header containing the menu items 'HOME', 'ADD', and 'CALENDER'. Below the header is a large tan-colored form area. At the top of the form, there is a text input field labeled 'Reservation id:' with the value 'auto'. The form is organized into several sections: 'DRIVER INFORMATION' with fields for 'Name:', 'Last Name:', and 'Birthday:' (with a date format 'ηη/μμ/εεεε'); 'PICK UP INFORMATION' with a dropdown for 'Pick Up Location:' (set to 'Office'), a date field, and a time field; 'RETURN INFORMATION' with a dropdown for 'Return Location:' (set to 'Office'), a date field, and a time field; and 'CAR INFORMATION' with a 'Plate:' field. A 'SAVE' button is located at the bottom left of the form area. Below the form area, centered, is a 'Change User' button.

Η παραπάνω οθόνη εμφανίζεται στο διαχειριστή όταν πατάει από το menu Add -> Reservation.

Ο διαχειριστής συμπληρώνει τα στοιχεία της κράτησής δηλαδή τα βασικά στοιχεία του πελάτη τις πληροφορίες της παράδοσης του αυτοκινήτου στο πελάτη (μέρος, μέρα, ώρα) καθώς και τις αντίστοιχες πληροφορίες για τη μέρα της επιστροφής. Τέλος αναθέτει στη κράτησή ένα αυτοκίνητο, το οποίο όταν γίνει το συμβόλαιο μπορεί να αλλάξει, για τυπικούς λόγους έτσι ώστε να γίνεται καλύτερος έλεγχος του στόλου (τόσα αυτοκίνητα είναι ελεύθερα και πόσα όχι)

Όταν ο διαχειριστής πατάει το κουμπί save αποθηκεύονται όλα τα στοιχεία στη βάση δεδομένων.

HOME ADD CALENDER

Contract id: auto

DRIVER INFORMATION
 Name: Last Name: Birthday:
 Licence Number: Licence Expiry Date: Country:

PICK UP INFORMATION
 Pick Up Location: Office Date: Time:

RETURN INFORMATION
 Return Location: Office Date: Time:

CAR INFORMATION
 Plate:
 Days: Cost Per Day: Total Cost:
 Payment State

CARD INFORMATION
 Owner: Card Number: Expire Date:

SAVE

Change User

Η παραπάνω οθόνη εμφανίζεται στο διαχειριστή όταν πατάει από το menu Add -> Contract

Ο διαχειριστής συμπληρώνει τα στοιχεία του συμβολαίου δηλαδή τα προσωπικά στοιχεία του πελάτη/οδηγού, τα στοιχεία του διπλώματος οδήγησης, της πιστωτικής κάρτας, τις πληροφορίες της παράδοσης του αυτοκινήτου στο πελάτη (μέρος, μέρα, ώρα) καθώς και τις αντίστοιχες πληροφορίες για τη μέρα της επιστροφής. Επίσης τη πινακίδα του αυτοκινήτου που θα πάρει ο πελάτης, το κόστος ανά μέρα καθώς και το συνολικό και τέλος τη κατάσταση της πληρωμής.

Όταν ο διαχειριστής πατάει το κουμπί save αποθηκεύονται όλα τα στοιχεία στη βάση δεδομένων.

HOME ADD VIEW CARS

RESERVATION CONTRACT OK

SELECT I

CAR

Car Information

Plate: Brand:

Miles: Color:

State

Free

SAVE Rental

Out Of Order

Change User

Η παραπάνω οθόνη εμφανίζεται στο διαχειριστή όταν πατάει από το menu Add -> Car

Τη παραπάνω οθόνη τη χρησιμοποιούμε όταν θέλουμε να προσθέσουμε καπό καινούργιο αμαξί στο στόλο μας

Όταν ο διαχειριστής πατάει το κουμπί save αποθηκεύονται όλα τα στοιχεία στη βάση δεδομένων.

- Κάτω από όλες τις οθόνες υπάρχει το κουμπί “change user” το οποίο εμφανίζει πάλι την οθόνη σύνδεσης έτσι ώστε να συνδεθεί κάποιος άλλος υπάλληλος

ΠΑΡΑΡΤΗΜΑ 2 - Κώδικας Βάσης δεδομένων

Κώδικας δημιουργίας βάσης Δεδομένων

--Δημιουργία χρηστών βάσης

```
CREATE USER `fay`@`localhost` IDENTIFIED BY `$$1237` ;
```

--σύνδεση στη βάση

```
mysql -h localhost -u fay -p $$1237
```

--δικαιώματα χρηστών

```
GRANT ALL PRIVILEGES ON *.* TO `fay`@`localhost`;
```

```
FLUSH PRIVILEGES;
```

```
Mysql -h localhost -u root
```

-- Δημιουργία βάσης

```
CREATE DATABASE IF NOT EXISTS `carrental` DEFAULT CHARACTER SET  
utf8 COLLATE utf8_general_ci ;
```

- χρησιμοποίηση βάσης

```
USE `carrental` ;
```

- Δημιουργία πίνακα αυτοκίνητο

```
CREATE TABLE `carrental`.`car`(  
  `plate` VARCHAR (45) NOT NULL,  
  `category` VARCHAR (45) NOT NULL ,  
  `brand` VARCHAR (45) NOT NULL,  
  `state` VARCHAR (45) NULL,  
  `colour` VARCHAR (45) NULL,  
  `km` INT NULL ,  
  PRIMARY KEY (`plate`));
```

```

--Δημιουργία πίνακα πελάτη
CREATE TABLE `carrental`.`costumer` (
  `idcostumer` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR (45) NOT NULL,
  `lastname` VARCHAR (45) NOT NULL ,
  `phone` VARCHAR (45) NOT NULL ,
  `mail` VARCHAR (45) NOT NULL ,
  `birthday` DATE ,
  `card_owner` VARCHAR (45),
  `reservation_idres` INT ,
  `licence_owner` VARCHAR (45),
  PRIMARY KEY (`idcostumer`),
  INDEX `ixlastname` (`lastname` ASC),
  CONSTRAINT `fk_costumer_card`
  FOREIGN KEY (`card_owner`)
  REFERENCES `carrental`.`card` (`owner`)
  ON DELETE NO ACTION
  ON UPDATE CASCADE ,
  CONSTRAINT `fk_costumer_reservation`
  FOREIGN KEY (`reservation_idres`)
  REFERENCES `carrental`.`reservation` (`idres`)
  ON DELETE CASCADE
  ON UPDATE CASCADE ,
  CONSTRAINT `fk_costumer_licence`
  FOREIGN KEY (`licence_owner`)
  REFERENCES `carrental`.`licence` (`owner`)

```

ON DELETE NO ACTION

ON UPDATE CASCADE) ;

ALTER TABLE costumer AUTO_INCREMENT=1;

- Δημιουργία πίνακα δίπλωμα

```
CREATE TABLE `carrental`.`licence` (sho
`owner` VARCHAR (45 ) NOT NULL,
`number` VARCHAR (45) NOT NULL,
`date` DATE NOT NULL,
`country` VARCHAR (45) NOT NULL,
PRIMARY KEY (`owner` ) );
```

- Δημιουργία πίνακα κάρτα

```
CREATE TABLE `carrental`.`card` (
`owner` VARCHAR (45 ) NOT NULL,
`number` VARCHAR (45) NOT NULL,
`expired` DATE NOT NULL,
`company` VARCHAR (45) NOT NULL,
PRIMARY KEY (`owner` ) );
```

- Δημιουργία πίνακα εταιρεία

```
CREATE TABLE `carrental`.`company` (
`name` VARCHAR (45) NOT NULL ,
`phone` VARCHAR (45) NOT NULL,
`address` VARCHAR (45) NOT NULL,
`employee_username` VARCHAR (45),
```

```

`car_plate` VARCHAR (45),
PRIMARY KEY (`name`),
CONSTRAINT `fk_company_employee`
FOREIGN KEY (`employee_username`)
REFERENCES `carrental`.`employee` (`username`)
ON DELETE NO ACTION
ON UPDATE CASCADE,
CONSTRAINT `fk_company_car`
FOREIGN KEY (`car_plate`)
REFERENCES `carrental`.`car` (`plate`)
ON DELETE NO ACTION
ON UPDATE CASCADE );

```

- Δημιουργία πίνακα υπάλληλος

```

CREATE TABLE `carrental`.`employee` (
`username` VARCHAR (45) NOT NULL,
`password` VARCHAR (45) NOT NULL,
`name` VARCHAR (45) NOT NULL ,
`lastname` VARCHAR (45) NOT NULL ,
`phone` VARCHAR (45) NULL,
PRIMARY KEY (`username`));

```

- Δημιουργία πίνακα συνεργείο

```

CREATE TABLE `carrental`.`workshop` (
`name` VARCHAR (45) NOT NULL ,
`phone` VARCHAR (45) NULL,

```

```
`address` VARCHAR (45) NULL,  
PRIMARY KEY (`name`));
```

-Δημιουργία πίνακα επισκευή

```
CREATE TABLE `carrental`.`repair`(  
`idepiskeuis` INT NOT NULL AUTO_INCREMENT ,  
`pdate` DATE,  
`retdate` DATE,  
`workshop_name` VARCHAR (45) NULL,  
`car_plate` VARCHAR (45),  
PRIMARY KEY (`idepiskeuis`),  
CONSTRAINT `fk_repair_workshop`  
FOREIGN KEY (`workshop_name`)  
REFERENCES `carrental`.`workshop` (`name`)  
ON DELETE NO ACTION  
ON UPDATE CASCADE,  
CONSTRAINT `fk_repair_car`  
FOREIGN KEY (`car_plate`)  
REFERENCES `carrental`.`car` (`plate`)  
ON DELETE NO ACTION  
ON UPDATE CASCADE );
```

-Δημιουργία πίνακα συμβόλαιο

```
CREATE TABLE `carrental`.`sumvolaio`(  
`idsumvolaio` INT NOT NULL AUTO_INCREMENT ,  
`payment` VARCHAR (45) NULL,
```

```

`cost` INT NOT NULL,
`car_plate` INT,
`reservation_idres` INT,
PRIMARY KEY (`idsumvolaio`),
INDEX `ixidsumvolaio` (`idsumvolaio`),
CONSTRAINT `fk_sumvolaio_car`
FOREIGN KEY (`car_plate`)
REFERENCES `carrental`.`car` (`plate`)
ON DELETE NO ACTION
ON UPDATE CASCADE ,
CONSTRAINT `fk_sumvolaio_reservation`
FOREIGN KEY (`reservation_idres`)
REFERENCES `carrental`.`reservation` (`idres`)
ON DELETE CASCADE
ON UPDATE CASCADE );

ALTER TABLE sumvolaio AUTO_INCREMENT=1;

```

```

--Δημιουργία πίνακα κράτησης
CREATE TABLE `carrental`.`reservation`(
`idres` INT NOT NULL AUTO_INCREMENT ,
`category` VARCHAR (45)NOT NULL,
`state` VARCHAR (45) NOT NULL,
PRIMARY KEY (`idres`),
INDEX `ixidres` (`idres`));

```

```
ALTER TABLE reservation AUTO_INCREMENT=1;
```

--Δημιουργία πίνακα παράδοση

```
CREATE TABLE `carrental`.`paradosi`(  
  `idparadosi` INT NOT NULL AUTO_INCREMENT ,  
  `place` VARCHAR (45) NOT NULL,  
  `date` DATE NOT NULL,  
  `time` TIME NOT NULL,  
  PRIMARY KEY (`idparadosi`));
```

```
ALTER TABLE paradosi AUTO_INCREMENT=1;
```

-- Δημιουργία πίνακα επιστροφή

```
CREATE TABLE `carrental`.`return`(  
  `idret` INT NOT NULL AUTO_INCREMENT ,  
  `place` VARCHAR (45) NOT NULL,  
  `date` DATE NOT NULL,  
  `time` TIME NOT NULL,  
  PRIMARY KEY (`idret`));
```

```
ALTER TABLE return AUTO_INCREMENT =1;
```

--Ενωση πίνακα κρατησης με πίνακες παραδοση και επιστροφη

```
CREATE TABLE `carrental`.`stkratisi`(  
  `paradosi_idparadosi` INT NOT NULL,  
  `return_idret` INT NOT NULL,
```



```

`reservation_idres` INT NOT NULL,
PRIMARY KEY (`paradosi_idparadosi`,`return_idret`,`reservation_idres`),
CONSTRAINT KEY `fk_stkratisis_paradosi`)
FOREIGN KEY (`paradosi_idparadosi`)
REFERENCES `carrental`.`paradosi`(`idparadosi`)
ON DELETE NO ACTION
ON UPDATE CASCADE,
CONSTRAINT KEY `fk_stkratisis_return`)
FOREIGN KEY (`return_idret`)
REFERENCES `carrental`.`return`(`idret`)
ON DELETE NO ACTION
ON UPDATE CASCADE,
CONSTRAINT KEY `fk_stkratisis_reservation`)
FOREIGN KEY (`reservation_idres`)
REFERENCES `carrental`.`reservation`(`idres`)
ON DELETE NO ACTION
ON UPDATE CASCADE );

```

--Ενωση πινακα συμβολαιου με πινακες παραδοση και επιστροφη

```

CREATE TABLE `carrental`.`stsumvolaio` (
`sumvolaio_idsumvolaio` INT NOT NULL,
`paradosi_idparadosi` INT NOT NULL,
`return_idret` INT NOT NULL,
PRIMARY KEY
(`sumvolaio_idsumvolaio`,`paradosi_idparadosi`,`paradosi_idparadosi`),
CONSTRAINT KEY `fk_stsumvolaio_paradosi` ,

```

```

FOREIGN KEY (`paradosi_idparadosi`)
REFERENCES `carrental`.`paradosi`(`idparadosi`)
ON DELETE NO ACTION
ON UPDATE CASCADE,
CONSTRAINT KEY `fk_stsumvolaio_return`)
FOREIGN KEY (`return_idret`)
REFERENCES `carrental`.`return`(`idret`)
ON DELETE NO ACTION
ON UPDATE CASCADE,
CONSTRAINT KEY `fk_stsumvolaio_sumvolaio`)
FOREIGN KEY (`sumvolaio_idsumvolaio`)
REFERENCES `carrental`.`sumvolaio`(`idsumvolaio`)
ON DELETE NO ACTION
ON UPDATE CASCADE );

```

ΠΑΡΑΡΤΗΜΑ 3 - Κωδικές (Html - Php - Css) διεπαφών πελάτης

Κώδικας Css - Πελάτη

```

body {font-family: "Ariel", sans-serif;}
ul.tab {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    border: 1px solid #ccc;
    background-color: #f1f1f1;
}

```

```
/* Float the list items side by side */
ul.tab li {float: left;}

/* Style the links inside the list items */
ul.tab li a {
    display: inline-block;
    color: black;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    transition: 0.3s;
    font-size: 17px;
}
/* Change background color of links on hover */
ul.tab li a:hover {
    background-color: #ddd;
}
/* Create an active/current tablink class */
ul.tab li a:focus, .active {
    background-color: #ccc;
}
/* Style the tab content */
.tabcontent {
    display: none;
    padding: 6px 12px;
    border: 1px solid #ccc;
}
```

```
border-top: none;
}
body {height:100%;
width:100%;
background-image:url(carphoto.jpg);
background-repeat:no-repeat;
background-size:cover;}
```

Κώδικας Html - Πελάτη

```
<!DOCTYPE html>
<html>
<link rel="stylesheet" type="text/css" href="pelatis.css">
<head> <meta charset="UTF-8"> </head>
<body>
<h1>WELCOME TO BALLOS RENT A CAR </h1>
<ul class="tab">
  <li><a href="javascript:void(0)" class="tablinks" onClick="openCar(event,
'newres')"> <strong>MAKE A RESERVATION </strong></a></li>
  <li><a href="javascript:void(0)" class="tablinks" onclick="openCar(event,
'cancelres')"><b>CANCEL A RESERVATION </b></a></li>
<div id="newres" class="tabcontent">
<form name="res" action="pelatissave.php" method="POST" onsubmit="return
validateForm1()">
<br> <br>
<p> <b>Personal Information </b></p>
First name:
<input type="text" name="firstname" value="">
```

Last name:

```
<input type="text" name="lastname" value="">
```

Email:

```
<input type="text" name="email" value="">
```

Phone:

```
<input type="text" name="phone" value="">
```

```
<br> <br>
```

```
<p> <b>Pick Up Information </b></p> <br>
```

Pick up Location:

```
<select name="pickuplocation">
```

```
<option value="office">Office </option>
```

```
<option value="port">Port</option>
```

```
<option value="airport">Airport</option>
```

```
</select>
```

Date:

```
<input type="date" name="pickupdate" value="">
```

Time:

```
<input type="time" name="pickuptime" value="">
```

```
<br> <br>
```

```
<p> <b>Return Information </b></p> <br>
```

Return Location:

```
<select name="returnlocation">
```

```
<option value="office">Office </option>
```

```
<option value="port">Port</option>
```

```
<option value="airport">Airport</option>
```

```
</select>
```

Date:

```
<input type="date" name="returndate" value="">
```

Time:

```
<input type="time" name="returntime" value="">
```

```
<br> <br>
```

```
<p> <b> Select Category </b> </p>
```

```
<select name="category">
```

```
<option value="a">A (small car) </option>
```

```
<option value="b">B (medium car)</option>
```

```
<option value="j">J (Jeep)</option>
```

```
<option value="e">E (Automatic)</option>
```

```
</select>
```

```
<br><br>
```

```
<input type="submit" name="Submit1" value="Submit" >
```

```
</form>
```

```
</div>
```

```
<div id="cancelres" class="tabcontent" >
```

```
<form name="delete" action="delete.php" method="POST" onsubmit="return  
validateForm2()">
```

```
<br><br>
```

```
<p> <b>Personal Information </b></p>
```

First name:

```
<input type="text" name="firstname" value="">
```

Last name:

```

<input type="text" name="lastname" value="">

<p> <b>Reservation Number </b></p>

Reservation number:

<input type="text" name="resnumber" value="">

<br><br>

<br> <br>

<input type="submit" name="Submit" value="Submit" >

</form>

</div>

</ul>

<script>
function openCar(evt, state) {
    var i, tabcontent, tablinks;

    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }

    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].className = tablinks[i].className.replace(" active", "");
    }

    document.getElementById(state).style.display = "block";
    evt.currentTarget.className += " active";
}
</script>

```

```
<script>
function validateForm1() {
    var x = document.forms["res"]["firstname"].value;
    if (x == "") {
        alert("Please enter your name");
        return false;
    }
    var f = document.forms["res"]["lastname"].value;
    if (f == "") {
        alert("Please enter your last name");
        return false;
    }
    var e = document.forms["res"]["email"].value;
    if (e == "") {
        alert("Please enter your mail");
        return false;
    }
    var c = document.forms["res"]["category"].value;
    if (c == "") {
        alert("Please select a car category");
        return false;
    }
    var pl = document.forms["res"]["pickuplocation"].value;
    if (pl == "") {
        alert("Please select the place you want to receive the car");
    }
}
```



```

        return false;
    }
var pd = document.forms["res"]["pickupdate"].value;
    if (pd == "") {
        alert("Please select the date you want to receive the car");
        return false;
    }
var pt = document.forms["res"]["pickuptime"].value;
    if (pt == "") {
        alert("Please select the time you want to receive the car");
        return false;
    }
var rl = document.forms["res"]["returnlocation"].value;
    if (rl == "") {
        alert("Please select the place you want to return the car");
        return false;
    }
var rd = document.forms["res"]["returndate"].value;
    if (rd == "") {
        alert("Please select the date you want to return the car");
        return false;
    }
var rt = document.forms["res"]["returntime"].value;
    if (rt == "") {
        alert("Please select the time you want to return the car");
        return false;
    }
}

```

```
}  
</script>  
  
<script>  
function validateForm2() {  
    var x = document.forms["delete"]["firstname"].value;  
    if (x == "") {  
        alert("Please enter your name");  
        return false;  
    }  
    var f = document.forms["delete"]["lastname"].value;  
    if (f == "") {  
        alert("Please enter your last name");  
        return false;  
    }  
    var r = document.forms["delete"]["resnumber"].value;  
    if (r == "") {  
        alert("Please enter your reservation number");  
        return false;  
    }  
}  
</script>  
  
</body>  
</html>
```

Κώδικας Php - Αποθήκευση Κράτησης και Αποστολής email

```
<?php
```

```
//Αποθήκευση στοιχείων στη βάση
```

```
$mysqli = mysqli_connect("localhost","fay", "$$1237" , "carrental");
```

```
if(mysqli_connect_errno()) {
```

```
    prin4f ("Connect Fail");
```

```
    exit ();
```

```
}
```

```
else {
```

```
    $add_personalinfo_sql = "INSERT INTO  
costumer(idcostumer,name,lastname,phone,mail)
```

```
    VALUES (NULL,
```

```
        `".$_POST["name"]."`,
```

```
        `".$_POST["lastname"]."`,
```

```
        `".$_POST["phone"]."`,
```

```
        `".$_POST["email"]."`)";
```

```
$res = mysqli_query($mysqli, $add_personalinfo_sql);
```

```
if($res===TRUE) {
```

```
    echo "SUCCESS"; }
```

```
else {
```

```
    echo ("ERROR");
```

```

$add_pickupinfo_sql = "INSERT INTO paradosi(idparadosi,place,date,time)
VALUES ( ".$_POST["pickuplocation"].",
        `".$_POST["pickupdate"].`,
        `".$_POST["pickuptime"].`)";

$res = mysqli_query($mysqli, $add_personalinfo_sql);

if($res===TRUE) {
    echo "Thank you for your reservation"; }
else {
    echo ("Something gone wrong sorry");
}

$add_returninfo_sql = "INSERT INTO return(idret,place,date,time)
VALUES (NULL,`".$_POST["returnlocation"].`,
        `".$_POST["returndate"].`,
        `".$_POST["returntime"].`)";

$res = mysqli_query($mysqli, $add_returninfo_sql)

if($res===TRUE) {
    echo "SUCCESS"; }
else {
    echo ("ERROR");
}

$add_category_sql = "INSERT INTO reservation(category,state)
VALUES ( ".$_POST["category"].",
        `confirm`)";

$res = mysqli_query($mysqli, $add_personalinfo_sql);

if($res===TRUE) {

```

```

        echo "SUCCESS"; }
else {
        echo ("ERROR");
}

// Αποστολή ενημερωτικού mail

$admin_email = "someone@example.com";
$email = $_POST["email"];
$subject = "Confirmation";
$resid = " ..... ";

mail($admin_email, "$subject", "From:" . $email);
echo "Thank you for your reservation. Your reservation number is
", $resid ;
mysqli_close($mysqli);
}
?php>

```

Κώδικας Php - Διαγραφή Κράτησης

```

<?php
$mysqli = mysqli_connect("localhost","fay", "$$1237", "carrental");

if(mysqli_connect_errno()) {
        prin4f ("Connect Fail");
        exit ();
}

```

```

else {
    $sql = "SELECT name, lastname ,reservation_idres FROM costumer
WHERE
        name =`.POST["name"].` AND
        lastname =`.POST["lastname"].` AND
        reservation_idres =`.POST["resnumber"].`";
$result = mysqli_query ($mysqli, $sql) or die(mysqli_error($mysqli));

    if (mysqli_num_rows(result) == 1){
        $sql = "UPDATE `carrental`.`reservation` SET `state` = `canceled`
        WHERE idres =`.POST["resnumber"].`";
        echo "Your reservation is canceled";}
    else {
        echo "Wrong Informations!";}
?>

```

ΠΑΡΑΡΤΗΜΑ 4 - Κωδικές (Html - Php - Css) διεπαφών διαχειριστή

Κώδικας Css - Διαχειριστή

```

@charset "utf-8";

/* CSS Document */

.login {border: 2px solid black;

    height:200px;

    width:300px;

    margin:auto;

    padding:10px;

    background-color: lightgray;

```

```

    }
.login input {
    padding:5px;
    margin:5px
}

.drop_menu {
    background:#005555;
    padding:0;
    margin:0;
    list-style-type:none;
    height:30px;
}

.drop_menu li { float:left; }
.drop_menu li a {
padding:9px 20px;
display:block;
color:#fff;
text-decoration:none;
font:12px arial, verdana, sans-serif;
}

/* Submenu */
.drop_menu ul {
position:absolute;
left:-9999px;
top:-9999px;
list-style-type:none;

```

```

}
.drop_menu li:hover { position:relative; background:#DEB887; }
.drop_menu li:hover ul {
    left:0px;
    top:30px;
    background:#DEB887;
    padding:0px;
}
.drop_menu li:hover ul li a {
padding:5px;
display:block;
width:168px;
text-indent:15px;
background-color:#DEB887;
}
.drop_menu li:hover ul li a:hover { background:#005555; }
.day {border-style: groove;
        background-color:#DEB887;
        vertical-align:middle;
        margin:auto;
        height:320px;
        overflow-y: scroll;
        overflow-x: hidden;}

/*Scroll*/
.result {border-style: groove;
        background-color:#DEB887;

```



```

        vertical-align:middle;

        margin:auto;

        overflow:scroll;

        height:500px;

        overflow-y: scroll;

    overflow-x: hidden;}

/*Telos Scroll*/

a.button {

    -webkit-appearance: button;

    -moz-appearance: button;

    appearance: button;

    text-decoration: none;

    color: initial;

}

.bottom {

    text-align: center;

}

```

Κώδικας Html - Σύνδεσης

```

<!doctype html>

<html>

<link rel="stylesheet" type="text/css" href="menu.css">

<head>

<meta charset="utf-8">

<title>Login in Form</title>

</head>

```

```

<body>
<div class="login">
<h1>Login Form</h1>
<form method="post" action="login.php">
<b> Username: </b> <input type="text" name="username" /> <br>
<b> Password: </b> <input type="text" name="pass" /> <br>
<input type="submit" name="submit" value="Login" />
</form>
</div>
</body>
</html>

```

Κώδικας Php - Σύνδεσης

```

<?php
/*connect to db*/
$mysqli = mysqli_connect("localhost","fay", "$$1237" , "carrental");

if(mysqli_connect_error()) {
    printf ("Connect Fail");
    exit ();
}
else { $sql= "SELECT username, password FROM employee WHERE
    username = '$_POST["username"]' AND
    password = '$_POST["password"]' ";
    $result = mysqli_query($mysqli, $sql) or die (mysqli_error($mysqli));
}
if (mysqli_num_row($result) == 1) {

```

```

echo "Welcome";

echo "<a href='homepage.gr'> Go to Home Page";

}

else { header ("Location: loginform.html"); }

?>

```

Κώδικας Html - Νέα Κράτηση

```

<!doctype html>

<html>

<link rel="stylesheet" type="text/css" href="menu.css" />

<head>

<meta charset="utf-8">

<title>HOME</title>

</head>

<body>

<table style="width:100%">

<tr>

<div class="drop">

<ul class="drop_menu">

<li><a href="home.html">HOME</a> </li>

<li><a href="#">ADD</a>

<ul>

<li><a href="reservation.html">RESERVATION</a></li>

<li><a href="contract.html">CONTRACT</a></li>

```

```
<li><a href="car.html">CAR</a></li>
```

```
</ul>
```

```
</li>
```

```
<li><a href="#">CALENDER</a></li>
```

```
</ul>
```

```
</div>
```

```
</tr>
```

```
<div class="date">
```

```
<br>
```

```
</div>
```

```
<tr>
```

```
<div class="result">
```

```
<form action="adres.php" method="POST">
```

```
<b>Reservation id:</b>
```

```
<input type="text" name="name" value="auto">
```

```
<br><br>
```

```
<b>DRIVER INFORMATION </b>
```

```
<br>
```

```
<b>Name: </b>
```

```
<input type="text" name="name" value="">
```

```
<b>Last Name: </b>
```

```
<input type="text" name="lastname" value="">
```

```
<b>Birthday: </b><input type="date" name="birthday" value=""> <br><br>
```

```
<br>
```

PICK UP INFORMATION

 Pick Up Location:

<select name="pickuplocation">

<option value="office">Office </option>

<option value="port">Port</option>

<option value="airport">Airport</option>

</select>

Date:

<input type="date" name="pickupndate" value="">

Time:

<input type="time" name="pickuptime" value="">

RETURN INFORMATION

Return Location:

<select name="returnplace">

<option value="office">Office </option>

<option value="port">Port</option>

<option value="airport">Airport</option>

</select>

Date:

<input type="date" name="returndate" value="">

Time:

<input type="time" name="returntime" value="">

CAR INFORMATION


```

<b> Plate: </b>
  <input type="text" name="plate" value="">
<br> <br>
  <input type="submit" value="SAVE">
</form>
</div>
</tr>
<th>
<input type="button" onclick="location.href='http://loginform.html';" value="Change
User" />
</th>
</table>
</body>
</html>

```

Κώδικας Php - Νέα Κράτηση

```

<?php
$mysqli = mysqli_connect("localhost","fay", "$$1237" , "carrental");

if(mysqli_connect_errno()) {
    prin4f ("Connect Fail");
    exit ();
}
else {
    $add_costumerinfo_sql = "INSERT INTO
costumer(name,lastname,birthday)

```

```

VALUES ( `".$_POST["name"]."`,
        `".$_POST["lastname"]."`,
        `".$_POST["birthday"]."");

$costumer = mysqli_query($mysqli, $add_costumerinfo_sql);

        $add_pickupinfo_sql = "INSERT INTO
paradosi(place,date,time)

VALUES (

        `".$_POST["pickupplace"]."`,

        `".$_POST["pickupdate"]."`,
        `".$_POST["pickuptime"]."");

$pickup = mysqli_query($mysqli, $add_pickupinfo_sql);

        $add_returninfo_sql = "INSERT INTO
return(place,date,time)

VALUES (

        `".$_POST["returnplace"]."`,

        `".$_POST["returndate"]."`,
        `".$_POST["returntime"]."");

$return = mysqli_query($mysqli, $add_returninfo_sql);
}
mysqli_close($mysqli);
?>

```

Κώδικας Html - Αυτοκίνητο

```

<!doctype html>

<html>

<link rel="stylesheet" type="text/css" href="menu.css" />

```

```

<head>
<meta charset="utf-8">
<title>HOME</title>
</head>
<body>
<table style="width:100%">
<tr>
<div class="drop">
<ul class="drop_menu">
<li><a href="home.html">HOME</a> </li>

<li>ADD
<ul>
<li><a href="reservation.html">RESERVATION</a></li>
<li><a href="contract.html">CONTRACT</a></li>
<li><a href="car.html">CAR</a></li>
</ul>
</li>
<li><a href="#">VIEW CARS</a></li>
</ul>
</div>
</tr>
<tr>
<div class="date">
<br>
<form action="date.php" method="POST"> <b>SELECT DATE </b><input

```



```

type="date" name="resdate" value="">
<input type="submit" name="Submit" value="OK" >
</form>
</div>
<div class="result">
<form action="addcar.php" method="POST">
<h3>Car Information</h3>
Plate:
  <input type="text" name="plate" value="">
  Brand: <input type="text" name="brand" value=""> <br><br>
Miles:
  <input type="int" name="miles" value="">
Color:
  <input type="text" name="color" value="">
<br>
  <br>
<b> State </b>
<select name="category">
  <option value="typea">Free </option>
  <option value="typeb">Rental</option>
  <option value="typeb">Out Of Order</option>
</select>
<br>
<br>
  <input type="submit" value="SAVE">
</form>

```

```

</div>
</tr>
<th>
<input type="button" onclick="location.href='http://loginform.html';" value="Change
User" />
</th>
</table>
</body>
</html>

```

Κώδικας Php - Αυτοκίνητο

```

<?php
$mysqli = mysqli_connect("localhost","fay", "$$1237" , "carrental");

if(mysqli_connect_errno()) {
    prin4f ("Connect Fail");
    exit ();
}
else {
    $add_carinfo_sql = "INSERT INTO
car(plate,category,brand,state,colour,km)
VALUES (
`".$_POST["plate"]."`,
`".$_POST["category"]."`,
`".$_POST["brand"]."`,
`".$_POST["state"]."`,
`".$_POST["colour"]."`,
`".$_POST["km"]."`)";

```



```
<li><a href="contract.html">CONTRACT</a></li>
<li><a href="car.html">CAR</a></li>
</ul>
</li>
<li><a href="#">CALENDER</a></li>
</ul>
</div>
</tr>
<tr>
<div class="date">
<br>

</div>
<div class="result">
<form action="contract.php" method="POST">
<b>Contract id:</b>
<input type="text" name="name" value="auto">
<br><br>
<b>DRIVER INFORMATION </b>
<br>
<b>Name: </b>
  <input type="text" name="name" value="">
<b>Last Name: </b>
  <input type="text" name="lastname" value="">
<b>Birthday: </b><input type="date" name="lastname" value=""> <br><br>
<b>Licence Number: </b>
```

```
<input type="text" name="licence number" value="">
<b>Licence Expiry Date:</b>
<input type="date" name="expirydate" value="">
<b>Country:</b>
<input type="text" name="licencecountry" value="">
<br> <br>
```

```
<b>PICK UP INFORMATION </b><br>
<b> Pick Up Location: </b>
<select name="pickuplocation">
  <option value="office">Office </option>
  <option value="port">Port</option>
  <option value="airport">Airport</option>
</select>
```

```
<b>Date: </b>
<input type="date" name="returndate" value="">
<b>Time: </b>
<input type="time" name="returntime" value="">
<br> <br>
```

```
<b>RETURN INFORMATION </b> <br>
<b>Return Location: </b>
<select name="returnlocation">
  <option value="office">Office </option>
  <option value="port">Port</option>
  <option value="airport">Airport</option>
```

</select>

Date:

<input type="date" name="returndate" value="">

Time:

<input type="time" name="returntime" value="">

CAR INFORMATION

 Plate:

<input type="text" name="plate" value="">

Days: <input type="text" name="days" value="">

Cost Per Day:<input type="text" name="costperday" value="">

Total Cost: <input type="text" name="totalcost" value="">

 Payment State

<select name="category">

<option value="typea">Pending </option>

<option value="typeb">Paid</option>

</select>

CARD INFORMATION

 Owner:

<input type="text" name="owner" value="">

Card Number:

<input type="text" name="cardnumber" value="">

```

<b>Expire Date: </b>
  <input type="date" name="expdate" value="">
<br> <br>
  <input type="submit" value="SAVE">
</form>
</div>
</tr>
<th>
<input type="button" onclick="location.href='http://loginform.html';" value="Change
User" />
</th>
</table>
</body>
</html>

```

Κώδικας Php - Συμβόλαιο

```

<?php
$mysqli = mysqli_connect("localhost","fay", "$$1237" , "carrental");

if(mysqli_connect_erno()) {
    printf ("Connect Fail");
    exit ();
}
else {
    $add_sumvolaio_sql = "INSERT INTO
sumvolaio(idsumvolaio, payment,cost,carplate)

```

```

VALUES ( `".$_POST["payment"]."`,
        `".$_POST["totalcost"]."`,
        `".$_POST["carplate"]."");

$sumvolaio = mysqli_query($mysqli, $add_sumvolaio_sql);

$update_car_sql = "UPDATE car
SET state= 'rented'
WHERE plate = `".$_POST["plate"]." `)";

$car = mysqli_query($mysqli, $update_car_sql);

$add_costumerinfo_sql = "INSERT INTO costumer(idcostumer,
name,lastname,birthday,card_owner,licence_owner)
VALUES ( `".$_POST["name"]."`,
        `".$_POST["lastname"]."`,
        `".$_POST["birthday"]."`,
        `".$_POST["lastname"]."`,
        `".$_POST["lastname"]."");

$costumer = mysqli_query($mysqli, $add_costumerlinfo_sql);

$add_pickupinfo_sql = "INSERT INTO paradosi(place,date,time)
VALUES (
        `".$_POST["pickupplace"]."`,
        `".$_POST["pickupdate"]."`,
        `".$_POST["pickuptime"]."");

$pickup = mysqli_query($mysqli, $add_pickupinfo_sql);

$add_returninfo_sql = "INSERT INTO return(place,date,time)

```



```

VALUES (
    `".$_POST["returnplace"]."`,
    `".$_POST["returndate"]."`,
    `".$_POST["returntime"]."");
$return = mysqli_query($mysqli, $add_returninfo_sql);

$add_licence_sql = "INSERT INTO licence (owner,number,date,country)
VALUES (
    `".$_POST["lastname"]."`,
    `".$_POST["licencenumber"]."`,
    `".$_POST["expirydate"]."`,
    `".$_POST["licencecountry"]."");
$licence = mysqli_query($mysqli, $add_licence_sql);

$add_card_sql = "INSERT INTO return(place,date,time)
VALUES (
    `".$_POST["returnplace"]."`,
    `".$_POST["returndate"]."`,
    `".$_POST["returntime"]."");
$card = mysqli_query($mysqli, $add_card_sql);
}
mysqli_close($mysqli);
?>

```