Ομάδα 11: Εφαρμογή ενοικίασης αυτοκινήτων και διαχείρισης κρατήσεων

Ειρήνη Καλαϊτζή (ΑΜ: 1063960), Νικόλαος Πράσινος (AM:1059382)

Πανεπιστήμιο Πατρών, [up1063960@upnet.gr](mailto:up1063960@upnet.gr), [up1059382@upnet.gr](mailto:up1059382@upnet.gr)

Η παρούσα εργασία στο μάθημα Βάσεις Δεδομένων πραγματεύεται τη σχεδίαση και την υλοποίηση μιας εφαρμογής ενοικιάσεων αυτοκινήτων και διαχείρισης κρατήσεων. Παρακάτω παρουσιάζονται ο σχεδιασμός της βάσης δεδομένων που χρειάστηκε για τη λειτουργία της η εφαρμογή, καθώς και οι γραφικές διεπαφές και οι λειτουργίες τους.

1. εισαγωγη

Στόχος της ομαδικής εργασίας μας ήταν η δημιουργία μιας εφαρμογής ενοικίασης αυτοκινήτων και διαχείρισης online κρατήσεων.

Για να δουλέψουμε, δημιουργήσαμε μια βάση δεδομένων βασισμένη στο μικρόκοσμο μιας εταιρείας ενοικίασης αυτοκινήτων και δύο ξεχωριστές εφαρμογές για τη διαχείριση των δεδομένων της εταιρείας.

 Η εφαρμογή πελατών επιτρέπει:

* Εκτέλεση κράτησης αυτοκινήτου
* Ακύρωση κράτησης
* Δήλωση επιστροφής αυτοκινήτου
* Βαθμολογία και Σχολιασμός κράτησης

Η εφαρμογή του διαχειριστή  (admin) του επιτρέπει:

* Αναζήτηση αυτοκινήτου βάσει τοποθεσίας, μοντέλου, ή πινακίδων
* Αναζήτηση κράτησης και δυνατότητα προσθήκης ζημιών στο συνολικό κόστος
* Προβολή στατιστικών στοιχείων σχετικά με το πλήθος των κρατήσεων και τα κέρδη
* Προβολή της μέσης βαθμολογίας της εφαρμογής πελάτη καθώς επίσης και τον σχολίων που άφησαν οι πελάτες

Για τον κώδικα των εφαρμογών και τη δημιουργία της βάσης χρησιμοποιήσαμε γλώσσα Python και τη βιβλιοθήκη sqlite3.

1. μεθοδολογια

Εργαστήκαμε με τον εξής τρόπο:

Αρχικά, αναζητήσαμε πληροφορίες για τον τρόπο λειτουργίας ενοικίασης αυτοκινήτων στο διαδίκτυο. Για αυτό συμβουλευτήκαμε γνωστές ιστοσελίδες, όπως η Avis [1] και η Car and Motion [2] και προσπαθήσαμε να καταλάβουμε τον μικρόκοσμο και το σχεδιασμό της βάσης δεδομένων τους. Επιπλέον επειδή κανένα μέλος της ομάδας δεν είχε ιδιαίτερη εμπειρία με ενοικιάσεις αυτοκινήτων, πήραμε μερικές πληροφορίες πάνω στις διαθέσιμες υπηρεσίες που προσφέρουν τέτοιες επιχειρήσεις, μέσω γνωστών που εργάζονται σε αυτό το τομέα. Έτσι αποφασίσαμε τα χαρακτηριστικά που θέλουμε να έχει ο μικρόκοσμος της εταιρείας μας, αλλά και τις λειτουργίες που θα διαθέτουν οι εφαρμογές μας.

Στη συνέχεια δημιουργήσαμε το μοντέλο συσχετίσεων-οντοτήτων (ERD) βάση του μικρόκοσμου που ετοιμάσαμε και μετά το μετατρέψαμε σε σχεσιακό διάγραμμα. Έτσι ήμασταν έτοιμοι να συντάξουμε τη βάση δεδομένων μας. Συντάξαμε κώδικα Python, που δημιουργεί τη βάση δεδομένων και την αποθηκεύει σε ένα αρχείο db, στο οποίο προσθέσαμε και την εισαγωγή δεδομένων στη βάση μας.

Έχοντας φτιάξει την βάση δεδομένων, έπρεπε να φτιάξουμε μια γραφική διεπαφή, με την οποία θα μπορούσε να αλληλοεπιδράσει ο πελάτης και να δει τα αποτελέσματα των πράξεων του. Όμως επειδή ορισμένες λειτουργίες, δεν είχε νόημα να μπορεί να τις κάνει ο πελάτης, στα πλαίσια του μικρόκοσμου, φτιάξαμε 2 ξεχωριστές εφαρμογές.

Η μία είναι η εφαρμογή που χρησιμοποιεί ο πελάτης, για να  να κάνει κρατήσεις αυτοκινήτων, να ακυρώσει τις ήδη υπάρχουσες, να ειδοποιήσει για την επιστροφή και να αξιολογήσει την κράτηση του.

Η άλλη είναι η εφαρμογή που χρησιμοποιεί ο διαχειριστής (administrator) των κρατήσεων, ο οποίος θα είναι υπάλληλος της εταιρείας ενοικίασης και θα του επιτρέπει να αναζητά ανά πάσα στιγμή κρατήσεις και αυτοκίνητα, να βλέπει στατιστικά στοιχεία, να προσθέτει τυχόν ζημιές στο πληρωτέο ποσό όταν ένα αυτοκίνητο έχει επιστραφεί αλλά και να παρατηρεί σχόλια και αξιολογήσεις που αφήνουν οι πελάτες.

**Καταμερισμός εργασίας**

Ειρήνη Καλαϊτζή: ERD, Relational Schema, Δημιουργία βάσης δεδομένων, Συλλογή και εισαγωγή δεδομένων στην βάση, Client GUI, Make a Reservation (Client), Cancelation (Client)

Νικόλας Πράσινος: ERD, Admin GUI, Search Car (Admin), Search Reservation (Admin), Return car (Client), Statistics (Admin), Ratings (Admin), Rate Reservation (Client)

**Χρονοδιάγραμμα:**

1η βδομάδα: Αναζήτηση πληροφοριών

2η βδομάδα: Διάγραμμα Συσχετίσεων - Οντοτήτων

3η βδομάδα: Διάγραμμα Συσχετίσεων – Οντοτήτων

4η βδομάδα: Σχεσιακό Μοντέλο

5η βδομάδα: Γραφική διεπαφή πελάτη

6η βδομάδα: Ένωση της γραφική διεπαφής του πελάτη με τα SQL queries που ετοιμάσαμε

7η βδομάδα: Επιπρόσθετες λειτουργίες πελάτη & γραφική διεπαφή διαχειριστή

8η βδομάδα: Ένωση της γραφική διεπαφής του διαχειριστή με τα SQL queries που ετοιμάσαμε

9η βδομάδα: Επιπρόσθετες λειτουργίες πελάτη & επιπρόσθετες λειτουργίες διαχειριστή

10η βδομάδα: Επιπρόσθετες λειτουργίες πελάτη & επιπρόσθετες λειτουργίες διαχειριστή

11η βδομάδα: Συγγραφή αναφοράς & debugging

1. περιγραφη βασησ δεδομενων

Για τον σχεδιασμό της βάσης δεδομένων χρειάστηκε σαφής ανάλυση και κατανόηση του μικρόκοσμου ενός online ενοικιαστήριου αυτοκινήτων. Μόλις επιτεύχθηκε αυτό, μπορέσαμε και κάναμε το διάγραμμα οντοτήτων – συσχετίσεων, το οποίο μετατρέψαμε σε σχεσιακό μοντέλο και στη συνέχεια μέσω SQL δημιουργήσαμε τη βάση δεδομένων.

* 1. Μικρόκοσμος

Έχουμε μια εταιρεία ενοικίασης αυτοκινήτων, που διαθέτει παραρτήματα σε διάφορες πόλεις της Ελλάδας και επιτρέπει στον χρήστη την παραλαβή οχήματος και την επιστροφή του σε  κάποιο από αυτά τα παραρτήματα. Η εταιρεία μας διαθέτει οχήματα 3 κατηγοριών: Μικρά αμάξια πόλης, SUV και Jeep. Μαζί με την κράτηση προσφέρουμε 3 διαφορετικά ασφαλιστικά πακέτα το καθένα με διαφορετική τιμή, καθώς και τέσσερις έξτρα υπηρεσίες. Οι κρατήσεις οχημάτων και η διαχείρισή τους γίνονται μέσω δύο εφαρμογών. Τη μια την χρησιμοποιεί ο πελάτης και την άλλη ένας υπάλληλος της εταιρείας.

Ο πελάτης μπαίνει στην εφαρμογή ενοικίασης αυτοκινήτων για να εκτελέσει μια κράτηση. Διαλέγει τοποθεσίες και ημερομηνίες παραλαβής και επιστροφής οχήματος. Επιλέγει την κατηγορία (μέγεθος) αυτοκινήτων που θέλει να νοικιάσει, εφόσον υπάρχουν διαθέσιμα αυτοκίνητα. Κάθε κατηγορία έχει διαφορετική χρέωση ανά ημέρα ενοικίασης. Κατόπιν επιλέγει ασφαλιστικό πακέτο, που αφορά το ποσοστό κάλυψης πιθανών ζημιών και την περίπτωση κλοπής. Επιλέγει ένα εκ των τριών διαθέσιμων πακέτων: Silver, Golden, Platinum. Ακόμα έχει τη δυνατότητα να επιλέξει συμπεριλάβει στην κράτηση του έξτρα υπηρεσίες. Συγκεκριμένα μπορεί να επιλέξει ανάμεσα σε έξι υπηρεσίες: GPS, κάθισμα βρέφους, τηλεφωνική εξυπηρέτηση και επιπλέον οδηγό**.**

Στο τέλος συμπληρώνει τα στοιχεία του και προχωρά στην πληρωμή που γίνεται με πιστωτική ή χρεωστική κάρτα. Η κάρτα δεσμεύεται μέχρι την ολοκλήρωση της κράτησης, όπου και ελέγχεται αν υπάρχουν επιπλέον χρεώσεις (π.χ ζημιές).

Ο πελάτης έχει ακόμη τη δυνατότητα να ακυρώσει την κράτησή του, χρησιμοποιώντας τον αριθμό κράτησης και το email του. Αν ακυρώσει εντός της προθεσμίας (10 ημέρες πριν την έναρξη της κράτησης), του επιστρέφεται ολόκληρο το ποσό, ενώ αν καθυστερήσει του επιστρέφεται το 50% του συνολικού κόστους κράτησης.

Η λήξη της κράτησης γίνεται με την επιστροφή του οχήματος στη τοποθεσία που έχει δηλώσει ότι θα το επέστρεφε. Ο χρήστης ενημερώνει για την παράδοση του οχήματος μέσω της εφαρμογής, όπου και ελέγχεται αν παρέδωσε το όχημα καθυστερημένα και χρεώνεται επιπλέον αν χρειάζεται. Τότε το συνολικό κόστος ανανεώνεται κατάλληλα.

Τότε ο χρήστης έχει τη δυνατότητα να αφήσει μια αξιολόγηση για την κράτησή του (μέγιστος βαθμός ικανοποίησης 5) καθώς και να κάνει σχόλια αν θέλει.

Από την εφαρμογή διαχείρισης κρατήσεων της εταιρείας ο admin μπορεί να δει πληροφορίες για κάθε αμάξι και για κάθε κράτηση. Συγκεκριμένα ξέρει που βρίσκεται κάθε αμάξι όταν δεν βρίσκονται σε κράτηση, ενώ αν βρίσκονται σε κράτηση γνωρίζει ποιος πελάτης την έχει κάνει. Επιπλέον μπορεί να εισάγει επιπλέον χρεώσεις στο πληρωτέο ποσό λόγω ζημιάς του οχήματος, μόλις ένα αμάξι έχει επιστραφεί.

Ακόμη ο διαχειριστής έχει τη δυνατότητα να δει όλα τα σχόλια, που έχουν αφήσει οι χρήστες καθώς επίσης και να δει τον μέσο όρο των βαθμολογιών που έχουν αφήσει για τις κρατήσεις τους. Τέλος ο διαχειριστής μπορεί να παρακολουθήσει ορισμένα στατιστικά στοιχεία, όπως να δει ποιους μήνες έχει τις περισσότερες κρατήσεις και τα περισσότερα κέρδη αλλά και πόσες κρατήσεις έχει ανά κατηγορία.

* 1. Διάγραμμα Οντοτήτων – Συσχετίσεων

Με βάση την παραπάνω περιγραφή του μικρόκοσμου κατασκευάσαμε το ακόλουθο μοντέλο οντοτήτων-συσχετίσεων:

Diagram

Description automatically generated

Διάγραμμα Οντοτήτων - Συσχετίσεων

Το μοντέλο που κατασκευάσαμε αποτελείται από 9 οντότητες, την κράτηση, τον πελάτη, το αυτοκίνητο, την κατηγορία, την πληρωμή, το ασφαλιστικό πακέτο, τις υπηρεσίες και την αξιολόγηση.

Βασική οντότητα είναι η κράτηση, η οποία συσχετίζεται με όλες τις υπόλοιπες οντότητες. Το συνολικό κόστος της κράτησης υπολογίζεται από την κατηγορία οχήματος, τις ημέρες, την ασφάλιση, τις υπηρεσίες και τις ζημιές.

Η κράτηση γίνεται από τον πελάτη. Ο πελάτης κατά την κράτηση επιλέγει κατηγορία αυτοκινήτου, μοντέλο οχήματος, τοποθεσία παραλαβής κι επιστροφής, ασφαλιστικό πακέτο και υπηρεσίες.

Τα στοιχεία του και η πιστωτική κάρτα του αποθηκεύονται και η πληρωμή ολοκληρώνεται με την επιστροφή του οχήματος. Τέλος, ο πελάτης έχει την δυνατότητα να αξιολογήσει την κράτησή του.

Η πληρωμή και η αξιολόγηση εξαρτώνται αποκλειστικά από την ύπαρξη της εκάστοτε κράτησης, γι’ αυτό και είναι ασθενείς οντότητες.

Σημαντικό ρόλο διαδραματίζει η τοποθεσία ενός οχήματος καθώς πρέπει ανά πάσα στιγμή να γνωρίζουμε που βρίσκεται. Έτσι κάθε αυτοκίνητο έχει μια τοποθεσία και σε κάθε κράτηση σημειώνεται η τοποθεσία ενός οχήματος όταν παραλαμβάνεται, η οποία αλλάζει όταν το αυτοκίνητο επιστρέφεται.

**Παραδοχές**

* Κάθε πελάτης στο σύστημα έχει τουλάχιστον μία κράτηση.
* Κάθε κράτηση έχει απαραίτητα μία πληρωμή, μία κατηγορία, ένα αυτοκίνητο και ένα ασφαλιστικό πακέτο.
* Μπορεί ακόμα να έχει μία ή καμία αξιολόγηση και καμία ή πολλές επιπλέον υπηρεσίες.
* Η τοποθεσία αναφέρεται στα παραρτήματα που έχει η εταιρεία, τα οποία είναι τα διαθέσιμα μέρη από τα οποία μπορεί ο πελάτης να επιλέξει να παραλάβει και να επιστρέψει το αυτοκίνητο. Η τοποθεσία παραλαβής κι επιστροφής δε χρειάζεται να είναι οι ίδιες.
  1. Σχεσιακό Μοντέλο

Με βάση το ERD κατασκευάσαμε το λογικό σχεσιακό μοντέλο:

Diagram

Description automatically generated

Σχεσιακό Μοντέλο

**Primary Keys**

Τα βασικά κλειδιά κάθε πίνακα είναι πρώτα και ονομάζονται με το όνομα του πίνακα και ID εκτός από τον πίνακα του πελάτη, του οποίο το βασικό κλειδί είναι το ΑΦΜ του.

**Foreign Keys**

Η κράτηση περιέχει τα ξένα κλειδιά που την συνδέουν με τον πελάτη (Client\_ID), το ασφαλιστικό πακέτο (Ins\_ID), την κατηγορία (Cat\_ID), το αυτοκίνητο (C\_ID), την τοποθεσία παραλαβής (PUL\_ID) και την τοποθεσία επιστροφής (DOL\_ID).

O πίνακας CONTAINS αποτελείται μόνο από 2 ξένα κλειδιά και συνδέει τις πολλές υπηρεσίες (S\_ID) που μπορεί να έχει επιλέξει ο πελάτης με την κάθε κράτηση(R\_ID).

H αξιολόγηση περιέχει ένα ξένο κλειδί που τη συνδέει με την κράτηση (R\_ID).

Το αντίστοιχο συμβαίνει και με την πληρωμή (R\_ID).

* 1. SQL βάση δεδομένων

Για να κατασκευάσουμε τη βάση δεδομένων χρησιμοποιήσαμε τη βιβλιοθήκη sqlite3.

Ξεκινάμε εισάγοντας την sqlite3:



Συνδεόμαστε στη βάση δεδομένων δημιουργώντας ένα αντικείμενο με την εντολή connect(). Αν το αρχείο αυτό δεν υπάρχει, δημιουργείται αυτόματα από την sqlite3.



Δημιουργούμε ένα κέρσορα:



Κατασκευάζουμε τους πίνακες που φτιάξαμε στο σχεσιακό μοντέλο με την παρακάτω διαδικασία:



Η εντολή execute() είναι μέθοδος του κέρσορα και χρησιμοποιείται για όλα τα SQL queries.

Μέσα στις παρενθέσεις τοποθετούνται αριστερά τα ονόματα των ιδιοτήτων του πίνακα και δίπλα η μορφή που θα πρέπει να έχουν τα δεδομένα.

Χρησιμοποιήσαμε τους ακόλουθους τύπους δεδομένων:

INTEGER, για όλους τους ακέραιους αριθμούς, όπως είναι τα κλειδιά

TEXT, για όλα τα αλφαριθμητικά δεδομένα

VARCHAR, για δεδομένα που περιέχουν διαφορετικούς χαρακτήρες όπως το email

DATE, για όλες τις ημερομηνίες

DECIMAL, για τους δεκαδικούς αριθμούς.

Στο τέλος τοποθετούνται τυχόν περιορισμοί. Χρησιμοποιήσαμε τον περιορισμό NOT NULL για όλες τις τιμές που θέλαμε οπωσδήποτε να αποθηκεύονται σε κάθε νέα καταγραφή δεδομένων και DEFAULT 0 για τις τιμές που θέλαμε να αλλάζουν υπό συνθήκη, όπως η τιμή damages.

Το βασικό κλειδί του κάθε πίνακα που είναι και το αναγνωριστικό του, το ορίζουμε ξεχωριστά μετά τη δήλωση των μεταβλητών όπως φαίνεται παραπάνω. Κάθε πίνακας έχει μόνο ένα βασικό κλειδί, το οποίο επίσης πρέπει να δηλωθεί ως NOT NULL.

Τα ξένα κλειδιά δημιουργούνται με παρόμοιο τρόπο και δηλώνονται στο τέλος προσθέτοντας τον πίνακα στον οποίο αντιστοιχούν:



Τοποθετήσαμε τα ξένα κλειδιά στους πίνακες όπως φαίνεται στο σχεσιακό μοντέλο.

Με βάση τα προηγούμενα, κατασκευάσαμε και τους υπόλοιπους πίνακες της βάσης.

Τέλος για να αποθηκεύσουμε όλες τις αλλαγές που έγιναν στη βάση χρησιμοποιούμε την εντολή:



Ενώ όταν δεν χρειαζόμαστε πια τη βάση, κλείνουμε τη σύνδεση με την ακόλουθη εντολή:



* 1. Δεδομένα βάσης

Χρειάζεται ακόμα να εισάγουμε στη βάση τα δεδομένα που έχει ο μικρόκοσμός μας.

Σκεφτήκαμε ότι σε ένα πραγματικό σενάριο, μια εταιρεία ενοικίασης αυτοκινήτων θα αγόραζε αυτοκίνητα, τα οποία θα ήταν συγχρόνως οικονομικά και εύκολα στη συντήρηση. Για αυτό το λόγο συμβουλευτήκαμε ορισμένα άρθρα [3], τα οποία είχαν τα αμάξια που πληρούσαν αυτές τις προϋποθέσεις σε λίστες. Αυτά τα οχήματα χρησιμοποιήσαμε ως μοντέλα στην βάση δεδομένων μας. Αυτά τα μοντέλα ανήκουν σε ορισμένες κατηγορίες οχημάτων, οι οποίες έχουν τη δική τους χρέωση ανά μέρα. Στην βάση μας καταλήξαμε σε τρείς κατηγορίες αυτοκινήτων:

* City με 20 ευρώ/μέρα
* SUV με 30 ευρώ/μέρα
* Jeep με 50 ευρώ/μέρα

Όσον αφορά τα δεδομένα για τις τοποθεσίες, τα ασφαλιστικά πακέτα και τις επιπλέον υπηρεσίες, αυτά συλλέχθηκαν δειγματοληπτικά από περιήγηση σε ιστοσελίδες ενοικίασης αυτοκινήτων, όπως η Avis και η Car and Motion [1,2].

Έτσι καταλήξαμε στα παρακάτω ασφαλιστικά πακέτα:

* Silver με 20 επιπλέον ευρώ
* Golden με 30 επιπλέον ευρώ
* Platinum με 40 επιπλέον ευρώ

Το πρώτο πακέτο καλύπτει μόνο τη βασική ασφάλεια, το 2ο καλύπτει το 50% των πιθανών ζημιών και το 3ο καλύπτει ολόκληρο το ποσό και σε περίπτωση κλοπής.

Για τις επιπλέον παρεχόμενες υπηρεσίες καταλήξαμε στις παρακάτω:

* GPS με επιπλέον 7 ευρώ
* Κάθισμα βρέφους με επιπλέον 12 ευρώ
* Επιπλέον οδηγός με επιπλέον 10 ευρώ
* Τηλεφωνική υποστήριξη με επιπλέον 3 ευρώ

Η εισαγωγή δεδομένων στους πίνακες γίνεται με τον εξής τρόπο:



Με αυτόν τον τρόπο τοποθετούνται όλες οι τιμές του πίνακα ανά εισαγωγή.

Ακόμα, αν θέλουμε να τοποθετήσουμε συγκεκριμένα δεδομένα στον πίνακα χρησιμοποιούμε την εντολή:



Για να τοποθετήσουμε τα υπόλοιπα δεδομένα σε όλους τους πίνακες χρησιμοποιούμε την εντολή INSERT INTO table, όπου table είναι το αντίστοιχο όνομα του πίνακα.













1. Εφαρμογη πελατη
   1. Περιγραφή λειτουργίας

Η εφαρμογή ενοικίασης αυτοκινήτων που προορίζεται για χρήση από τον πελάτη, έχει τις εξής λειτουργίες:

1) Δημιουργία Κράτησης

2) Ακύρωση Κράτησης

3) Επιστροφή Οχήματος

4) Αξιολόγηση Κράτησης

Η διεπαφή αποτελείται από ένα αρχικό παράθυρο με 4 κουμπιά, καθένα από το οποίο επιτελεί τις παραπάνω διαδικασίες.

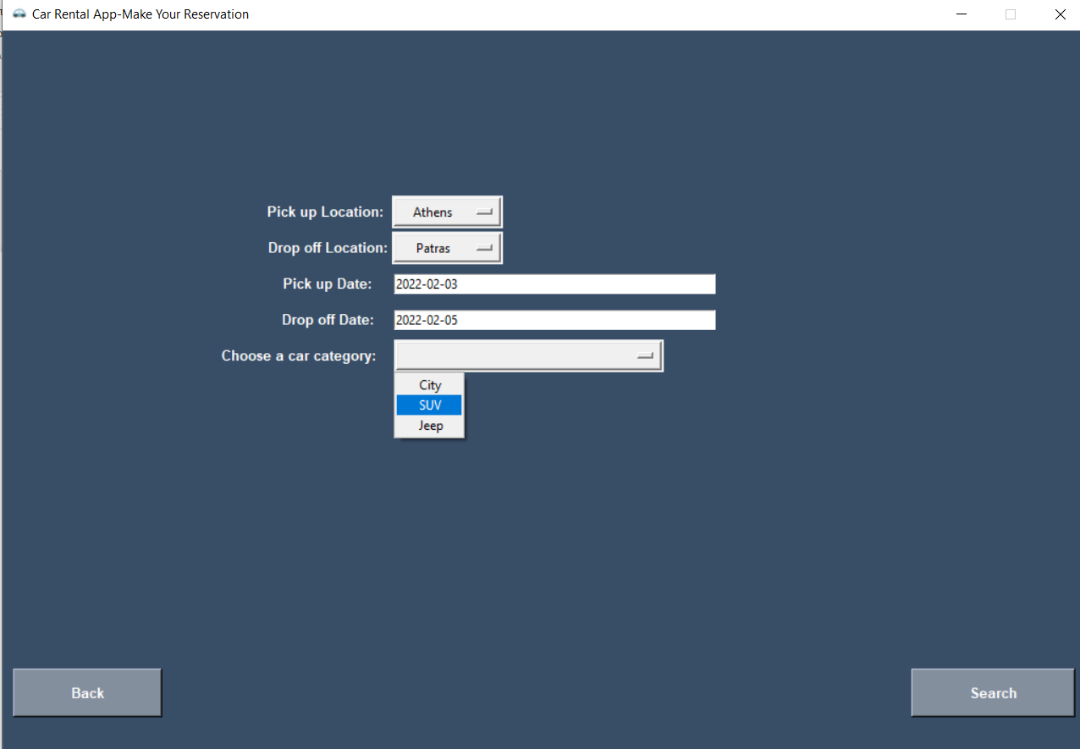
Εικόνα που περιέχει κείμενο, χλόη, υπαίθριος, αυτοκίνητο

Περιγραφή που δημιουργήθηκε αυτόματα

Homepage της εφαρμογή του πελάτη

* 1. Λειτουργίες
     1. Δημιουργία κράτησης

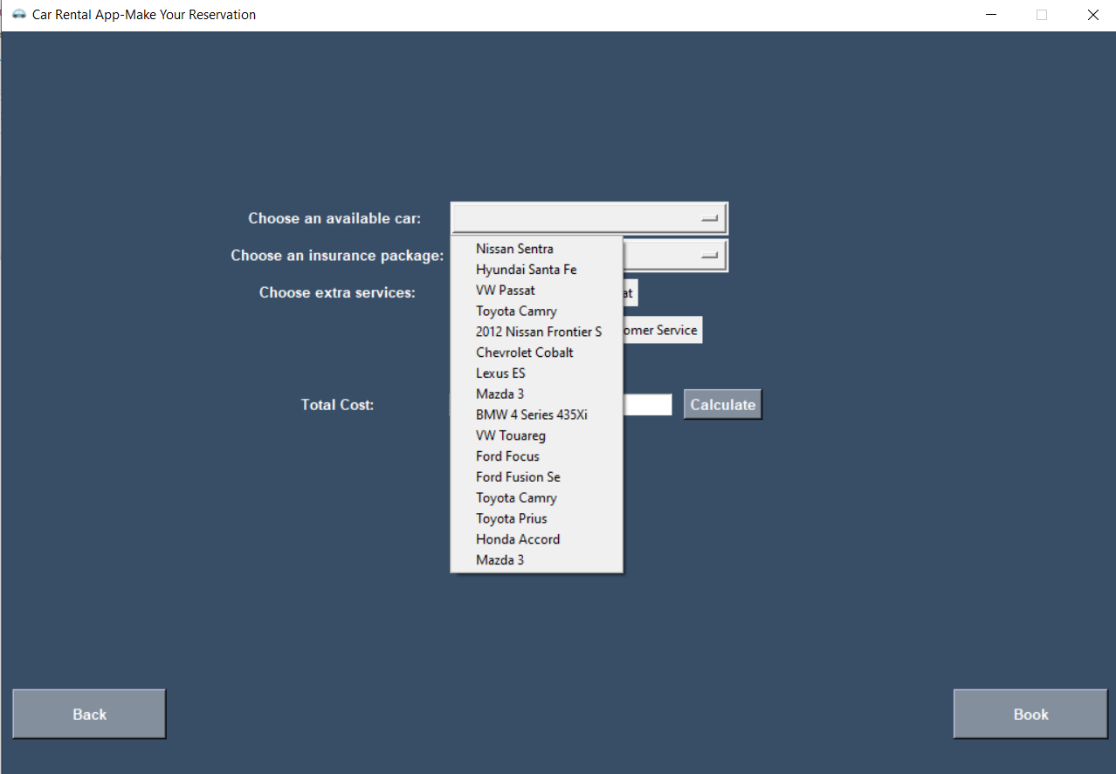
Πατώντας αυτό το κουμπί ο πελάτης μεταβαίνει σε άλλη σελίδα, η οποία αποτελείται από λεζάντες και μενού επιλογής για τα βασικά στοιχεία της κράτησης. Σε αυτή τη σελίδα ο πελάτης επιλέγει τοποθεσία παραλαβής κι επιστροφής οχήματος, ημερομηνίες κράτησης και την κατηγορία οχήματος που επιθυμεί. Τέλος στο κάτω μέρος του παραθύρου έχει 2 κουμπιά που δίνουν τις επιλογές να επιστρέψει στην προηγούμενο μενού ή να προχωρήσει στην κράτηση.



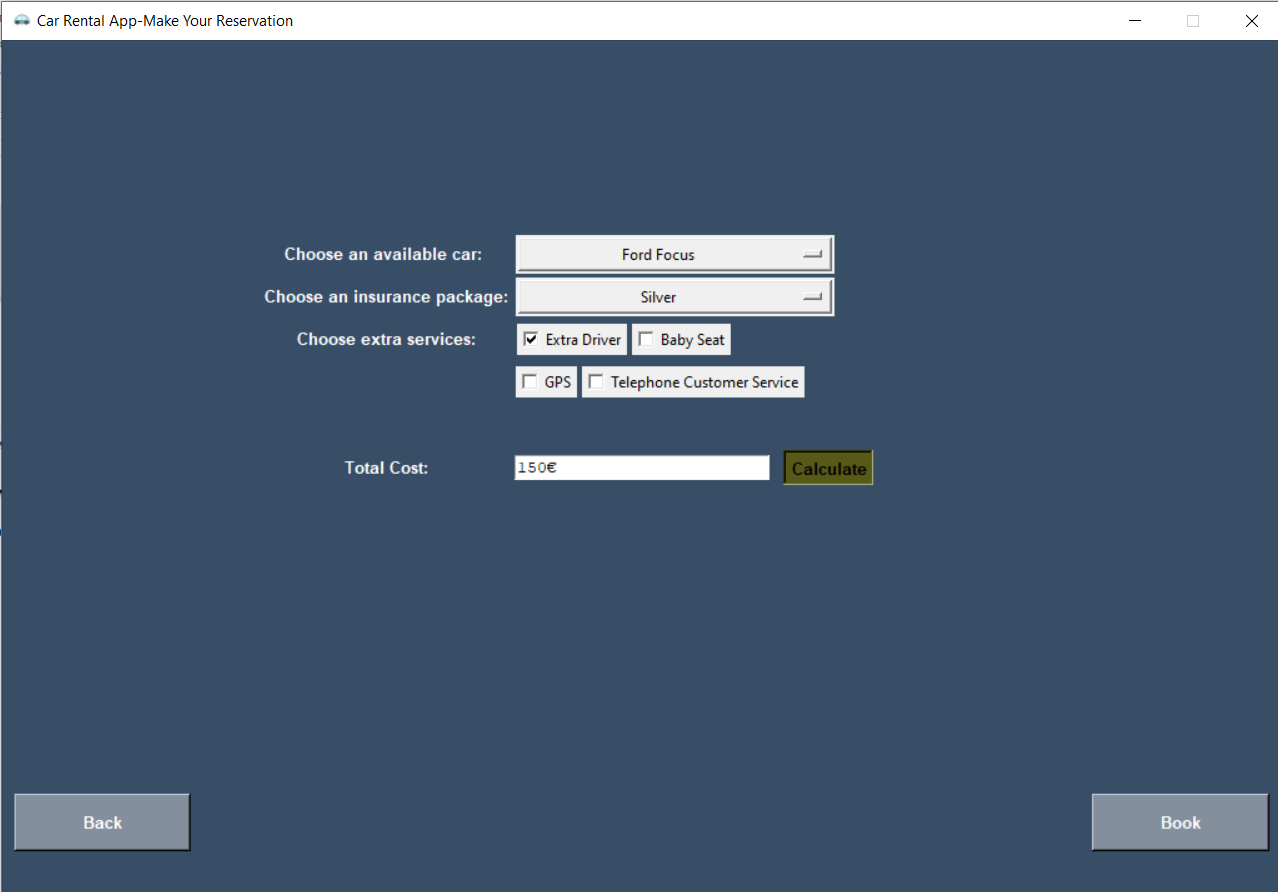
Δημιουργία κράτησης

Στην επόμενη σελίδα επιλέγει ένα από τα διαθέσιμα οχήματα, το ασφαλιστικό πακέτο και αν θέλει, κάποιες από τις διαθέσιμες υπηρεσίες. Ακόμα, πατώντας το κουμπί Calculate, μπορεί να δει το συνολικό κόστος της κράτησής του.

Τέλος πατάει το κουμπί BOOK για να προχωρήσει στην κράτηση.

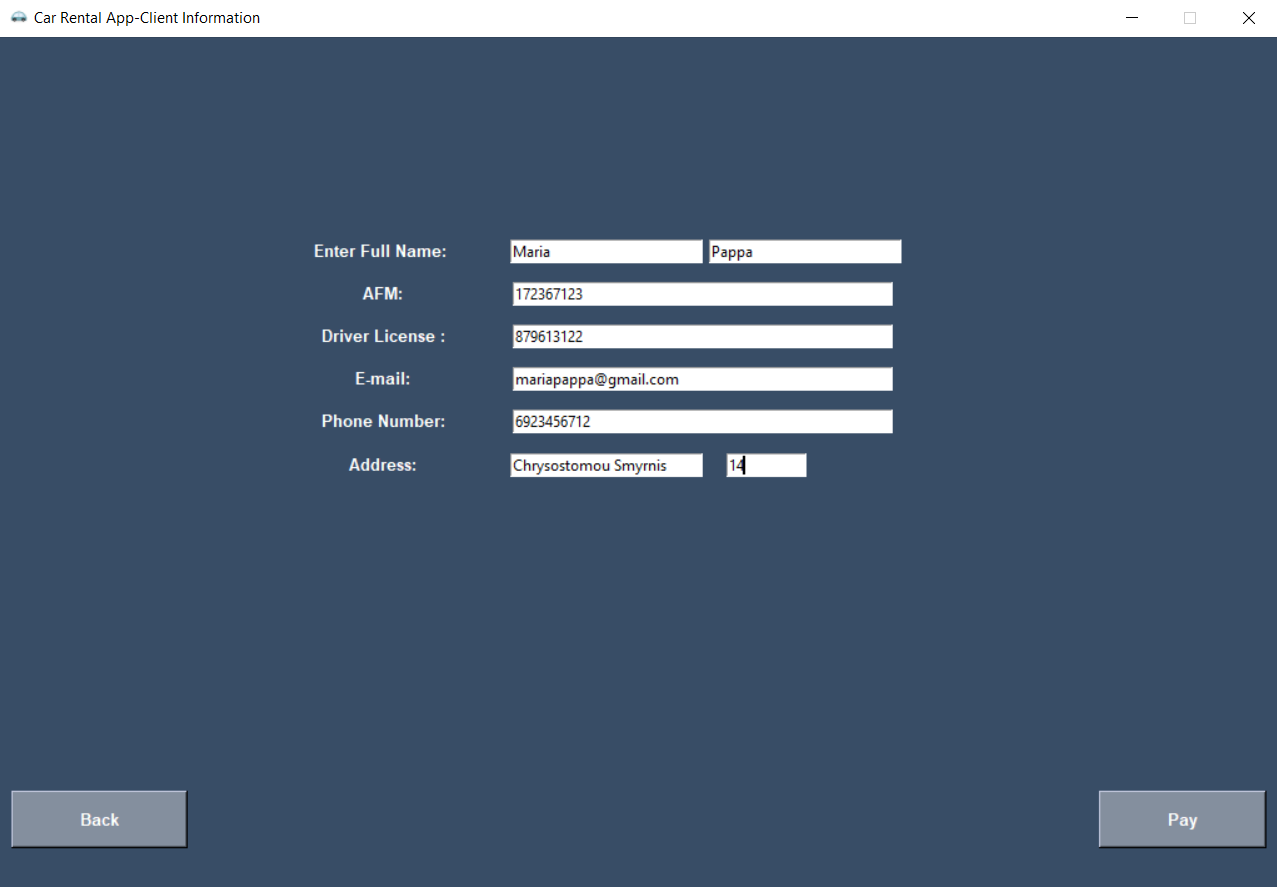


Επιλογή αυτοκινήτου



Υπολογισμός συνολικού κόστους (χωρίς ζημιές και καθυστερήσεις)

Στην επόμενη σελίδα συμπληρώνει τα προσωπικά του στοιχεία.

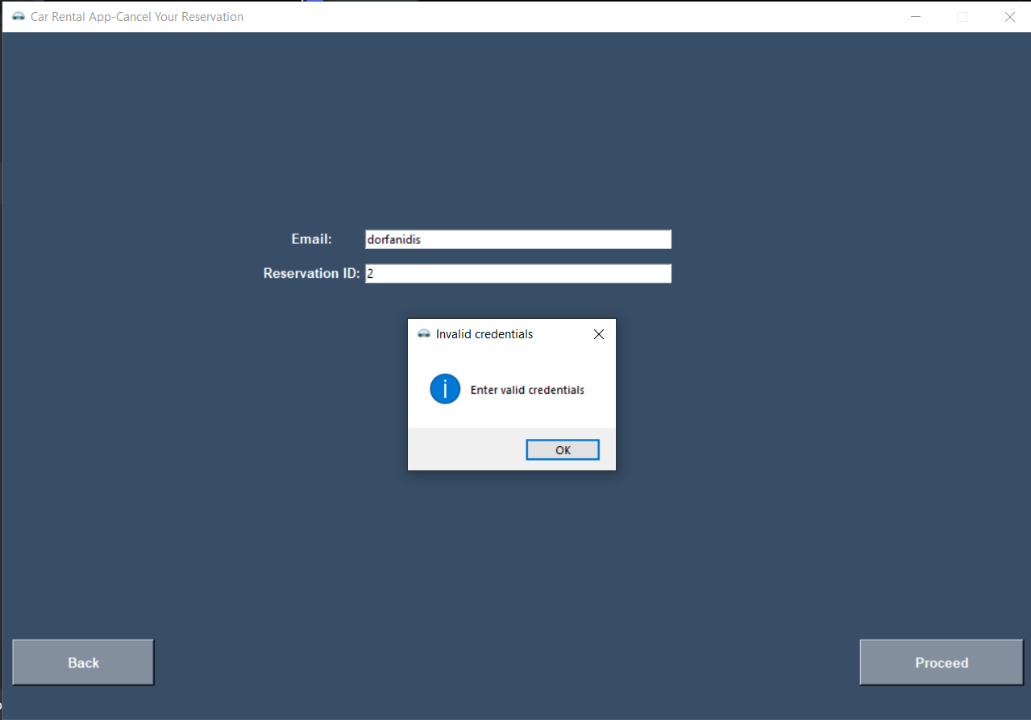


Εισαγωγή προσωπικών στοιχείων πελάτη

Τέλος συμπληρώνει τα στοιχεία της χρεωστικής ή πιστωτικής του κάρτας. Η κράτηση ολοκληρώνεται και εμφανίζεται στην οθόνη ένα μήνυμα που δείχνει στον πελάτη τον αριθμό κράτησής του.

* + 1. Ακύρωση κράτησης

Αυτό το κουμπί οδηγεί σε ένα παράθυρο που ζητάει από τον πελάτη να πληκτρολογήσει το email του και τον αριθμό της κράτησης που θέλει να ακυρώσει. Αν δεν υπάρχει διαθέσιμη κράτηση, εμφανίζεται ένα παράθυρο που ενημερώνει το χρήστη.



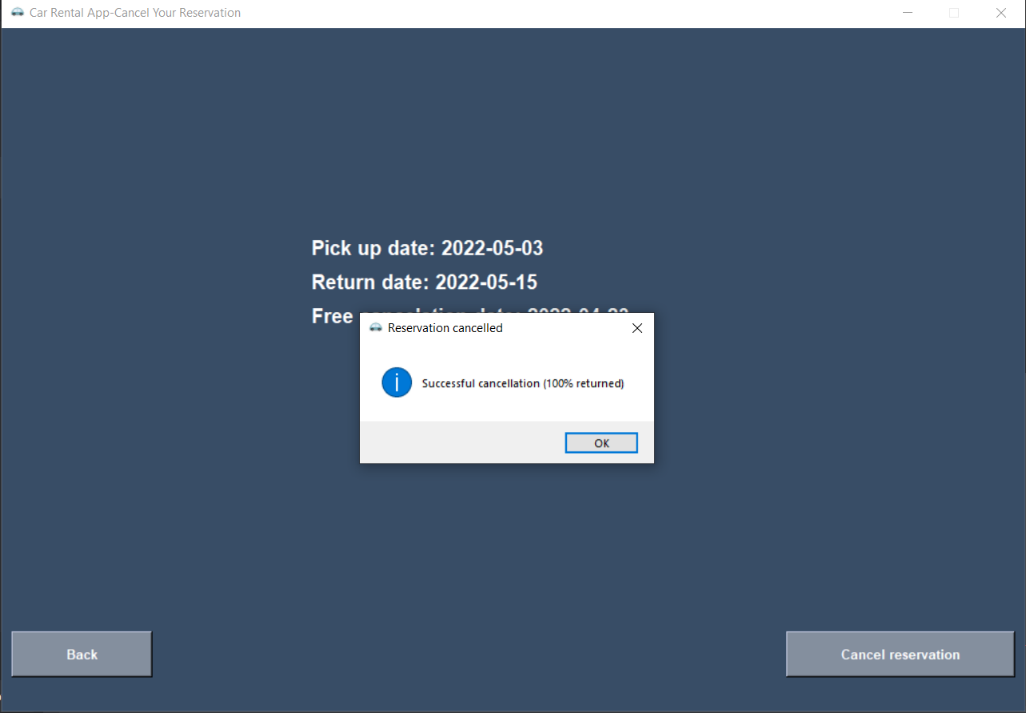
Μήνυμα εσφαλμένων αναγνωριστικών

Αν τα στοιχεία είναι σωστά και αντιστοιχούν σε κάποια κράτηση, εμφανίζονται στην οθόνη και πατώντας το κουμπί Cancel Reservation η κράτηση ακυρώνεται και επιστρέφεται το ανάλογο ποσό.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

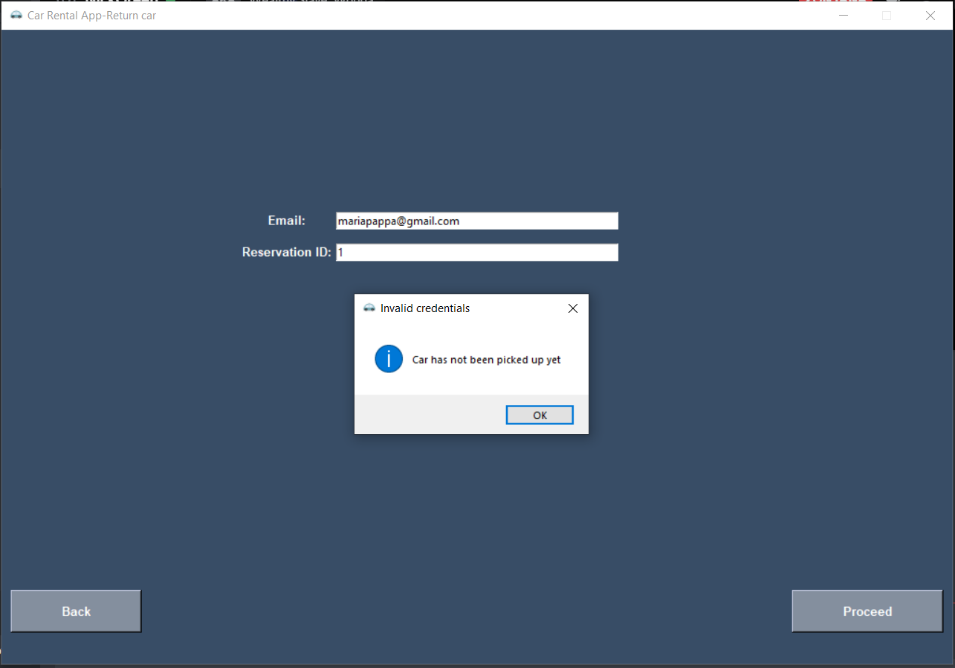
Στοιχεία κράτησης και δυνατότητας δωρεάν ακύρωσης



Μήνυμα επιτυχημένης δωρεάν ακύρωσης

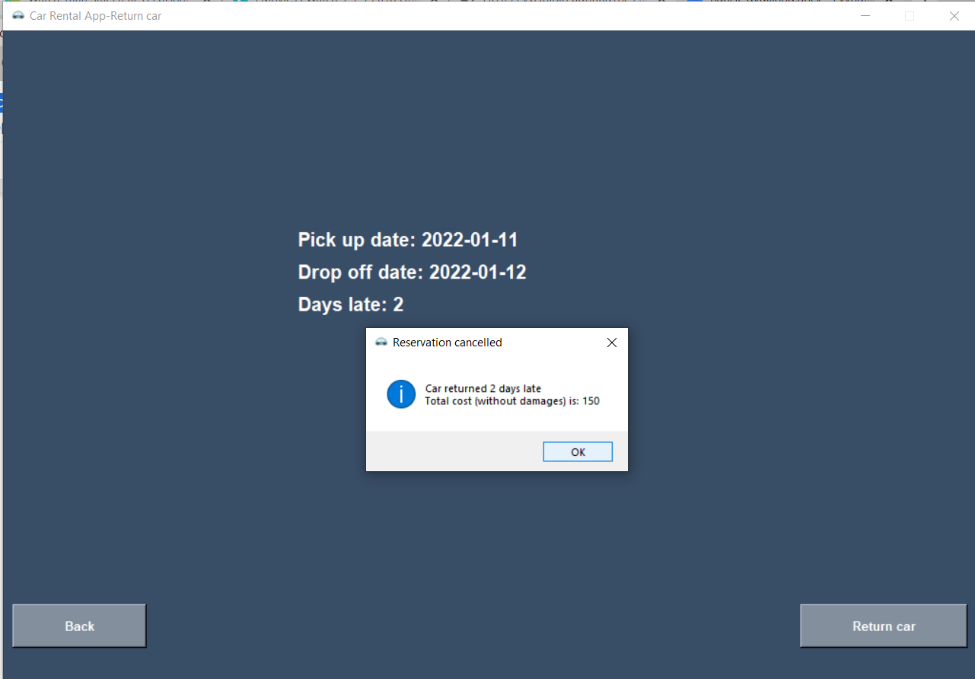
* + 1. Επιστροφή αυτοκινήτου

Όπως και με την ακύρωση κράτησης ο πελάτης πρέπει να πληκτρολογήσει το email του και τον αριθμό της κράτησης στην οποία αντιστοιχεί το όχημα που θέλει να επιστρέψει.



Μήνυμα σε περίπτωση που ο πελάτης πατήσει "Επιστροφή αυτοκινήτου" πριν το παρλάβει

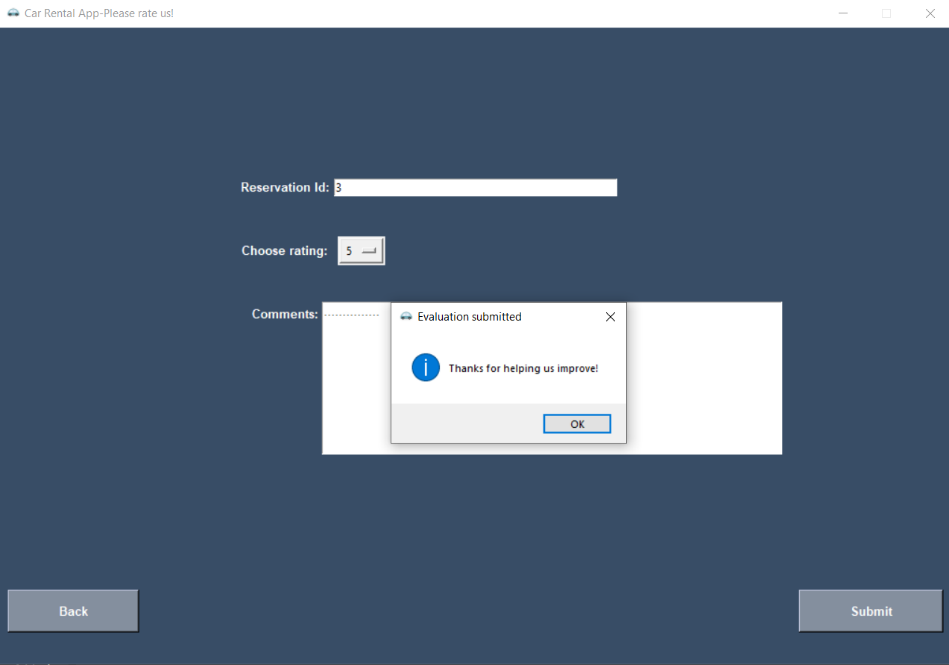
Εμφανίζονται οι λεπτομέρειες της κράτησης καθώς και πόσες μέρες έχει καθυστερήσει η επιστροφή. Πατώντας το κουμπί επιστροφής, εμφανίζεται η επιπλέον χρέωση σε περίπτωση που έχει καθυστερήσει η επιστροφή του οχήματος.



Μήνυμα σε περίπτωση καθυστερημένης επιστροφής

* + 1. Αξιολόγηση κράτησης

Τέλος η αξιολόγηση κράτησης επιτρέπει στον χρήστη να βαθμολογήσει την κράτηση του με κλίμακα από το 1 εώς το 5, καθώς και να αφήσει σχόλια, αν το επιθυμεί.



Μήνυμα επιτυχούς αξιολόγησης

Σε περίπτωση που οποιοδήποτε από τα στοιχεία δεν έχει την απαιτούμενη μορφή ή δεν έχει συμπληρωθεί, εμφανίζεται ένα παράθυρο που ενημερώνει το χρήστη και του ζητά να τα επαναπληκτρολογήσει.

* 1. Μέθοδος κατασκευής

Χρησιμοποιήθηκαν οι ακόλουθες βιβλιοθήκες για το πρόγραμμα:



Παρακάτω εξηγούνται ορισμένες από τις βασικότερες λειτουργίες των παραπάνω βιβλιοθηκών μαζί με τις σημαντικότερες συναρτήσεις που διατρέχουν το πρόγραμμα.

**Tkinter**

Για την κατασκευή της διεπαφής της εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη της Python, Tkinter, η οποία είναι η βασική βιβλιοθήκη της python για τη δημιουργία διεπαφών.

Η κύρια λειτουργία της tkinter είναι να δημιουργεί παράθυρα, κουμπιά, κουτιά εισαγωγής δεδομένων και ετικέτες, τα οποία ονομάζονται widgets και τοποθετούνται μέσα στο παράθυρο.

Για να αποκτήσει η Python πρόσβαση στη βιβλιοθήκη, την εισάγουμε με την εντολή import, όπως φαίνεται παραπάνω.

Οι βασικές εντολές για να δημιουργηθεί και να τρέξει ένα παράθυρο της tkinter είναι οι παρακάτω:



Χρειάζεται να υπάρχουν μία φορά στο πρόγραμμα, αφού δουλεύουμε στο ίδιο παράθυρο της tkinter.

Στη συνέχεια έχουμε τη δυνατότητα να κάνουμε αλλαγές στην εμφάνιση του παραθύρου, όπως είναι το χρώμα, το μέγεθος, το εικονίδιο και το φόντο:



Ακόμα χρησιμοποιήσαμε τα widgets Label, Button, OptionMenu, Checkbutton,Text και Entry για τη διαμόρφωση της εφαρμογής.

Το Label widget δημιουργεί μια ετικέτα με κείμενο και έχει την ακόλουθη σύνταξη:



Μέσα στη παρένθεση τοποθετείται το tk παράθυρο στο οποίο θέλουμε να βάλουμε το widget και μετά χαρακτηριστικά όπως το κείμενο, η γραμματοσειρά, το χρώμα των γραμμάτων και του φόντου και το μέγεθος. Τέλος με την εντολή place τοποθετούμε το widget σε συγκεκριμένο σημείο του παραθύρου.

Το Entry widget επιτρέπει την εισαγωγή δεδομένων από τον χρήστη και χρησιμοποιείται όπως φαίνεται παρακάτω:



Το OptionMenu επιτρέπει την επιλογή συγκεκριμένων δεδομένων από το χρήστη:



Η μεταβλητή options περιέχει τις επιλογές που εμφανίζονται στο μενού επιλογής ενώ η μεταβλητή dolclicked αποθηκεύει την επιλογή που θα κάνει ο χρήστης και ο τύπος της δηλώνεται με την εντολή StringVar().

Με την εντολή config ελέγχουμε το μέγεθος του μενού.

Τέλος το Button widget δημιουργεί ένα κουμπί το οποίο επιτελεί μια λειτουργία.



Το κουμπί όταν επιλέγεται ενεργοποιείται και εκτελεί τη συνάρτηση που βρίσκεται στην επιλογή command. Είναι σημαντικό να επισημάνουμε ότι όταν καλείται η συνάρτηση εκεί γίνεται χωρίς τη χρήση παρενθέσεων.

Αν χρειάζεται να εκτελέσουμε μια συνάρτηση που να δέχεται ως όρισμα μεταβλητές, το κάνουμε όπως φαίνεται παρακάτω:



Η εντολή grid χρησιμοποιείται αντί της place όταν θέλουμε να στοιχίσουμε τα widgets σε σειρές και στήλες.

Για την συλλογή σχολίων από τους πελάτες χρησιμοποιούμε το widget Τext:



To Checkbutton widget χρησιμοποιείται για την εμφάνιση επιλογών (toggle button) στο χρήστη και επιτρέπει την επιλογή πολλών ή και κανενός στοιχείου.



Στο παραπάνω παράδειγμα η μεταβλητή variable παίρνει την τιμή που δηλώνεται στις μεταβλητές onvalue & offvalue ανάλογα από το αν θα επιλεχθεί το κουτάκι ή όχι.

Το κάθε Checkbutton αποτελεί μία ξεχωριστή επιλογή.

Τέλος , για τα μηνύματα που θέλουμε να εμφανίζονται στην οθόνη κατά τη χρήση του προγράμματος χρησιμοποιούμε το message box, το οποίο χρειάζεται να το εισάγουμε ξεχωριστά με την εντολή import:



Μεταβλητές μέσα στις συναρτήσεις πρέπει να δηλωθούν ως Global για να μπορέσουν να χρησιμοποιηθούν από άλλες συναρτήσεις του προγράμματος, επομένως όλες οι μεταβλητές που περιέχουν τις επιλογές του χρήστη πρέπει να δηλωθούν έτσι ώστε να μπορέσουν να αποθηκευτούν στην αντίστοιχη συνάρτηση.

Για να χρησιμοποιήσουμε τις επιλογές του χρήστη από τα διάφορα widget σε όλες τις συναρτήσεις του προγράμματος δηλώνουμε τη μεταβλητή στη συνάρτηση όπου πρωτοεμφανίζεται ως global και ύστερα παίρνουμε την τιμή της με τη συνάρτηση get:



Ακόμα όταν χρησιμοποιούμε sqlite, είναι απαραίτητο να κάνουμε όλη τη διαδικασία σύνδεσης και αποσύνδεσης με τη βάση για κάθε συνάρτηση ξεχωριστά.

**Σημαντικά queries**

Ορισμένα από τα βασικότερα queries φαίνονται παρακάτω. Τα περισσότερα είναι παραλλαγή αυτών και δεν συμπεριλήφθηκαν στην αναφορά.

**Ανάκτηση πληροφοριών από τη βάση δεδομένων**

Για να εμφανίσουμε στοιχεία από τη βάση δεδομένων στο χρήστη χρησιμοποιήσαμε το ακόλουθο query:



H παραπάνω διαδικασία βρίσκει από τη βάση δεδομένων που έχουμε δημιουργήσει, όλα τα ονόματα των ασφαλιστικών πακέτων και τα αποθηκεύει στη λίστα ioptions, την οποία χρησιμοποιούμε σε ένα μενού επιλογής, επιτρέπωντας στο χρήστη να διαλέξει ένα από τα διαθέσιμα ασφαλιστικά πακέτα.

Με παρόμοιο τρόπο εμφανίζουμε τις διαθέσιμες τοποθεσίες και την επιλογή κατηγορίας αμαξιού.

**Queries με είσοδο μεταβλητές**

Όταν θέλουμε να φτιάξουμε ένα query που δέχεται ως είσοδο μια μεταβλητή χρησιμοποιούμε την παρακάτω σύνταξη:



**Διαθεσιμότητα οχημάτων**

Για να βρούμε τα διαθέσιμα αμάξια προς κράτηση χρησιμοποιούμε το ακόλουθο query:



**Αποθήκευση δεδομένων**

Αποθηκεύουμε τα δεδομένα που εισήγαγε ο χρήστης στην εφαρμογή με τον παρακάτω τρόπο:



Στο παραπάνω παράδειγμα αποθηκεύουμε τα στοιχεία της κάρτας του πελάτη στον πίνακα payment. Με παρόμοιο τρόπο αποθηκεύουμε όλα τα δεδομένα.

Ο ίδιος πελάτης, ωστόσο μπορεί να κάνει παραπάνω από μία κρατήσεις οπότε χρησιμοποιούμε το παρακάτω query για τον πίνακα στοιχείων του πελάτη, που αγνοεί δεδομένα που υπάρχουν ήδη στη βάση δεδομένων κι έτσι καταγράφεται νέα κράτηση με τα προϋπάρχοντα στοιχεία πελάτη:



**Ανάκτηση κλειδιών**

Ακόμα, με βάση τις επιλογές που έκανε ο χρήστης πρέπει να βρούμε από τη βάση τα αντίστοιχα κλειδιά που αποθηκεύονται στον πίνακα της κράτησης. Παρακάτω βρίσκουμε τα κλειδιά του οχήματος και της κατηγορίας στην οποία ανήκει, για να τα αποθηκεύσουμε μετά στον πίνακα reservation:



Ένα σημαντικό χαρακτηριστικό της sqlite είναι ότι δημιουργεί αυτόματα id key, αν δεν το ορίσουμε εμείς, όπως φαίνεται παρακάτω:



Αφού καταγράφουμε την καινούργια κράτηση στη βάση, της ανατείθεται αυτόματα ένας αριθμός, τον οποίο και βρίσκουμε από το ακόλουθο query:



**Ανανέωση πληροφοριών**

Η διορία δωρεάν ακύρωσης της κράτησης είναι έως 10 μέρες πριν την έναρξή της και υπολογίζεται με τον εξής τρόπο:



Κατά την ακύρωση της κράτησης ανανεώνουμε το τελικό κόστος, επιστρέφοντας το ανάλογο ποσό:

  
Τέλος κατά την επιστροφή αυτοκινήτου, ανανεώνουμε την ημερομηνία επιστροφής και τη τοποθεσία του έτσι ώστε να φαίνεται ότι βρίσκεται στο σημείο επιστροφής που επέλεξε ο πελάτης:





**Σημαντικές Συναρτήσεις**

Η εφαρμογή περιέχει πολλές συναρτήσεις που επιτελούν τις διάφορες λειτουργίες του προγράμματος. Παρακάτω επισημαίνονται οι ενδεικτικά ορισμένες από τις πιο βασικές.



H συνάρτηση αυτή διαγράφει όλα τα widgets στο παράθυρο προκειμένου να αλλάξουμε σελίδα στο πρόγραμμα.



To βασικό μενού της εφαρμογής. Ορίζει αρχικά τα χαρακτηριστικά του παραθύρου κι έπειτα έχει 4 κουμπιά που οδηγούν σε 4 διαφορετικές σελίδες του παραθύρου.



Αυτή η συνάρτηση ελέγχει αν έχουν επιλεγεί από τα Checkboxes οι τοποθεσίες και το μοντέλο του αυτοκινήτου και ενημερώνει με μήνυμα του widget messagebox μέχρι να επιλέξει τα απαραίτητα στοιχεία. Ελέγχει ακόμα αν οι ημερομηνίες έχουν την κατάλληλη μορφή και ρυθμίζει την μορφοποίηση. Αν τα στοιχεία είναι σωστά προχωράει αυτόματα στην επόμενη σελίδα της κράτησης.

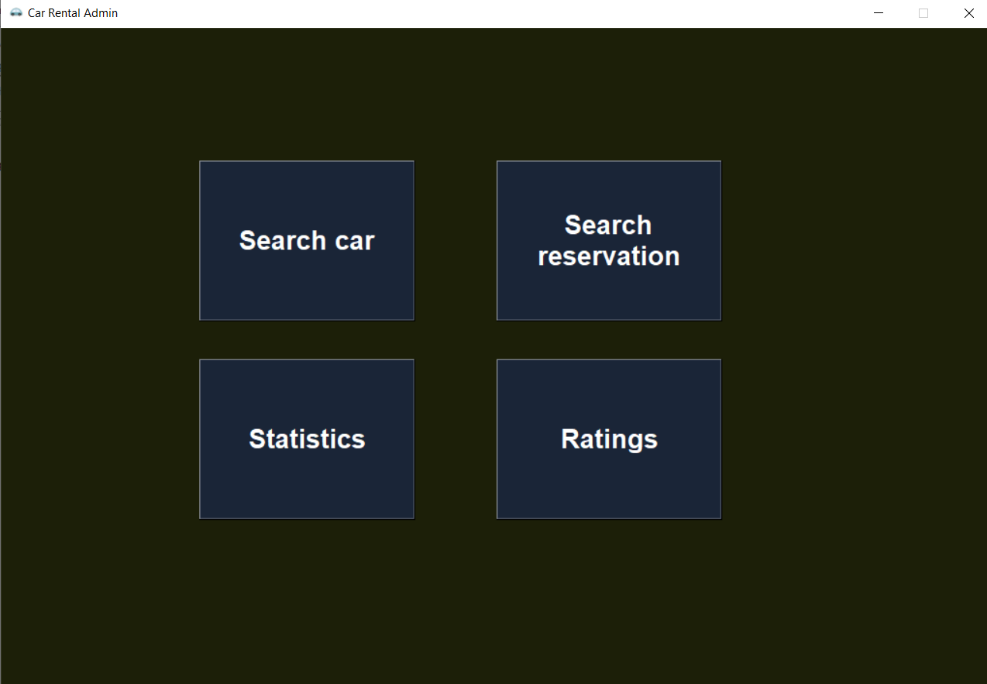


Η συνάρτηση Calculate cost υπολογίζει το συνολικό κόστος και το εμφανίζει στην οθόνη με το πάτημα του κουμπιού calculate. Χρησιμοποιεί σαν είσοδο τη μεταβλητή total\_cost από τη συνάρτηση και προσθέτει σε αυτή τα κόστη της ασφάλισης και των επιπλέον υπηρεσιών, τα οποία έχει λάβει από τη βάση με τη χρήση 2 queries και τα έχει περάσει σε 2 λίστες.



1. εφαρμογη διαχειριστη
   1. Περιγραφή λειτουργίας

Η εφαρμογή αυτή επιτρέπει στο διαχειριστή των κρατήσεων να βρει πληροφορίες με βάση τα οχήματα της εταιρείας και τις κρατήσεις. Μπορεί ακόμα να δει στατιστικά στοιχεία της επιχείρησης, καθώς και αξιολογήσεις πελατών για την εξυπηρέτησή τους.



Γραφική διεπαφή διαχειριστή

* 1. Λειτουργίες
     1. Αναζήτηση αυτοκινήτου

Βρίσκει και εμφανίζει τις πληροφορίες οχημάτων με βάση το μοντέλο, την πινακίδα ή την τοποθεσία στην οποία βρίσκονται.

Ο διαχειριστής επιλέγει μία από τις αυτές αναζητήσεις και πληκτρολογεί την αντίστοιχη πληροφορία στην μπάρα αναζήτησης. Εμφανίζονται τα οχήματα που αντιστοιχούν στοιχισμένα με βάση την πινακίδα, το μοντέλο, την κατηγορία, την τοποθεσία και την διαθεσιμότητά τους. Αν την χρονική στιγμή που γίνεται η αναζήτηση βρίσκεται κάποιο αυτοκίνητο σε κράτηση φαίνεται και ο αριθμός της.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, οθόνη

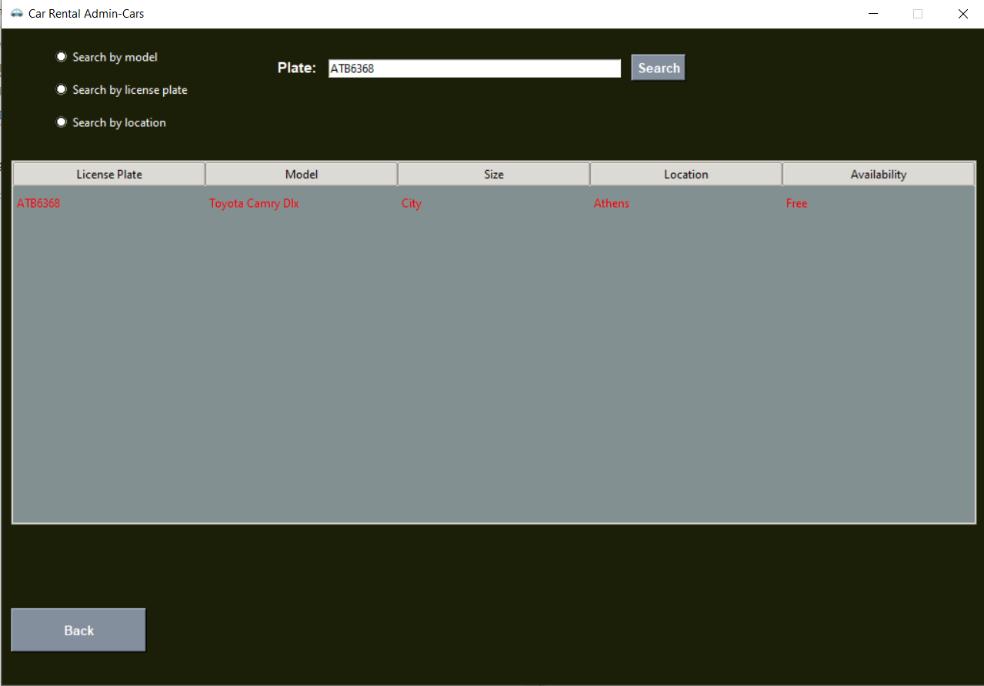
Περιγραφή που δημιουργήθηκε αυτόματα

Αναζήτηση με βάση το μοντέλο

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, οθόνη

Περιγραφή που δημιουργήθηκε αυτόματα

Αναζήτηση με βάση τη τοποθεσία



Αναζήτηση με βάση τις πινακίδες

* + 1. Αναζήτηση κράτησης

Με βάση τον αριθμό κράτησης που πληκτρολογεί ο διαχειριστής στην μπάρα αναζήτησης, εμφανίζονται τα στοιχεία της. Όπως και πριν μπορεί να επιλέξει να εμφανιστούν τα στοιχεία του πελάτη που την έχει κάνει.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Στοιχεία πελάτη σε μια συγκεκριμένη κράτηση

Επίσης μπορεί να δει τις λεπτομέρειες της κράτησης, δηλαδή τι πακέτα περιλαμβάνει, όπως τις επιπλέον υπηρεσίες και την ασφάλιση.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Στοιχεία αυτοκινήτου σε συγκεκριμένη κράτηση

Τέλος μπορεί να δει τις πληροφορίες που αφορούν το όχημα και τις ημερομηνίες παραλαβής και επιστροφής.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Στοιχεία κράτησης για μια συγκεκριμένη κράτηση σε περίπτωση που η κράτηση έχει ακυρωθεί

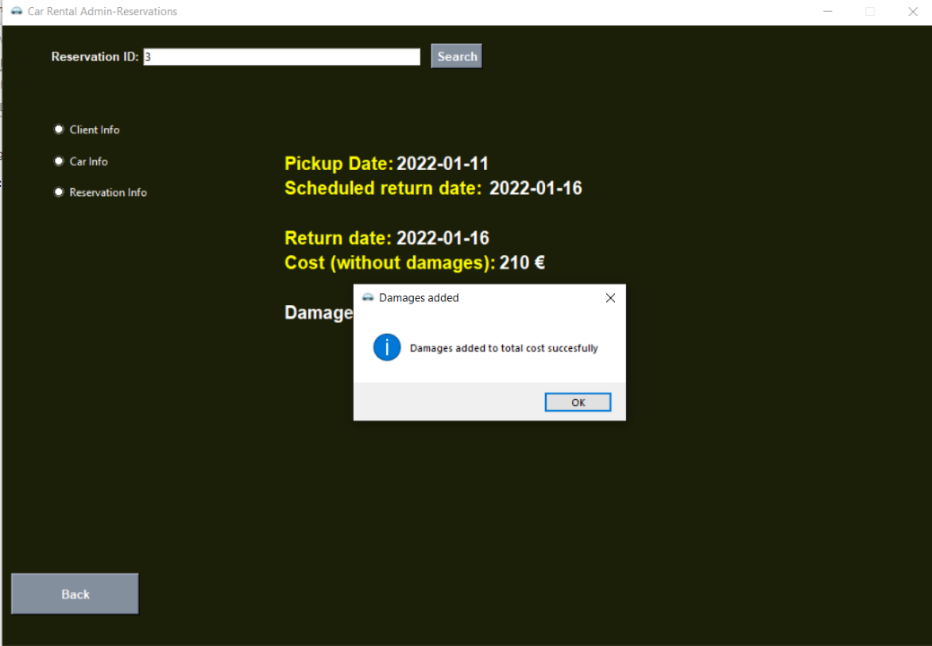
Εάν ο πελάτης έχει επιστρέψει το όχημα, τότε έχει τη δυνατότητα να προσθέσει και επιπλέον χρεώσεις που αφορούν τις ζημιές που μπορεί να έχει υποστεί.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Στοιχεία κράτησης για μια συγκεκριμένη κράτηση και δυνατότητα προσθήκης ζημιών στο συνολικό κόστος

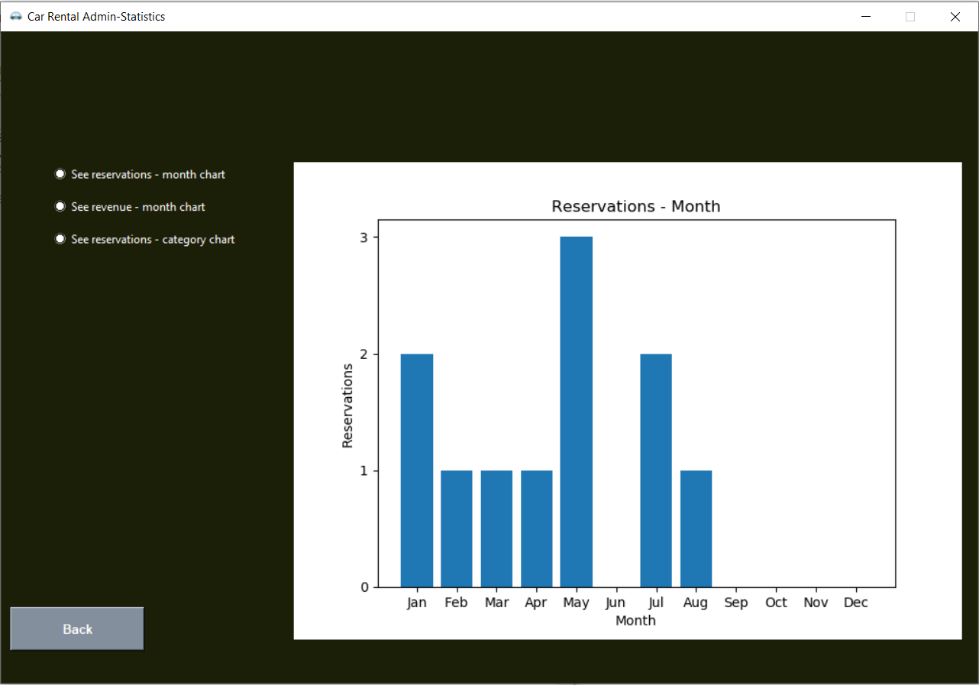
Σε αυτή την περίπτωση, αλλάζει το συνολικό κόστος κράτησης και χρεώνεται το επιπλέον ποσό.



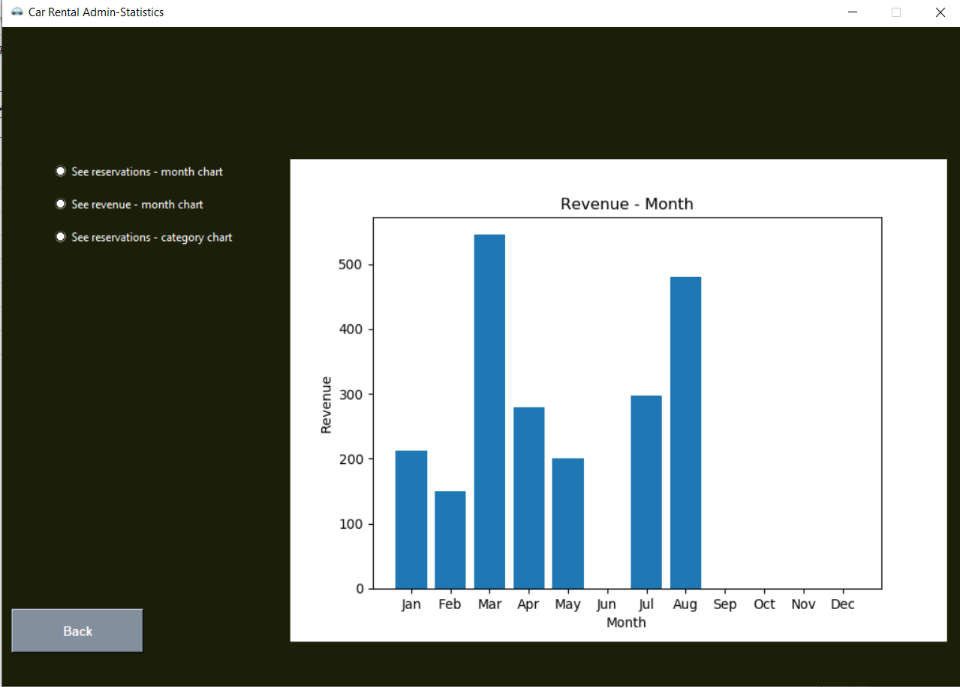
Μήνυμα επιτυχούς ενημέρωσης συνολικού κόστους

* + 1. Προβολή στατιστικών στοιχείων

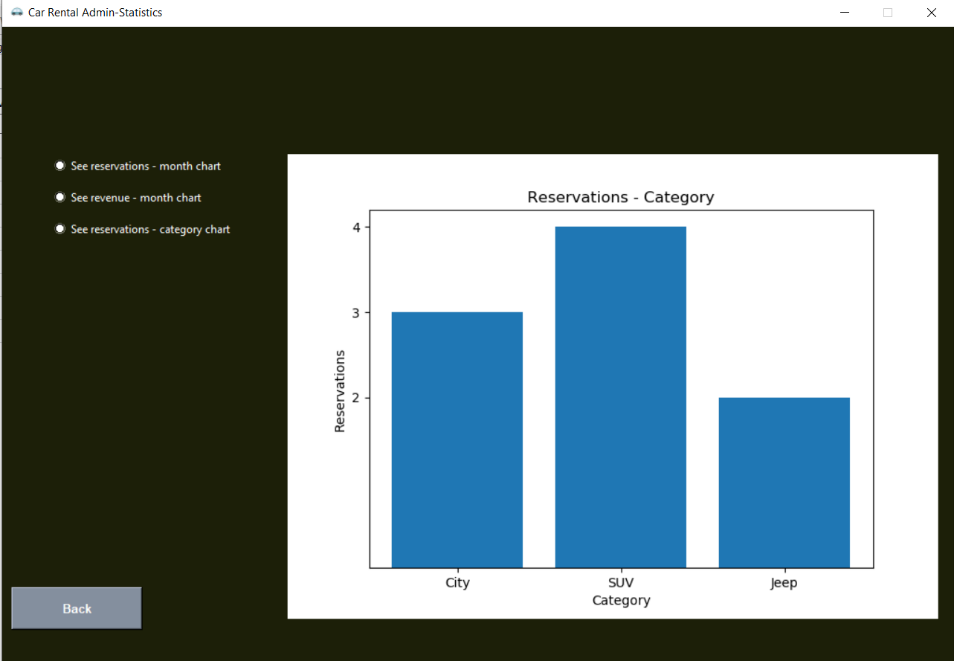
Αυτό το κουμπί οδηγεί σε μια σελίδα στην οποία φαίνονται σε διαγράμματα οι κρατήσεις ανά μήνα, τα μηνιαία έσοδα και οι κρατήσεις με βάση τις κατηγορίες οχημάτων.



Γράφημα κρατήσεων ανά μήνα



Γράφημα εσόδων ανά μήνα



Γράφημα κρατήσεων ανά κατηγορία αυτοκινήτου

* + 1. Προβολή αξιολογήσεων

Σε αυτή τη σελίδα βλέπουμε τη μέση τιμή αξιολόγησης και τα σχόλια που έχουν αφήσει οι πελάτες.

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Προβολή αξιολογήσεων και μέσου όρο

* 1. Μέθοδος κατασκευής

Για την εφαρμογή αυτή χρησιμοποιήθηκαν οι βιβλιοθήκες που έχουν προαναφερθεί, με επιπλέον χρήση των βιβλιοθηκών numpy και matplotlib.

Συγκεκριμένα εισαγάγαμε τις παρακάτω βιβλιοθήκες:



**Tkinter**

Η tkinter χρησιμοποιήθηκε για την κατασκευή της διεπαφής και αυτής της εφαρμογής. Χρησιμοποιήθηκαν τα widgets που προαναφέρθηκαν με παρόμοιο τρόπο. Επιπλέον χρησιμοποιήθηκε το widget Scrollbar,το Treeview, το Radiobutton και το Canvas.

To scrollbar widget δημιουργεί μια γραμμή κύλισης για την προβολή δεδομένων πάνω στο παράθυρο. Χρησιμοποιείται με την παρακάτω σύνταξη εντολών:



H yscrollcommand δημιουργεί τη μπάρα κατακόρυφα.

Το Radiobutton δημιουργεί κουμπιά πολλαπλής επιλογής και επιτρέπει την επιλογή ενός μόνο από αυτά:



To widget Canvas χρησιμοποιείται για την απεικόνιση γραφημάτων:



**Σημαντικά Queries**

Ορισμένα από τα βασικότερα queries φαίνονται παρακάτω. Τα περισσότερα είναι παραλλαγή αυτών και δεν συμπεριλήφθηκαν στην αναφορά.

**Μέσος όρος**

Βρίσκουμε το μέσο όρο αξιολογήσεων των κρατήσεων όπως φαίνεται παρακάτω:



**Έσοδα ανά μήνα**

Το παρακάτω query δημιουργεί μία καινούρια μεταβλητή που περιέχει το μήνα από την ημερομηνία κράτησης, επιλέγει το συνολικό κόστος και εμφανίζει το συνολικό κόστος των κρατήσεων ανά μήνα.



**Κρατήσεις ανά μήνα**

To παρακάτω query μετράει τις κρατήσεις ανά μήνα.



**Αυτοκίνητα που δεν βρίσκονται σε κράτηση**

Ενώ αυτό βρίσκει τα αμάξια που δεν συμμετέχουν σε κάποια κράτηση τη συγκεκριμένη χρονική στιγμή.



**Σημαντικές Συναρτήσεις**

Παρακάτω σημειώνονται ενδεικτικά ορισμένες βασικές συναρτήσεις του προγράμματος του διαχειριστή.



Η παραπάνω συνάρτηση αφορά την παρουσίαση των αξιολογήσεων των πελατών στην εφαρμογή του διαχειριστή.

Συνδέεται με τη βάση και υπολογίζει με sqlite το μέσο όρο των αξιολογήσεων των πελατών. Τον εμφανίζει στην οθόνη μαζί με τα σχόλια που έχουν αφήσει για την κάθε κράτηση σε ένα tkinter scrollbar widget.



H συνάρτηση αυτή δημιουργεί το γράφημα μηνιαίων εσόδων μέσω της sqlite και χρησιμοποιώντας τη matplotlib βιβλιοθήκη για το σχηματισμό του γραφήματος. Η tkinter έχει το ειδικό widget canvas για να εμφανίσει τα γραφήματα στην οθόνη. Παρόμοια λειτουργούν και οι συναρτήσεις που φτιάχνουν τα υπόλοιπα γραφήματα.





Η συνάρτηση παραπάνω δημιουργεί τις στήλες με τις πληροφορίες των οχημάτων. Μέσω της sqlite ενώνει τους πίνακες με τις απαραίτητες πληροφορίες προκειμένου να τις εμφανίσει στο παράθυρο.



Δέχεται ως ορίσματα τον αριθμό κράτησης και το ποσό των ζημιών και ανανεώνει το κόστος της κράτησης, προσθέτοντας σε αυτό τις ζημιές.



Αναζήτηση οχημάτων με βάση την τοποθεσία. Εμφανίζει σε πίνακα τα οχήματα μαζί με τις πινακίδες, το μοντέλο, την τοποθεσία τους και τη διαθεσιμότητα τους. Τη διαθεσιμότητα των αυτοκινήτων τη βρίσκει με το query free, το οποίο ελέγχει αν κάποιο αμάξι βρίσκεται σε κράτηση τη συγκεκριμένη χρονική στιγμή. Τέλος ελέγχει αν οι πληροφορίες που εισάχθηκαν είναι έγκυρες.

1. αξιολογηση

Η εφαρμογή πραγματοποίησης κρατήσεων προσομοιώνει αρκετά πιστά την πραγματική ενοικίαση αυτοκινήτων online, καθώς πετύχαμε όλες τις λειτουργίες που είχαμε σκοπό να έχει τη δυνατότητα να πραγματοποιεί ο πελάτης και ο διαχειριστής. Πιο συγκεκριμένα ο πελάτης μπορεί να εκτελέσει όλες τις ενέργειες που θα αναζητούσε σε ένα σύστημα online κρατήσεων αυτοκινήτων με ευκολία και απλότητα. Ακόμη η εφαρμογή του admin παρέχει σημαντικές πληροφορίες και λειτουργίες στον διαχειριστή, καθώς του δίνει ολοκληρωτική εποπτεία των κινήσεων που έχουν εκτελέσει όλοι οι πελάτες, ενώ ταυτόχρονα μέσω των στατιστικών στοιχείων λαμβάνει αναγκαίο feedback από τους πελάτες του, που θα τον βοηθούσε να βελτιώσει τις υπηρεσίες του.

Φυσικά οι λειτουργίες δεν είναι τέλειες. Επειδή δεν προλάβαμε να κάνουμε εκτενές debugging στην εφαρμογή, ενδέχεται να υπάρχουν bugs. Όσο κάναμε testing την εφαρμογή διορθώσαμε όσα βρήκαμε, όμως λόγω της έκτασης της, υπάρχει πιθανότητα να υπάρχουν και άλλα και θα πρέπει να διορθωθούν άμεσα. Όσον αφορά τώρα τις λειτουργίες των εφαρμογών, σκεφτήκαμε ότι θα μπορούσαμε να προσθέσουμε περισσότερα φίλτρα αναζήτησης όταν ο πελάτης κάνει κάποια κράτηση, όπως να μπορεί να επιλέξει αν ο μοχλός ταχυτήτων θα είναι αυτόματος ή manual.

Στο κομμάτι του DBMS, δεν τίθενται θέματα απόδοσης. Οι πίνακες συνεργάζονται αποδοτικά και αποτελεσματικά μεταξύ τους. Ενδεχομένως, στην περίπτωση μεγαλύτερου όγκου δεδομένων ίσως χρειαζόταν χρήση ευρετηρίων (Indexes), ώστε να επιταχύνονταν ορισμένες ενέργειας της βάσης δεδομένων. Αυτή θα ήταν μια προσθήκη, η οποία στην παρούσα φάση του συστήματος δεν θα άλλαζε αισθητά την εμπειρία χρήσης, όμως εάν είχαμε αποθηκευμένα στη βάση μας περισσότερα μοντέλα ή γινόντουσαν πάρα πολλές κρατήσεις θα ήταν ιδιαίτερα ωφέλιμο.

1. οδηγιες εγκαταστασησ

Για να λειτουργήσει το πρόγραμμα, πρέπει ο χρήστης να έχει εγκατεστημένη στον υπολογιστή του την Python, καθώς και όλες τις βιβλιοθήκες που χρησιμοποιούνται μέσα σε αυτό.

Η Python είναι μια πολύ φιλική προς το χρήστη γλώσσα προγραμματισμού, εύκολη στη σύνταξη και δημοφιλής για αρχάριους. Οι βιβλιοθήκες της και οι εφαρμογές της την καθιστούν πολύ ευέλικτη και χάρη στα έτοιμα πακέτα της είναι ιδανική για εξόρυξη πληροφοριών και επεξεργασία δεδομένων.

Οδηγίες Εγκατάστασης:

1. Από την επίσημη ιστοσελίδα της Python (<https://www.python.org/>) επιλέγουμε και κατεβάζουμε το κατάλληλο αρχείο, σύμφωνα με τις προδιαγραφές του υπολογιστή μας.

2. Τρέχουμε τον installer. Φροντίζουμε να επιλέξουμε τα κουτάκια στα οποία αναγράφεται “Install Launcher for All Users” και “Add Python to PATH”.

Πρέπει επίσης να εγκαταστήσουμε τη βιβλιοθήκη Numpy και τη matplotlib.

H Numpy είναι μία από τις πιο δημοφιλείς βιβλιοθήκες μηχανικής μάθησης. Χρησιμοποιείται από πολλές άλλες βιβλιοθήκες της Python αφού μπορεί και κάνει πολύπλοκες μαθηματικές πράξεις γρήγορα και με τη χρήση απλών εντολών.

Για την εγκατάσταση πατάμε στο command prompt την εντολή : pip install –upgrade numpy.

Η βιβλιοθήκη matplotlib παρέχει τη δυνατότητα να γίνει γραφική αναπαράσταση των δεδομένων με διάφορους τρόπους. Αποτελεί extension της Numpy.

Για την εγκατάσταση πατάμε στο command prompt την εντολή : pip install –upgrade matplotlib.

Ακόμα χρειάζεται η χρήση της sqlite, η οποία βρίσκεται στην standard library της Python.

H sqlite είναι μια βιβλιοθήκη διαχείρισης σχεσιακών βάσεων δεδομένων. Σε αντίθεση με άλλα συστήματα διαχείρισης βάσης δεδομένων, δεν είναι μια ξεχωριστή διεργασία που προσπελάσεται από μια εφαρμογή πελάτη, αλλά ένα ενσωματωμένο μέρος της.

REFERENCES

1. <https://www.avis.gr>
2. <https://www.carnmotion.gr/?gclid=Cj0KCQiAoY-PBhCNARIsABcz771-DCWq0ZTunEl7pE8HQ8u0APRytEzG2X6jVFE78PYd27_S39Xz334aAoGOEALw_wcB>
3. <https://www.yourmechanic.com/article/the-most-and-least-expensive-cars-to-maintain-by-maddy-martin>
4. <https://eclass.upatras.gr/modules/document/file.php/EE879/%CE%A5%CE%BB%CE%B9%CE%BA%CF%8C%20%CE%B1%CF%83%CE%BA%CE%AE%CF%83%CE%B5%CF%89%CE%BD/%CE%91%CE%B2%CE%BF%CF%8D%CF%81%CE%B7%CF%82%20%CE%BA.%CE%AC.%20-%20%CE%95%CF%85%CF%81%CE%B5%CF%84%CE%B9%CE%BA%CE%AE%20%CE%B1%CE%BE%CE%B9%CE%BF%CE%BB%CF%8C%CE%B3%CE%B7%CF%83%CE%B7.pdf>
5. <https://www.usabilitybok.org/heuristic-evaluation>
6. <https://www.python.org/>
7. <https://www.sqlitetutorial.net/sqlite-python/>
8. <https://www.tutorialspoint.com/python/python_gui_programming.htm>
9. <https://www.youtube.com/watch?v=byHcYRpMgI4&list=PLIiVi3_85D7K3lgYyGy05hUzFkCyfzAkZ&index=111&t=2870s>
10. <https://www.youtube.com/watch?v=YXPyB4XeYLA&list=PLIiVi3_85D7K3lgYyGy05hUzFkCyfzAkZ&index=108&t=12168s>
11. <https://www.studytonight.com/tkinter/python-tkinter-scrollbar-widget>
12. <https://www.tutorialspoint.com/python/tk_grid.htm>
13. <https://www.yourmechanic.com/article/the-most-and-least-expensive-cars-to-maintain-by-maddy-martin>