|  |
| --- |
|  |

Εικόνα που περιέχει καρτούν, κύκλος, γραφικά, clipart

Περιγραφή που δημιουργήθηκε αυτόματα

**Μηχανική Λογισμικού Ι**

**Τομέας Ηλεκτρονικής και Υπολογιστών**

**Τμήμα ΗΜΜΥ**

**Α.Π.Θ**

CurbSprings  
Your Reliable Partner in Urban Mobility

Προδιαγραφές Συστήματος

Del.1.2

Version 0.1

Πίττης Γεώργιος [gkpittis@ece.auth.gr](mailto:gkpittis@ece.auth.gr)Γουρδομιχάλης Αναστάσιος [anasgour@ece.auth.gr](mailto:anasgour@ece.auth.gr)   
Τσαρναδέλης Αθανάσιος Γρηγόριος [atsarnad@ece.auth.gr](mailto:atsarnad@ece.auth.gr)Φωτιάδης Αλέξανδρος afotiadis@ece.auth.gr

14/04/2024

Μέλη της Ομάδας Ανάπτυξης

|  |  |  |
| --- | --- | --- |
| **Όνομα** | **OA** | **Email** |
| Α. Συμεωνίδης | \* | [asymeon@issel.ee.auth.gr](mailto:asymeon@issel.ee.auth.gr) |
| Πίττης Γεώργιος | 25 | [gkpittis@ece.auth.gr](mailto:gkpittis@ece.auth.gr) |
| Γουρδομιχάλης Αναστάσιος | 25 | anasgour@ece.auth.gr |
| Τσαρναδέλης Αθανάσιος Γρηγόριος | 25 | [atsarnad@ece.auth.gr](mailto:atsarnad@ece.auth.gr) |
| Φωτιάδης Αλέξανδρος | 25 | afotiadis@ece.auth.gr |
|  |  |  |
|  |  |  |

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων 3

Λίστα Σχημάτων 5

1 Στατική Μοντελοποίηση 6

1.1 Πακέτο κλάσεων Feature 1: Reserve a parking spot 6

1.1.1 Ορισμός Κλάσεων 6

1.1.1.1 Ορισμός Κλάσης User 6

1.1.1.2 Ορισμός Κλάσης Reservation 7

1.1.1.3 Ορισμός Κλάσης ReservationController 8

1.1.1.4 Ορισμός Κλάσης MakeReservationGUI 8

1.1.1.5 Ορισμός Κλάσης SpotInformationGUI 9

1.1.1.6 Ορισμός Κλάσης DBProxy 9

1.1.2 Διάγραμμα κλάσεων 10

1.2 Πακέτο κλάσεων Feature 2: Manage Reservation 10

1.2.1 Ορισμός Κλάσεων 10

1.2.1.1 Ορισμός Κλάσης Manage Reservation GUI 10

1.2.1.2 Ορισμός Κλάσης Update Reservation GUI 11

1.2.1.2 Ορισμός Κλάσης Update Reservation GUI 11

1.2.2 Διάγραμμα κλάσεων 12

1.3 Πακέτο κλάσεων Feature 3: Manage parking spots 13

1.3.1 Ορισμός Κλάσεων 13

1.3.1.1 Ορισμός Κλάσης Spot Owner 13

1.3.1.2 Ορισμός Κλάσης Parking Spot Controller 14

1.3.1.3 Ορισμός Κλάσης Modify Spot Information GUI 15

1.3.1.3 Ορισμός Κλάσης Remove Spot GUI 15

1.3.2 Διάγραμμα κλάσεων 16

1.4 Πακέτο κλάσεων Feature 4: View available parking spots 16

1.4.1 Ορισμός Κλάσεων 16

1.4.1.1 Ορισμός Κλάσης < ShowAllAvailableSpotsGUI > 16

1.4.1.2 Ορισμός Κλάσης < HomePageGUI > 17

1.4.1.3 Ορισμός Κλάσης < StartPageGUI > 18

1.4.1.4 Ορισμός Κλάσης < DisplayController > 19

1.4.2 Διάγραμμα κλάσεων 20

1.5 Πακέτο κλάσεων Feature 5: Search for available parking spots 21

1.5.1 Ορισμός Κλάσεων 21

1.5.1.1 Ορισμός Κλάσης <SearchSpotByChargerAvailability> 21

1.5.2 Διάγραμμα κλάσεων 24

1.6 Πακέτο κλάσεων Feature 6: Add spot owner 25

1.6.1 Ορισμός Κλάσεων 25

1.6.1.1 Ορισμός Κλάσης <Όνομα Κλάσης> 25

1.6.2 Διάγραμμα κλάσεων 25

1.7 Πακέτο κλάσεων Feature 7: Add parking spots 25

1.7.1 Ορισμός Κλάσεων 25

1.7.1.1 Ορισμός Κλάσης Spot Owner Page GUI 25

1.7.2 Διάγραμμα κλάσεων 26

1.8 Πακέτο κλάσεων Feature 8: Make Payment 27

1.8.1 Ορισμός Κλάσεων 27

1.8.1.1 Ορισμός Κλάσης Payment 27

1.8.1.2 Ορισμός Κλάσης PaymentScreenGUI 27

1.8.1.3 Ορισμός Κλάσης PaymentScreenGUI 28

1.8.2 Διάγραμμα κλάσεων 28

1.9 Πακέτο κλάσεων Feature 9: Register vehicle license plate 29

1.9.1 Ορισμός Κλάσεων 29

1.9.1.1 Ορισμός Κλάσης <Όνομα Κλάσης> 29

1.9.2 Διάγραμμα κλάσεων 29

1.10 Πακέτο κλάσεων Feature 10: Modify vehicle license plate 29

1.10.1 Ορισμός Κλάσεων 29

1.10.1.1 Ορισμός Κλάσης <Όνομα Κλάσης> 29

1.10.2 Διάγραμμα κλάσεων 29

2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν 30

3 Δυναμική Μοντελοποίηση 31

3.1 Gherkin Scenario X (Feature Y) 31

Παράρτημα I – Γλωσσάριο 32

Παράρτημα IΙ – Ανοιχτά Θέματα 33

Λίστα Σχημάτων

**Δεν βρέθηκαν καταχωρήσεις πίνακα εικόνων.**

1 Στατική Μοντελοποίηση

1.1 Πακέτο κλάσεων Feature 1: Reserve a parking spot

1.1.1 Ορισμός Κλάσεων

1.1.1.1 Ορισμός Κλάσης User

A blue screen with black text

Description automatically generated

**Χαρακτηριστικά κλάσης**

* UserID: Integer : το μοναδικό αναγνωριστικό του χρήστη
* UserName: String : το ονοματεπώνυμο του χρήστη
* UserLicensePlate: LicensePlate : το αντικείμενο License Plate που έχει ο χρήστης
* UserReservation: <list> Reservation : μια λίστα αντικειμένων Reservation, με τις κρατήσεις του χρήστη

**Μέθοδοι κλάσης**

* User(UserID:integer, UserName:String, UserLicensePlate:LicensePlate, UserReservation:list <Reservation>): void : Συνάρτηση δόμησης της κλάσης user, παίρνει τα κατάλληλα ορίσματα για να δημιουργήσει ένα αντικείμενο user.
* setUserID(UserID:String): void : Παίρνει όρισμα τον αριθμό του χρήστη και τον τροποποιεί.
* setUserName(UserName:String): void : Παίρνει όρισμα ένα string και ορίζει το όνομα του χρήστη.
* setUserLicensePlate(UserLicensePlate:LicensePlate): void : Παίρνει όρισμα ένα αντικείμενο LicensePlate και ορίζει την πινακίδα του χρήστη.
* setUserReservation(UserReservation:list <Reservation>): void : Παίρνει όρισμα μια λίστα από Reservation και ορίζει την λίστα κρατήσεων του χρήστη.
* getUserID(): Integer : Επιστρέφει το μοναδικό ID του χρήστη.
* getUserName(): String : Επιστρέφει το ονοματεπώνυμο του χρήστη.
* getUserLicensePlate(): LicensePlate : Επιστρέφει το αντικείμενο LicensePlate με την πινακίδου του χρήστη.
* getUserReservation(): list <Reservation> : Επιστρέφει μια λίστα Reservation με τις κρατήσεις του χρήστη.

1.1.1.2 Ορισμός Κλάσης Reservation

A blue screen with black text

Description automatically generated

**Χαρακτηριστικά κλάσης**

* ReservationID: Integer : Το μοναδικό αναγνωριστικό ID της κράτησης.
* ReservationDate: String : Η ημερομηνία της κράτησης.
* ReservationStartTime: String : Η ώρα εκκίνησης της κράτησης.
* ReservationExpected Duration: Float : Η αναμενόμενη διάρκεια της κράτησης σε ώρες.
* ReservationUser: User : Ο χρήστης που έκανε την κράτηση.
* ReservationSpot: ParkingSpot : Η θέση την οποία αφορά η κράτηση

**Μέθοδοι κλάσης**

* Reservation(ReservationID: Integer, ReservationStartTime: String, ReservationExpected Duration: Float, ReservationSpot: ParkingSpot, ReservationUser: User, ReservationDate: String): void : Η συνάρτηση δόμησης της κλάσης, παίρνει τα κατάλληλα ορίσματα και δημιουργεί ένα αντικείμενο κράτησης.
* setReservationStartTime(ReservationStartTime: String): void : Παίρνει σαν είσοδο την ώρα της κράτησης και την ορίζει στην κλάση.
* setReservationExpectedDuration (Reservation Expected Duration: Float): void : Παίρνει όρισμα την αναμενόμενη διάρκεια και την ορίζει στην κλάση.
* setReservationSpot(ReservationSpot: ParkingSpot): void : Παίρνει όρισμα ένα αντικείμενο ParkingSpot και ορίζει την θέση πάρκινγκ της κράτησης.
* setReservationUser(ReservationUser: User): void : Παίρνει σαν όρισμα ένα αντικείμενο user και ορίζει στην κλάση τον χρήστη που έκανε την κράτηση.
* setReservationDate(Reservation Date: String): void : Παίρνει σαν όρισμα την ημερομηνία της κράτησης και την ορίζει στην κράτηση.
* setReservationID(ReservationID: Integer): void : Παίρνει σαν όρισμα ένα ακέραιο και τον ορίζει σαν ID στην κλάση.
* getReservationID(): Integer : Επιστρέφει το μοναδικό αναγνωριστικό της κράτησης.
* getReservationStartTime(): String : Επιστρέφει την ώρα της κράτησης.
* getReservationExpected Duration(): Float : Επιστρέφει την αναμενόμενη διάρκεια της κράτησης.
* getReservationSpot(): ParkingSpot : Επιστρέφει την θέση την οποία αφορά η κράτηση.
* getReservationUser(): User : Επιστρέφει τον χρήστη που έκανε την κράτηση.
* getReservationDate(): String : Επιστρέφει την ημερομηνία της κράτησης.

1.1.1.3 Ορισμός Κλάσης ReservationController

A yellow sign with black text

Description automatically generated

**Μέθοδοι κλάσης**

* ReservationController(): void : Η συνάρτηση δόμησης της κλάσης.
* reserveParkingSpot(Reservation Date: String, ReservationStartTime: String, Reservation Expected Duration: float): Reservation : Παίρνει σαν όρισμα την ημερομηνία, την ώρα και την αναμενόμενη διάρκεια και καλεί τον constructor του Reservation, για να δημιουργήσει ένα νέο αντικείμενο κράτησης.
* updateReservation (Reservation Date: String, ReservationStartTime: String, Reservation Expected Duration: float, UserReservation: Reservation): void : Παίρνει σαν όρισμα την ημερομηνία, την ώρα και την αναμενόμενη διάρκεια και καλεί τις κατάλληλες κλάσεις του Reservation, ώστε να ανανεωθούν οι πληροφορίες της κράτησης.
* deleteReservation (UserReservation: Reservation): void : Διαγράφει μια κράτηση.
* makePayment(UserReservation: Reservation): Payment : Καλεί τον constructor του payment, δημιουργώντας ένα νέο αντικείμενο πληρωμής.
* calculateCost(Reservation Expected Duration: Float): Float : Υπολογίζει το κόστος της κράτησης με βάση την αναμενόμενη διάρκεια της κράτησης.

1.1.1.4 Ορισμός Κλάσης MakeReservationGUI

A green box with black text

Description automatically generated**Χαρακτηριστικά κλάσης**

* labelMakeReservation: label : Επιγραφή του κειμένου “Make Reservation”.
* labelDate: label : Επιγραφή του κειμένου “Date:”.
* labelStartTime: label : Επιγραφή του κειμένου “Start time:”.
* labelExpected Duration: label : Επιγραφή του κειμένου “Expected duration:”.
* ReservationDate: String : Η ημερομηνία που εισάγει ο χρήστης.
* StartTime: String : Η ώρα που εισάγει ο χρήστης.
* Expected Duration: String : Η διάρκεια κράτησης που εισάγει ο χρήστης.
* buttonMakeReservation: button : Το κουμπί για την δημιουργία νέας κράτησης.
* buttonBack: button : Το κουμπί επιστροφής στην προηγούμενη σελίδα.

**Μέθοδοι κλάσης**

* MakeReservationGUI(): void : Η συνάρτηση δόμησης του γραφικού.
* buttonMakeReservationPress(): void : Όταν πατηθεί το κουμπί καλεί την συνάρτηση reserveParkingSpot του ReservationController ώστε να δημιουργήσει μια νέα κράτηση.
* buttonBackPress(): void : : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

1.1.1.5 Ορισμός Κλάσης SpotInformationGUI

A screen shot of a computer

Description automatically generated

**Χαρακτηριστικά κλάσης**

* labelSpotinfo: label : Επιγραφή του κειμένου “Spot Information”.
* labelAddress: label : Επιγραφή του κειμένου “Address:”.
* labelSpotType: label : Επιγραφή του κειμένου “Spot type:”.
* Spotlmage: image : Η εικόνα της θέσης πάρκινγκ.
* SpotAddress: String : Η διεύθυνση της θέσης πάρκινγκ.
* SpotType: String : Ο τύπος της θέσης πάρκινγκ.
* SpotChargerAvailability: Boolean : Η διαθεσιμότητα φορτιστή της θέσης πάρκινγκ.
* buttonBack: button : Το κουμπί επιστροφής στην προηγούμενη σελίδα.
* buttonReserveSpot: button : Το κουμπί για κράτηση της θέσης.

**Μέθοδοι κλάσης**

* SpotInformationGUI(): void : Η συνάρτηση δόμησης της κλάσης.
* buttonReserveSpotPress(): void : Όταν πατηθεί το κουμπί, ο χρήστης μεταφέρεται στην σελίδα νέας κράτησης.
* buttonBackPress(): void : : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

1.1.1.6 Ορισμός Κλάσης DBProxy

A close-up of a pink box

Description automatically generated

**Μέθοδοι κλάσης**

* getSpot(All: Boolean, Address: String, Type: String, Charger: Boolean): <list> ParkingSpot : Συνάρτηση που παίρνει ορίσματα τα κριτήρια αναζήτησης θέσεων πάρκινγκ και επιστρέφει μια λίστα θέσεων που πληρούν τα κριτήρια.
* getLicensePlate(User: User): License Plate : Συνάρτηση που παίρνει σαν όρισμα τον χρήστη και επιστρέφει την πινακίδα που του ανήκει.
* getReservation(User: User): <list> Reservation : Συνάρτηση που παίρνει σαν όρισμα τον χρήστη και επιστρέφει μια λίστα από κρατήσεις του χρήστη.
* getSpotOwner(SpotOwnerID: Integer): SpotOwner : Συνάρτηση που παίρνει σαν όρισμα το ID του ιδιοκτήτη θέσης και επιστρέφει το αντικείμενο SpotOwner.
* getUser(UserID: Integer): User : Συνάρτηση που παίρνει σαν όρισμα το ID του χρήστη και επιστρέφει το αντικείμενο User.

1.1.2 Διάγραμμα κλάσεων

A diagram of a computer

Description automatically generated with medium confidence

1.2 Πακέτο κλάσεων Feature 2: Manage Reservation

1.2.1 Ορισμός Κλάσεων

1.2.1.1 Ορισμός Κλάσης Manage Reservation GUI

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

**Χαρακτηριστικά κλάσης**

* labelManageReservation: label: Επιγραφή του κειμένου “Manage Reservation”
* buttonUpdateReservation: button: Κουμπί για ενημέρωση μιας ήδη υπάρχουσας κράτησης.
* buttonDeleteReservation: button: Κουμπί για διαγραφή μιας κράτησης.
* buttonBack: button : Το κουμπί επιστροφής στην προηγούμενη σελίδα.

**Μέθοδοι κλάσης**

* ManageReservationGUI(): void: Η συνάρτηση δόμησης της σελίδας για διαχείριση κρατήσεων.
* buttonUpdateReservationPress(): void: Όταν πατηθεί το κουμπί μεταφέρει τον χρήστη στη σελίδα για τροποποίηση της κράτησής του.
* buttonDeleteReservationPress(): void: Όταν πατηθεί το κουμπί μεταφέρει τον χρήστη στη σελίδα για διαγραφή της κράτησής του.
* buttonBackPress(): void : : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

1.2.1.2 Ορισμός Κλάσης Update Reservation GUI

Εικόνα που περιέχει κείμενο, γραμματοσειρά, στιγμιότυπο οθόνης, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα

**Χαρακτηριστικά κλάσης**

* labelUpdateReservation: label: Επιγραφή με το κείμενο “Update Reservation”
* labelDate: label: Επιγραφή με το κείμενο “Date”
* labelStartTime: Επιγραφή με το κείμενο “Start Time”
* labelExpectedDuration: label: Επιγραφή με το κείμενο “Expected Duration”
* UpdateReservationDate: String: Η ενημερωμένη ημερομηνία κράτησης
* UpdateStartTime: String: Η ενημερωμένη ημερομηνία έναρξης
* UpdateExpectedDuration: Η ενημερωμένη διάρκεια της κράτησης
* buttonSaveChanges: button: Κουμπί για την αποθήκευση των αλλαγών στις λεπτομέρειες κράτησης.
* buttonBack: button : Το κουμπί επιστροφής στην προηγούμενη σελίδα.

**Μέθοδοι κλάσης**

* UpdateReservationGUI(): void: Η συνάρτηση δόμησης της οθόνης για ενημέρωση των στοιχείων κράτησης.
* buttonSaveChangePress(): void: Όταν πατηθεί το κουμπί Save Changes η εν λόγω συνάρτηση αποθηκεύει τις αλλαγές που ο χρήστης πραγματοποίησε στην κράτησή του.
* buttonBackPress(): void : : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

1.2.1.2 Ορισμός Κλάσης Update Reservation GUI

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

**Χαρακτηριστικά κλάσης**

* labelDeleteReservation: label: Επιγραφή με το κείμενο “Delete Reservation”
* labelShowCurrentReservations: label: Επιγραφή με το κείμενο “Show Current Reservations”
* labelReservationDate: label: Επιγραφή με το κείμενο “Reservation Date”
* labelReservationStartTime: label: Επιγραφή με το κείμενο “Reservation Start Time”
* labelReservationExpectedDuration: label: Επιγραφή με το κείμενο “Reservation Expected Duration”
* ReservedSpotImage: image: Εικόνα που δείχνει την θέση πάρκινγκ που ο χρήστης ενοικιάζει.
* buttonDeleteReservation: Κουμπί για διαγραφή κράτησης
* buttonBack: button : Το κουμπί επιστροφής στην προηγούμενη σελίδα.
* ReservedParkingSpotsList: <list> Parking Spot: Λίστα με όλες τις κρατήσεις του χρήστη

**Μέθοδοι κλάσης**

* DeleteReservationGUI(): void: Η συνάρτηση δόμησης της σελίδας για διαγραφή κρατήσεων
* buttonDeleteReservationPress(): void: Όταν πατηθεί το κουμπί, η συνάρτηση αυτή μεταφέρει τον χρήστη στην επόμενη σελίδα για να διαγράψει κάποια κράτησή του
* buttonBackPress(): void : : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

1.2.2 Διάγραμμα κλάσεων

A diagram of a system

Description automatically generated

1.3 Πακέτο κλάσεων Feature 3: Manage parking spots

1.3.1 Ορισμός Κλάσεων

1.3.1.1 Ορισμός Κλάσης Spot Owner

Εικόνα που περιέχει κείμενο, γραμματοσειρά, λογισμικό, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

**Χαρακτηριστικά κλάσης**

* SpotOwnerID: Integer: Αριθμός ταυτοποίησης του ιδιοκτήτη θέσεων πάρκινγκ.
* SpotOwnerName: String: Το όνομα του ιδιοκτήτη θέσεων πάρκινγκ.
* SpotOwnerIDnumber: String: Ο αριθμός ταυτότητας του ιδιοκτήτη θέσεων πάρκινγκ.
* SpotOwnerEmail: String: Η διεύθυνση E-mail του ιδιοκτήτη θέσεων πάρκινγκ.
* SpotOwnerTelephone: String: Ο αριθμός τηλεφώνου του ιδιοκτήτη θέσεων πάρκινγκ.
* MySpots: list<ParkingSpot>: Λίστα με τις θέσεις πάρκινγκ που ανήκουν στον ιδιοκτήτη.

**Μέθοδοι κλάσης**

* SpotOwner(SpotOwnerName:String, SpotOwnerIDnumber:String, SpotOwnerEmail:String, SpotOwnerTelephone:String, MySpots:list<ParkingSpot>, SpotOwnerID:Integer): void: Η συνάρτηση δόμησης της κλάσης. Παίρνει τα κατάλληλα ορίσματα και δημιουργεί ένα αντικείμενο SpotOwner.
* SetSpotOwnerName(SpotOwnerName: String): void: Παίρνει σαν όρισμα το όνομα του ιδιοκτήτη θέσεων και το ορίζει στα στοιχεία του.
* SetSpotOwnerIDnumber(SpotOwnerIDnumber: String): void: Παίρνει σαν όρισμα τον αριθμό ταυτοποίησης του ιδιοκτήτη και τον ορίζει στα στοιχεία του.
* SetSpotOwnerEmail(SpotOwnerEmail: String): void: Παίρνει σαν όρισμα τη διεύθυνση E-mail του ιδιοκτήτη και την ορίζει στα στοιχεία του.
* SetSpotOwnerTelephone(SpotOwnerTelephone: String): void: Παίρνει σαν όρισμα το τηλέφωνο του ιδιοκτήτη και το ορίζει στα στοιχεία του.
* SetMySpots(MySpots: list<ParkingSpot>): void: Παίρνει σαν όρισμα μια λίστα με τις θέσεις πάρκινγκ που ανήκουν στον ιδιοκτήτη και τις καταχωρεί στο σύστημα.
* SetSpotOwnerID(SpotOwnerID: Integer): void: Παίρνει σαν όρισμα τον αριθμό ταυτότητας του ιδιοκτήτη θέσεων και την ορίζει στα στοιχεία του.
* GetSpotOwnerName(): String: Επιστρέφει το όνομα του ιδιοκτήτη θέσεων πάρκινγκ.
* GetSpotOwnerIDNumber(): String: Επιστρέφει τον αριθμό ταυτότητας του ιδιοκτήτη θέσεων πάρκινγκ.
* GetSpotOwnerEmail(): String: Επιστρέφει την διεύθυνση E-mail του ιδιοκτήτη θέσεων πάρκινγκ.
* GetSpotOwnerTelephone(): String: Επιστρέφει τον αριθμό τηλεφώνου του ιδιοκτήτη θέσεων πάρκινγκ.
* GetMySpots(): list<ParkingSpot>: Επιστρέφει την λίστα με τις θέσεις πάρκινγκ που ανήκουν στον ιδιοκτήτη.
* GetSpotOwnerID(): Integer: Επιστρέφει τον αριθμό ταυτοποίησης του ιδιοκτήτη θέσεων πάρκινγκ.

1.3.1.2 Ορισμός Κλάσης Parking Spot Controller

**Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, κίτρινο

Περιγραφή που δημιουργήθηκε αυτόματα**

**Μέθοδοι κλάσης**

* ParkingSpotController(): void: : Η συνάρτηση δόμησης της κλάσης.
* ViewAllAvailableSpots(): list<ParkingSpot>: Επιστρέφει μια λίστα με όλες τις διαθέσιμες θέσεις πάρκινγκ.
* SearchBySpotAddress(spotAddress: String): list<ParkingSpot>: Παίρνει σαν όρισμα τη διεύθυνση κοντά στην οποία αναζητούνται θέσεις πάρκινγκ και επιστρέφει μια λίστα με αντικείμενα που πληρούν το κριτήριο αναζήτησης.
* SearchBySpotType(spotType: String): list<ParkingSpot>: Παίρνει σαν όρισμα έναν τύπο θέσης πάρκινγκ και επιστρέφει μια λίστα αντικειμένων που πληρούν το κριτήριο αναζήτησης.
* SearchBySpotCharger(): list<ParkingSpot>: Επιστρέφει μια λίστα με θέσεις πάρκινγκ οι οποίες διαθέτουν ηλεκτρικό φορτιστή.
* AddParkingSpot(spotAddress:String, spotType:String, spotCharger:Boolean): ParkingSpot: Παίρνει σαν όρισμα την διεύθυνση, τον τύπο θέσης και την ύπαρξη ηλεκτρικού φορτιστή και καλεί τον constructor του Parking Spot, για να δημιουργήσει ένα νέο αντικείμενο θέσης.
* ModifyParkingSpot(spotAddress:String, spotType:String, spotCharger:Boolean, SpotOwnerParkingSpot: ParkingSpot):void: Παίρνει σαν όρισμα την διεύθυνση, τον τύπο θέσης, την ύπαρξη ηλεκτρικού φορτιστή και ….. (δεν ξέρω πώς να το πω), για να τροποποιήσει τις λεπτομέρειες μιας ήδη καταχωρημένης θέσης πάρκινγκ.
* RemoveParkingSpot( SpotOwnerParkingSpot: ParkingSpot): void: Δέχεται σαν όρισμα ένα αντικείμενο ParkingSpot και το διαγράφει από την εφαρμογή.

1.3.1.3 Ορισμός Κλάσης Modify Spot Information GUI

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα**Χαρακτηριστικά κλάσης**

* labelModifyInfo: label: Επιγραφή του κειμένου “Modify Info”.
* labelAddress: label: Επιγραφή του κειμένου “Address”.
* labelSpotType: label: Επιγραφή του κειμένου “Spot Type”.
* labeCharger: label: Επιγραφή του κειμένου “Charger”.
* textAddress: textbox: Πλαίσιο κειμένου στο οποίο ο ιδιοκτήτης εισάγει την διεύθυνση της προς ενοικίαση θέσης.
* textType: textbox: Πλαίσιο κειμένου στο οποίο ο ιδιοκτήτης εισάγει τον τύπο της προς ενοικίαση θέσης.
* sliderCharger: slider: Toggle button με το οποίο ο ιδιοκτήτης καταχωρεί εάν η προς ενοικίαση θέση διαθέτει ηλεκτρικό φορτιστή.
* buttonSaveChanges: button: Το κουμπί για αποθήκευση των αλλαγών στην καταχωρημένη θέση πάρκινγκ.
* buttonBack: button: Το κουμπί για επιστροφή στην προηγούμενη σελίδα.

**Μέθοδοι κλάσης**

* ModifySpotInformationGUI(): void: Η συνάρτηση δόμησης της σελίδας Modify Spot Information.
* buttonSaveChangesPress(): void: Όταν πατηθεί το κουμπί “Save Changes” αποθηκεύονται οι πληροφορίες για την καταχωρημένη θέση πάρκινγκ.
* buttonBackPress(): void : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

1.3.1.3 Ορισμός Κλάσης Remove Spot GUI

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, αριθμός

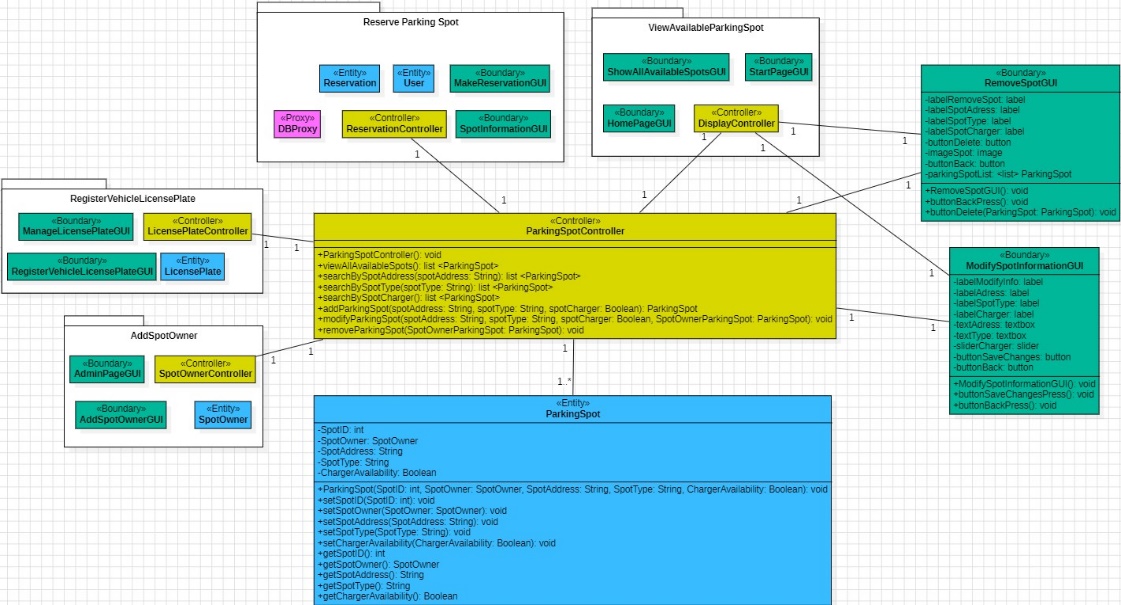
Περιγραφή που δημιουργήθηκε αυτόματα**Χαρακτηριστικά κλάσης**

* labelRemoveSpot: label: Επιγραφή του κειμένου “Remove Spot”
* labelSpotAddress: label: Επιγραφή του κειμένου “Spot Address”
* labelSpotType: label: Επιγραφή του κειμένου “Spot Type”
* labelSpotCharger: label: Επιγραφή του κειμένου “Spot Charger”
* buttonDelete: button: Το κουμπί για διαγραφή μιας καταχωρημένης θέσης.
* imageSpot: image: Εικόνα της θέσης πάρκινγκ.
* buttonBack: button: Το κουμπί για επιστροφή στην προηγούμενη σελίδα.
* parkingSpotList: <list> ParkingSpot: Λίστα αντικειμένων ParkingSpot τα οποία ανήκουν στον ίδιο SpotOwner.

**Μέθοδοι κλάσης**

* RemoveSpotGUI(): void: Η συνάρτηση δόμησης της σελίδας για διαγραφή θέσεων.
* buttonBackPress(): void : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.
* buttonDelete(ParkingSpot: ParkingSpot): void: Δέχεται σαν όρισμα μια θέση πάρκινγκ και την διαγράφει από το σύστημα.

1.3.2 Διάγραμμα κλάσεων



1.4 Πακέτο κλάσεων Feature 4: View available parking spots

1.4.1 Ορισμός Κλάσεων

Το πακέτο ViewAvailableParkingSpot υλοποιεί το Gherkin Feature 4. Πιο συγκεκριμένα περιέχει κλάσεις που σχετίζονται με την προβολή των διαθέσιμων θέσεων πάρκινγκ και με την προβολή όλων των οθονών της εφαρμογής.

1.4.1.1 Ορισμός Κλάσης ShowAllAvailableSpotsGUI

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

Η κλάση αυτή εκφράζει την διεπαφή της σελίδας προβολής όλων των διαθέσιμων θέσεων πάρκινγκ.

***Χαρακτηριστικά Κλάσης***

* labelShowAllAvailableSpots: label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “Showing all available spots”.
* labelAddress:label - Εκφράζει την διεύθυνση της διαθέσιμης θέσης πάρκινγκ.
* labelSpotType:label - Εκφράζει τον τύπο της διαθέσιμης θέσης πάρκινγκ.
* labelCharger:label - Εκφράζει την διαθεσιμότητα ηλεκτρικού φορτιστή των διαθέσιμων θέσεων πάρκινγκ.
* AvailableSpotsImage:image - Αποτελεί τις εικόνες των διαθέσιμων θέσεων πάρκινγκ.
* buttonSelectAvailableSpot:button – Είναι το βελάκι δεξιά από κάθε διαθέσιμη θέση. Πατάω αυτό το κουμπί για να διαλέξω θέση.
* buttonBack:button - Είναι το βελάκι πάνω αριστερά στη διεπαφή. Πατάω αυτό το κουμπί για να επιστρέψω στην προηγούμενη σελίδα.
* AvailableParkingSpotsList:list - Είναι η λίστα όλων των διαθέσιμων θέσεων πάρκινγκ που υπάρχουν στο σύστημα. Σε αυτή περιέχονται όλες οι θέσεις που εμφανίζονται στη διεπαφή.

***Μέθοδοι κλάσης***

* ShowAllAvailableSpotsGUI(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
* buttonSelectAvailableSpotPress():void - Καλεί τη displaySpotInformationGUI():void της κλάσης DisplayController, η οποία προβάλλει τις πληροφορίες της επιλεγμένης θέσης.
* buttonBackPress():void - Καλεί τη displayHomePage(): void της κλάσης DisplayController, η οποία προβάλλει την κεντρική οθόνη του χρήστη.

1.4.1.2 Ορισμός Κλάσης HomePageGUI

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, έγγραφο

Περιγραφή που δημιουργήθηκε αυτόματα

Η κλάση αυτή εκφράζει την διεπαφή της κεντρικής οθόνης του χρήστη.

***Χαρακτηριστικά Κλάσης***

* labelWelcome: label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “Welcome”.
* labelUsername: label - Χρησιμοποιείται για την εμφάνιση του ονόματος του χρήστη.
* textSearch: textbox - Εκφράζει την διεύθυνση που πληκτρολογεί ο χρήστης στο search bar.
* buttonSearch: button – Είναι ο μεγεθυντικός φακός του search bar. Ο χρήστης πατάει αυτό το κουμπί για να αναζητήσει διαθέσιμες θέσεις πάρκινγκ κοντά στην διεύθυνση που πληκτρολόγησε.
* buttonAllSpots: button – Είναι το κουμπί “All available parking spots”. Ο χρήστης πατάει αυτό το κουμπί για να δει όλες τις διαθέσιμες θέσεις πάρκινγκ που υπάρχουν στο σύστημα.
* buttonSearchByType: button - Είναι το κουμπί “Search by type”. Ο χρήστης πατάει αυτό το κουμπί για να αναζητήσει διαθέσιμες θέσεις πάρκινγκ ενός συγκεκριμένου τύπου.
* buttonSpotsWithCharger: button - Είναι το κουμπί “Show only spots with charger”. Ο χρήστης πατάει αυτό το κουμπί για να αναζητήσει διαθέσιμες θέσεις πάρκινγκ που είναι εξοπλισμένες με ηλεκτρικό φορτιστή οχημάτων.
* buttonMakePayment: button - Είναι το κουμπί “Make payment”. Ο χρήστης πατάει αυτό το κουμπί για να πληρώσει τις κρατήσεις του ή να εξοφλήσει τυχόν υπόλοιπο εντός της εφαρμογής.
* buttonManageReservation: button – Είναι το κουμπί “Manage Reservation”. Ο χρήστης πατάει αυτό το κουμπί για να διαχειριστεί τις κρατήσεις του.
* buttonManageLicensePlate: button - Είναι το κουμπί “Manage licence plate”. Ο χρήστης πατάει αυτό το κουμπί για να καταχωρήσει την πινακίδα του οχήματός του στο σύστημα ή για να επεξεργαστεί την ήδη καταχωρημένη πινακίδα του.
* buttonHelp: button – Είναι το κουμπί με σύμβολο ένα αγγλικό ερωτηματικό(βρίσκεται πάνω δεξιά στην οθόνη).Είναι το κουμπί βοήθειας. Προσφέρει πληροφορίες στους χρήστες προκειμένου να κατανοήσουν τις βασικές λειτουργίες του συστήματος.

***Μέθοδοι κλάσης***

* HomePageGUI(User user): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
* buttonSearchPress(): String - Καλεί τη

searchBySpotAddress(String spotAddress): list <ParkingSpot> του ParkingSpotController, η οποία βρίσκει διαθέσιμες θέσεις πάρκινγκ που βρίσκονται κοντά στην διεύθυνση spotAddress. Επιστρέφει μια λίστα με θέσεις πάρκινγκ. Το όρισμα εισόδου της searchBySpotAddress είναι το String που επιστρέφει η buttonSearchPress() και ταυτίζεται με την διεύθυνση που πληκτρολόγησε ο χρήστης.

* buttonAllSpotsPress(): void - Καλεί τη viewAllAvailableSpots(): list <ParkingSpot> του

ParkingSpotController, η οποία βρίσκει όλες τις διαθέσιμες θέσεις πάρκινγκ του συστήματος και τις επιστρέφει σε μία λίστα.

* buttonSearchByTypePress():void - Καλεί τη

displaySearchByTypePage(): void του DisplayController, η οποία προβάλλει την οθόνη αναζήτησης διαθέσιμης θέσης με βάση τον τύπο.

* buttonSpotsWithChargerPress(): void – Καλεί τη searchBySpotCharger(): list <ParkingSpot> του ParkingSpotController, η οποία βρίσκει όλες τις διαθέσιμες θέσεις πάρκινγκ που είναι εξοπλισμένες με φορτιστή και τις επιστρέφει σε μία λίστα.
* buttonMakePaymentPress(): void – Καλεί τη displayPaymentScreen(): void του DisplayController, η οποία προβάλλει την οθόνη προβολής.
* buttonManageReservationPress(): void – Καλεί τη displayManageReservationPage(): void του

DisplayController, η οποία προβάλλει την οθόνη διαχείρισης κράτησης.

* buttonManageLicensePlatePress(): void – Καλεί τη displayManageLicensePlatePage(): void του DisplayController, η οποία προβάλλει την οθόνη διαχείρισης πινακίδας οχήματος.
* buttonHelpPress(): void – Προβάλλει χρήσιμες πληροφορίες για τις βασικές λειτουργίες της εφαρμογής.

1.4.1.3 Ορισμός Κλάσης StartPageGUI

Εικόνα που περιέχει κείμενο, γραμματοσειρά, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Η κλάση αυτή εκφράζει την διεπαφή της αρχικής οθόνης της εφαρμογής

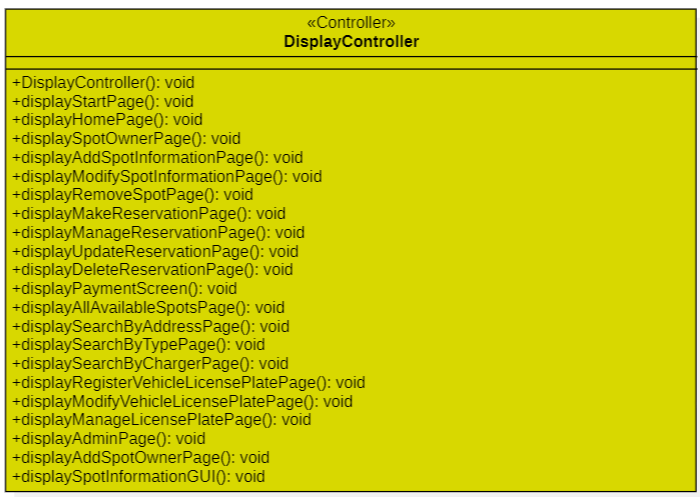
***Χαρακτηριστικά Κλάσης***

* imageLogo: image – Είναι το λογότυπο της εφαρμογής.
* labelName: label – Χρησιμοποιείται για την εμφάνιση του ονόματος της εφαρμογής(“CurbSprings”).
* labelMotto: label - Χρησιμοποιείται για την εμφάνιση του motto της εφαρμογής(“Your reliable partner in urban mobility”).
* buttonParkNow: button – Είναι το κουμπί “ParkNow”. Οι χρήστες πατούν αυτό το κουμπί για να χρησιμοποιήσουν την εφαρμογή.
* buttonAdmin: button - Είναι το κουμπί “I am an admin”. Ο διαχειριστής πατά αυτό το κουμπί για να χρησιμοποιήσει την εφαρμογή.
* buttonSpotOwner: button - Είναι το κουμπί “I am a spot owner”. Οι ιδιοκτήτες θέσεων πάρκινγκ πατούν αυτό το κουμπί για να χρησιμοποιήσουν την εφαρμογή.

***Μέθοδοι κλάσης***

* StartPageGUI(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
* ParkNowPress(): void - Καλεί τη displayHomePage(): void της κλάσης DisplayController, η οποία προβάλλει τη κεντρική οθόνη του χρήστη.
* IamAdminPress(): void - Καλεί τη displayAdminPage(): void της κλάσης DisplayController, η οποία προβάλλει τη κεντρική οθόνη του διαχειριστή του συστήματος.
* IamSpotOwnerPress(): void – Καλεί displaySpotOwnerPage(): void της κλάσης DisplayController, η οποία προβάλλει τη κεντρική οθόνη του ιδιοκτήτη θέσης.

1.4.1.4 Ορισμός Κλάσης DisplayController

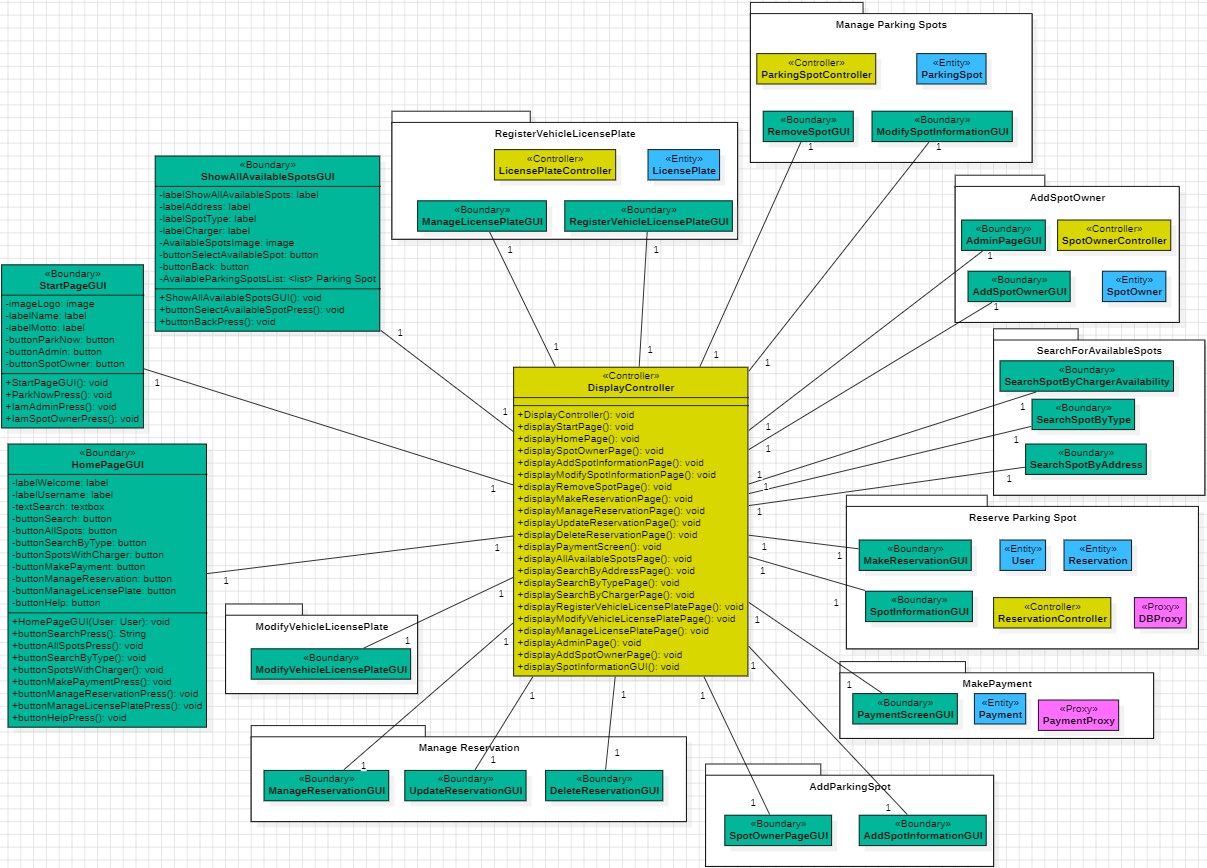


Η κλάση αυτή είναι υπεύθυνη για την πλοήγηση των χρηστών, των ιδιοκτητών θέσεων και του διαχειριστή στην εφαρμογή.

***Μέθοδοι κλάσης***

* DisplayController(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
* displayStartPage(): void – Προβάλλει την αρχική οθόνη της εφαρμογής.
* displayHomePage(): void – Προβάλει την κεντρική οθόνη της εφαρμογής.(Δηλαδή την κεντρική οθόνη του χρήστη).
* displaySpotOwnerPage(): void - Προβάλλει τη κεντρική οθόνη του ιδιοκτήτη θέσης.
* displayAddSpotInformationPage(): void – Προβάλλει την οθόνη καταχώρησης των στοιχείων της θέσης.
* displayModifySpotInformationPage(): void – Προβάλλει την οθόνη τροποποίησης των στοιχείων της θέσης.
* displayRemoveSpotPage(): void – Προβάλλει την οθόνη διαγραφής της θέσης.
* displayMakeReservationPage(): void – Προβάλλει την οθόνη κράτησης θέσης.
* displayManageReservationPage(): void – Προβάλλει την οθόνη διαχείρισης της κράτησης.
* displayUpdateReservationPage(): void - Προβάλλει την οθόνη τροποποίησης των πληροφοριών της κράτησης.
* displayDeleteReservationPage(): void – Προβάλλει την οθόνη διαγραφής της κράτησης.
* displayPaymentScreen(): void – Προβάλλει την οθόνη πληρωμής.
* displayAllAvailableSpotsPage(): void – Προβάλλει την οθόνη όλων των διαθέσιμων θέσεων.
* displaySearchByAddressPage(): void – Προβάλλει την οθόνη αναζήτησης διαθέσιμης θέσης με βάση την διεύθυνση.
* displaySearchByTypePage(): void – Προβάλλει την οθόνη αναζήτησης διαθέσιμης θέσης με βάση τον τύπο.
* displaySearchByChargerPage(): void – Προβάλλει την οθόνη αναζήτησης διαθέσιμης θέσης με βάση τη διαθεσιμότητα φορτιστή.
* displayRegisterVehicleLicensePlatePage(): void – Προβάλλει την οθόνη καταχώρησης της πινακίδας του οχήματος.
* displayModifyVehicleLicensePlatePage(): void – Προβάλλει την οθόνη τροποποίησης της πινακίδας του οχήματος.
* displayManageLicensePlatePage(): void – Προβάλλει την οθόνη διαχείρισης της πινακίδας του οχήματος
* displayAdminPage(): void – Προβάλλει την κεντρική οθόνη του διαχειριστή του συστήματος.
* displayAddSpotOwnerPage(): void - Προβάλλει την οθόνη καταχώρησης των στοιχείων των ιδιοκτητών των θέσεων.
* displaySpotInformationGUI(): void – Προβάλλει την οθόνη πληροφοριών της θέσης.

1.4.2 Διάγραμμα κλάσεων

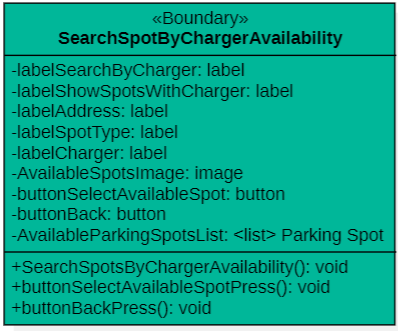


1.5 Πακέτο κλάσεων Feature 5: Search for available parking spots

1.5.1 Ορισμός Κλάσεων

Το πακέτο SearchForAvailableSpots υλοποιεί το Gherkin Feature 5. Πιο συγκεκριμένα περιέχει κλάσεις που σχετίζονται με την αναζήτηση διαθέσιμων θέσεων πάρκινγκ με βάση την διεύθυνσή τους, τον τύπο τους καθώς και την διαθεσιμότητα ηλεκτρικού φορτιστή.

1.5.1.1 Ορισμός Κλάσης SearchSpotByChargerAvailability

Η κλάση αυτή εκφράζει την διεπαφή της σελίδας επιλογής διαθέσιμων θέσεων πάρκινγκ που είναι εξοπλισμένες με ηλεκτρικό φορτιστή οχημάτων.

***Χαρακτηριστικά Κλάσης***

* labelSearchByCharger:label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “Search by Charger”.
* labelShowSpotsWithCharger:label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “Showing spots with charger!”.
* labelAddress:label - Εκφράζει την διεύθυνση της διαθέσιμης θέσης πάρκινγκ που είναι εξοπλισμένη με ηλεκτρικό φορτιστή.
* labelSpotType:label - Εκφράζει τον τύπο της διαθέσιμης θέσης πάρκινγκ που είναι εξοπλισμένη με ηλεκτρικό φορτιστή.
* labelCharger:label - Επισημαίνει την ύπαρξη ηλεκτρικού φορτιστή στη διαθέσιμη θέση πάρκινγκ.
* AvailableSpotsImage:image - Αποτελεί τις εικόνες των διαθέσιμων θέσεων πάρκινγκ.
* buttonSelectAvailableSpot:button - Είναι το βελάκι δεξιά από κάθε διαθέσιμη θέση. Πατάω αυτό το κουμπί για να διαλέξω θέση.
* buttonBack:button - Είναι το βελάκι πάνω αριστερά στη διεπαφή. Πατάω αυτό το κουμπί για να επιστρέψω στην προηγούμενη σελίδα.
* AvailableParkingSpotsList:list - Είναι η λίστα των διαθέσιμων θέσεων που είναι εξοπλισμένες με ηλεκτρικό φορτιστή. Σε αυτή περιέχονται όλες οι θέσεις που εμφανίζονται στη διεπαφή.

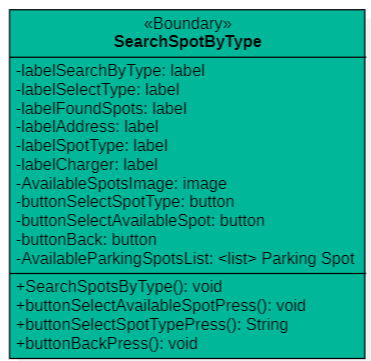
***Μέθοδοι κλάσης***

* SearchSpotsByChargerAvailability():void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
* buttonSelectAvailableSpotPress():void - Καλεί τη displaySpotInformationGUI():void της κλάσης DisplayController, η οποία προβάλλει τις πληροφορίες της επιλεγμένης θέσης.
* buttonBackPress():void - Καλεί τη displayHomePage():void της κλάσης DisplayController.

**1.5.1.2 Ορισμός Κλάσης SearchSpotByType**

Η κλάση αυτή εκφράζει την διεπαφή της σελίδας επιλογής διαθέσιμων θέσεων πάρκινγκ συγκεκριμένου τύπου.

***Χαρακτηριστικά Κλάσης***

* labelSearchByType:label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “Search by type”.
* labelSelectType:label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “Select a specific type:”.
* labelFoundSpots:label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “We found some spots!”.
* labelAddress:label - Εκφράζει την διεύθυνση των διαθέσιμων θέσεων πάρκινγκ.
* labelSpotType:label - Εκφράζει τον τύπο των διαθέσιμων θέσεων πάρκινγκ που επιλέχθηκε από τον χρήστη.
* labelCharger: label – Εκφράζει την διαθεσιμότητα ηλεκτρικού φορτιστή των διαθέσιμων θέσεων πάρκινγκ.
* AvailableSpotsImage:image - Αποτελεί τις εικόνες των διαθέσιμων θέσεων πάρκινγκ.
* buttonSelectSpotType:button – Με αυτό το κουμπί, ο χρήστης επιλέγει τον τύπο των θέσεων πάρκινγκ που επιθυμεί(Open, Underground,Garage).
* buttonSelectAvailableSpot:button - Είναι το βελάκι δεξιά από κάθε διαθέσιμη θέση. Πατάω αυτό το κουμπί για να διαλέξω θέση.
* buttonBack:button - Είναι το βελάκι πάνω αριστερά στη διεπαφή. Πατάω αυτό το κουμπί για να επιστρέψω στην προηγούμενη σελίδα.
* AvailableParkingSpotsList:list - Είναι η λίστα των διαθέσιμων θέσεων πάρκινγκ ενός συγκεκριμένου τύπου. Σε αυτή περιέχονται όλες οι θέσεις που εμφανίζονται στη διεπαφή.

***Μέθοδοι κλάσης***

* SearchSpotsByType(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
* buttonSelectSpotTypePress(): String – Καλεί τη

searchBySpotType(String spotType): list <ParkingSpot> του ParkingSpotController, η οποία βρίσκει όλες τις διαθέσιμες θέσεις πάρκινγκ που έχουν τύπο spotType και τις επιστρέφει σε μία λίστα. Η buttonSelectSpotTypePress() επιστρέφει τον τύπο θέσης που επέλεξε ο χρήστης σε μορφή String. Αυτό το String αποτελεί όρισμα εισόδου της searchBySpotType(String spotType).

* buttonSelectAvailableSpotPress():void - Καλεί τη displaySpotInformationGUI():void της κλάσης DisplayController, η οποία προβάλλει τις πληροφορίες της επιλεγμένης θέσης.
* buttonBackPress():void - Καλεί τη displayHomePage():void της κλάσης DisplayController, η οποία προβάλλει την κεντρική οθόνη του χρήστη.

**1.5.1.3 Ορισμός Κλάσης SearchSpotByAddress**

Η κλάση αυτή εκφράζει την διεπαφή της σελίδας επιλογής διαθέσιμων θέσεων πάρκινγκ που βρίσκονται κοντά σε μια συγκεκριμένη διεύθυνση.

***Χαρακτηριστικά Κλάσης***

* labelSearchByAddress:label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “Search by

address”.

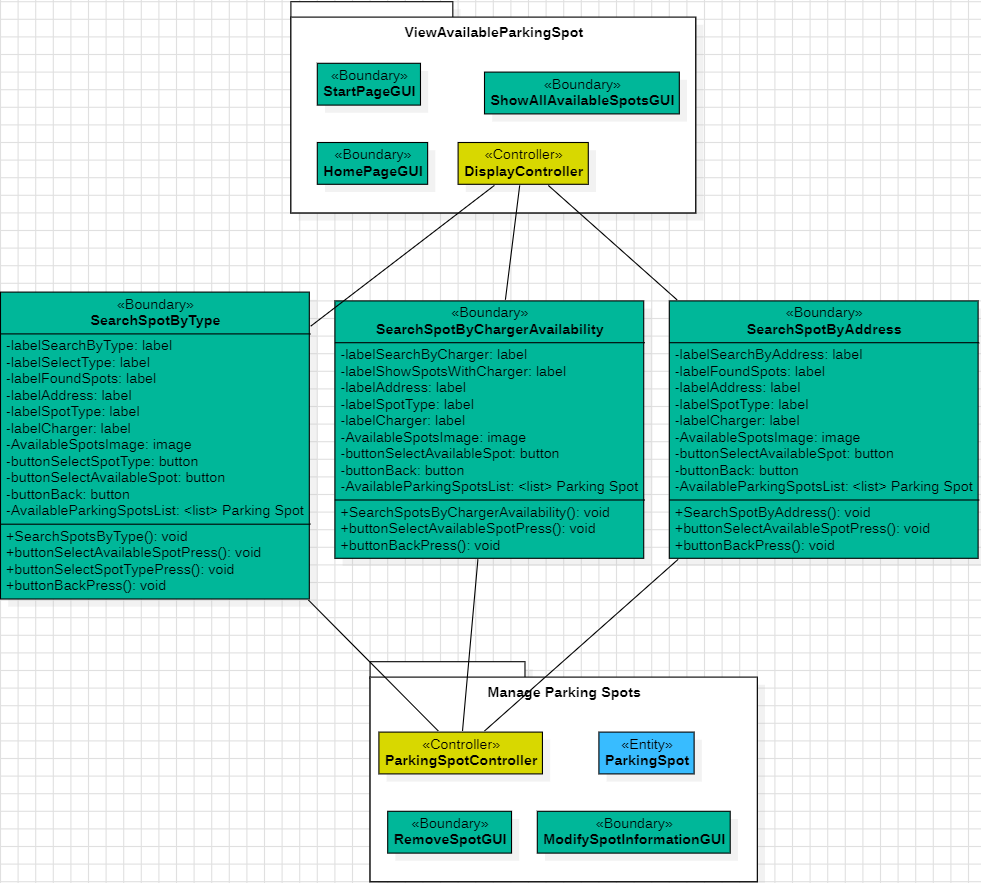
* labelFoundSpots:label - Χρησιμοποιείται για την εμφάνιση του μηνύματος “We found some spots!”.
* labelAddress:label - Εκφράζει την διεύθυνση της διαθέσιμης θέσης πάρκινγκ.
* Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

  Περιγραφή που δημιουργήθηκε αυτόματαlabelSpotType:label - Εκφράζει τον τύπο της διαθέσιμης θέσης πάρκινγκ.
* labelCharger:label - Εκφράζει την διαθεσιμότητα ηλεκτρικού φορτιστή των διαθέσιμων θέσεων πάρκινγκ.
* AvailableSpotsImage:image - Αποτελεί τις εικόνες των διαθέσιμων θέσεων πάρκινγκ.
* buttonSelectAvailableSpot:button - Είναι το βελάκι δεξιά από κάθε διαθέσιμη θέση. Πατάω αυτό το κουμπί για να διαλέξω θέση.
* buttonBack:button - Είναι το βελάκι πάνω αριστερά στη διεπαφή. Πατάω αυτό το κουμπί για να επιστρέψω στην προηγούμενη σελίδα.
* AvailableParkingSpotsList:list - Είναι η λίστα των διαθέσιμων θέσεων που βρίσκονται κοντά στην διεύθυνση που πληκτρολόγησε ο χρήστης. Σε αυτή περιέχονται όλες οι θέσεις που εμφανίζονται στη διεπαφή.

***Μέθοδοι κλάσης***

* SearchSpotByAddress(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
* buttonSelectAvailableSpotPress():void - Καλεί τη displaySpotInformationGUI():void της κλάσης DisplayController, η οποία προβάλλει τις πληροφορίες της επιλεγμένης θέσης.
* buttonBackPress():void - Καλεί τη displayHomePage():void της κλάσης DisplayController, η οποία προβάλλει την κεντρική οθόνη του χρήστη.

1.5.2 Διάγραμμα κλάσεων

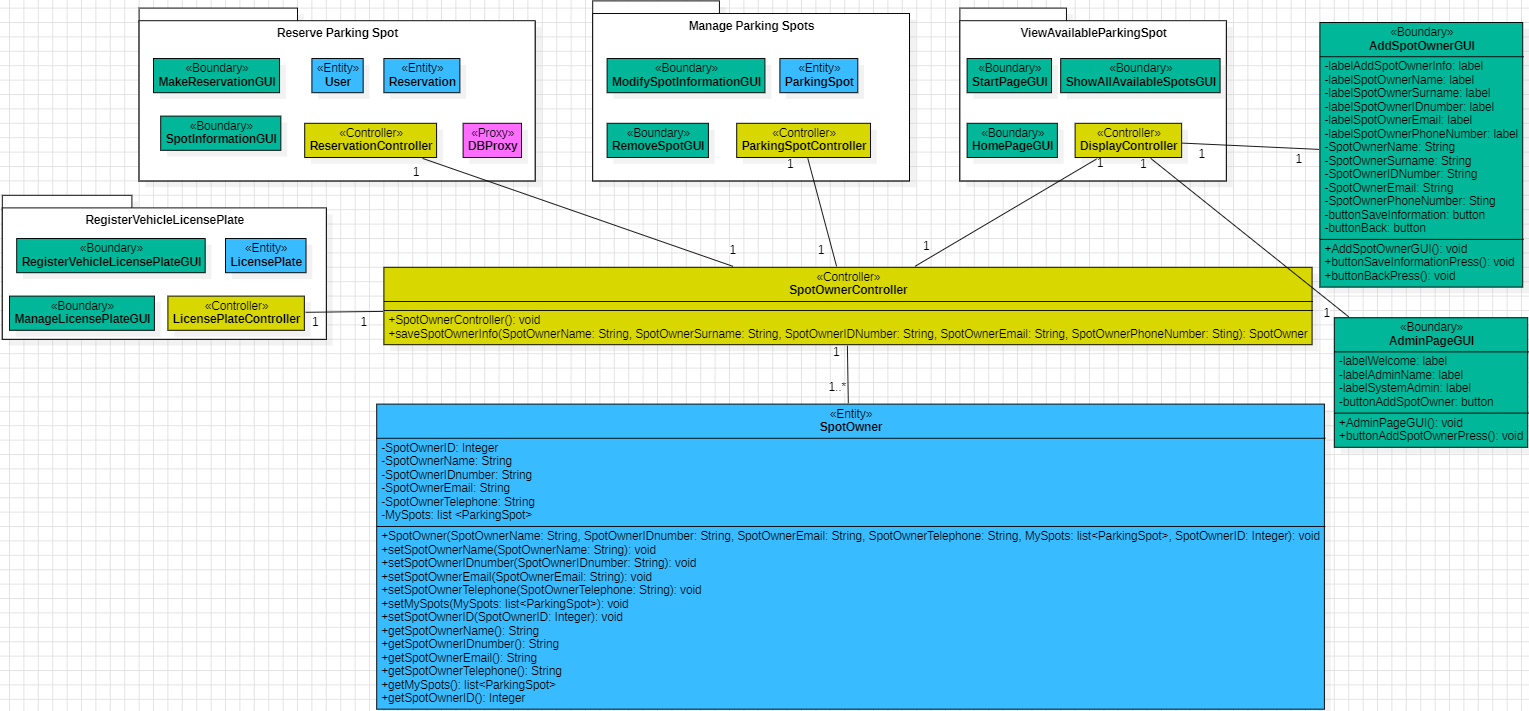


1.6 Πακέτο κλάσεων Feature 6: Add spot owner

1.6.1 Ορισμός Κλάσεων

1.6.1.1 Ορισμός Κλάσης <Όνομα Κλάσης>

1.6.2 Διάγραμμα κλάσεων

****

1.7 Πακέτο κλάσεων Feature 7: Add parking spots

1.7.1 Ορισμός Κλάσεων

1.7.1.1 Ορισμός Κλάσης Spot Owner Page GUI

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

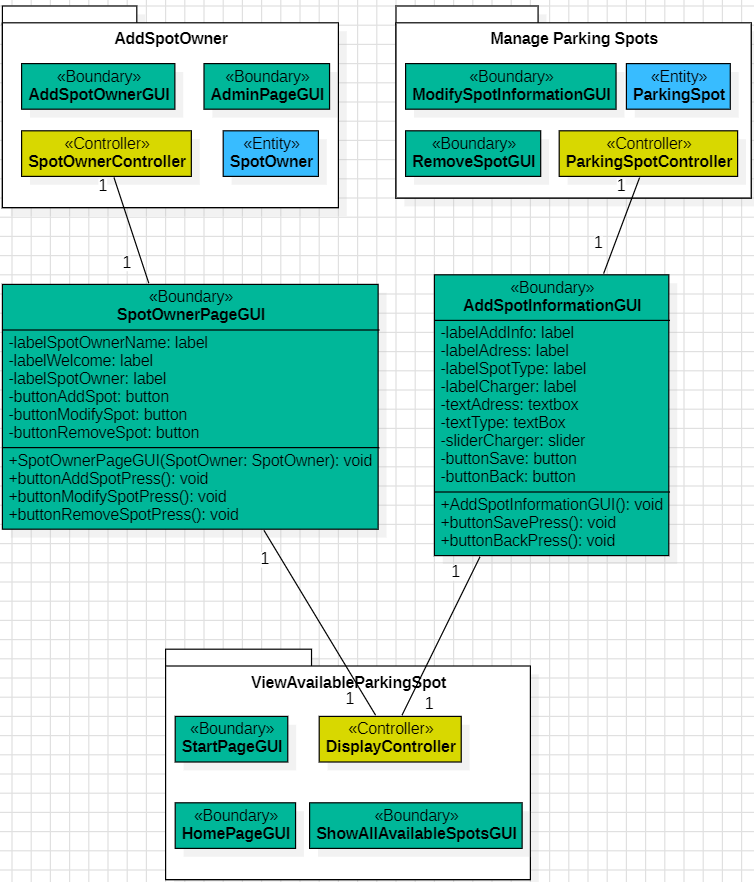
Περιγραφή που δημιουργήθηκε αυτόματα**Χαρακτηριστικά κλάσης**

* labelSpotOwnerName: label: Επιγραφή με το κείμενο “Spot Owner Name”
* labelWelcome: label: Επιγραφή με το κείμενο “Welcome”
* labelSpotOwner: label: Επιγραφή με το κείμενο “Spot Owner”
* buttonAddSpot: button: Κουμπί για την προσθήκη θέσης πάρκινγκ στο σύστημα
* buttonModifySpot: button: Κουμπί για την επεξεργασία στοιχείων ήδη καταχωρημένης θέσης πάρκινγκ
* buttonRemoveSpot: button: Κουμπί για την αφαίρεση μιας θέσης πάρκινγκ από το σύστημα

**Μέθοδοι κλάσης**

* SpotOwnerPageGUI( SpotOwner: SpotOwner): void: Η συνάρτηση δόμησης της σελίδας για την διαχείριση θέσεων. Παίρνει ως όρισμα το όνομα του Spot Owner και το εμφανίζει στο επάνω μέρος της οθόνης.
* buttonAddSpotPress(): void: Όταν πατηθεί το κουμπί, μεταφέρει τον ιδιοκτήτη θέσης στην επόμενη σελίδα για να προσθέσει κι άλλες θέσεις.
* buttonModifySpotPress(): void: Όταν πατηθεί το κουμπί, μεταφέρει τον ιδιοκτήτη στη σελίδα επεξεργασίας των ήδη καταχωρημένων από αυτόν θέσεων.
* buttonRemoveSpotPress(): void: Όταν πατηθεί το κουμπί, μεταφέρει τον ιδιοκτήτη στη σελίδα με τις καταχωρημένες θέσεις του για να επιλέξει ποια θα αφαιρέσει.

1.7.2 Διάγραμμα κλάσεων

****

1.8 Πακέτο κλάσεων Feature 8: Make Payment

1.8.1 Ορισμός Κλάσεων

1.8.1.1 Ορισμός Κλάσης Payment

**Χαρακτηριστικά κλάσης**

PaymentUser: User : Ο χρήστης που πραγματοποιεί την πληρωμή.

PaymentAmount: Float : Το ποσό χρέωσης της πληρωμής.

**Μέθοδοι κλάσης**

A blue screen with black text

Description automatically generatedPayment(PaymentUser: User, PaymentAmount: Float) : Η συνάρτηση δόμησης της κλάσης, παίρνει ως όρισμα τον χρήστη και το ποσό και δημιουργεί το αντικείμενο πληρωμής.

setPaymentUser(PaymentUser: User): void : Η συνάρτηση παίρνει σαν όρισμα ένα αντικέιμενο User και ορίζει τον χρήστη που πραγματοποιεί την πληρωμή.

setPaymentAmount(PaymentAmount:Float): void : Η συνάρτηση παίρνει σαν όρισμα το ποσό και το ορίζει σαν ποσό πληρωμής.

getPaymentUser(): User : Επιστρέφει τον χρήστη που πληρώνει.

getPaymentAmount(): Float : Επιστρέφει το ποσό που πληρώνει ο χρήστης.

1.8.1.2 Ορισμός Κλάσης PaymentScreenGUI

A screen shot of a computer

Description automatically generated

**Χαρακτηριστικά κλάσης**

* labelMakePayment: label : Επιγραφή του κειμένου “Make payment”.
* labelSecureGatewayMessage: label : Επιγραφή του κειμένου “Transfer to secure payment gateway”.
* buttonPayNow: button : Το κουμπί πληρωμής.
* buttonBack: button : Το κουμπί επιστροφής στην προηγούμενη σελίδα.

**Μέθοδοι κλάσης**

* PaymentScreenGUI(): void : Η συνάρτηση δόμησης του γραφικού.
* buttonPayNowPress(): void : Όταν πατηθεί το κουμπί πληρωμής, ο χρήστης μεταφέρεται σε ασφαλές περιβάλλον πληρωμών για να πραγματοποιήσει πληρωμή.
* buttonBackPress(): void : Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

1.8.1.3 Ορισμός Κλάσης PaymentScreenGUI

A close-up of a pink sign

Description automatically generated**Μέθοδοι κλάσης**

* makePayment(Cost:Float): Boolean : Η συνάρτηση πληρωμής παίρνει σαν όρισμα το ποσό πληρωμής και επιστρέφει True αν η πληρωμή πέτυχε ή False αν η πληρωμή απέτυχε.

1.8.2 Διάγραμμα κλάσεων

A diagram of a diagram

Description automatically generated with medium confidence

1.9 Πακέτο κλάσεων Feature 9: Register vehicle license plate

1.9.1 Ορισμός Κλάσεων

1.9.1.1 Ορισμός Κλάσης <Όνομα Κλάσης>

1.9.2 Διάγραμμα κλάσεων

A diagram of a computer

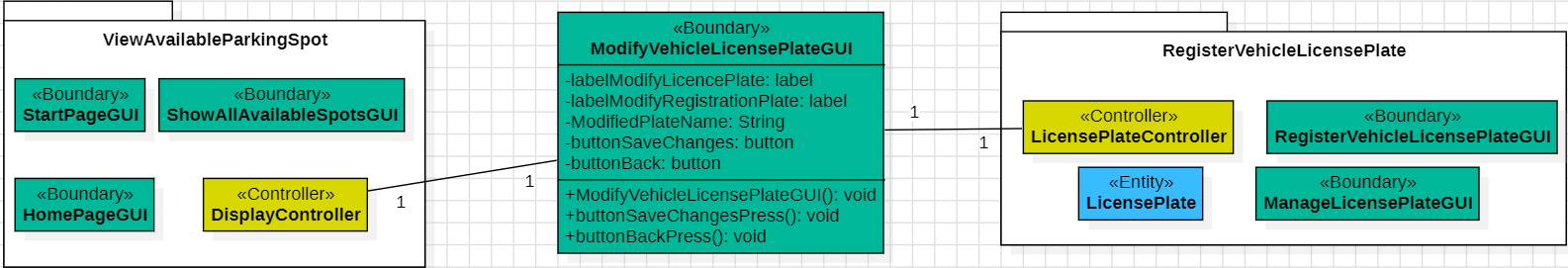
Description automatically generated with medium confidence

1.10 Πακέτο κλάσεων Feature 10: Modify vehicle license plate

1.10.1 Ορισμός Κλάσεων

1.10.1.1 Ορισμός Κλάσης <Όνομα Κλάσης>

1.10.2 Διάγραμμα κλάσεων

****

2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

**Οδηγία:** *Δεν υπάρχει μία μοναδική σχεδίαση για ένα σύστημα λογισμικού. Μελέτη διαφορετικών επιλογών και διαφορετικών προτύπων σχεδίασης είναι αναγκαία. Σε αυτό το μέρος του εγγράφου πρέπει να καθορίσετε τα πρότυπα που θα χρησιμοποιήσετε, ώστε να αντιμετωπίσετε συγκεκριμένα προβλήματα στην αρχιτεκτονική σας, τα οποία καταγράψατε στις Μη Λειτουργικές Απαιτήσεις (και όχι μόνο!!).*

*Χρησιμοποιήστε* ***τρία*** *Πρότυπα σχεδίασης για να ικανοποιήσετε κάποια/κάποιες ΜΛΑ του συστήματός σας. Για κάθε πρότυπο που θα χρησιμοποιήσετε, πρέπει να παρέχετε τις παρακάτω πληροφορίες:*

* *Όνομα*
* *Σχεδιαστική ανάγκη που ικανοποιήθηκε*
* *Εφαρμογή του προτύπου σχεδίασης στις απαιτούμενες κλάσεις. (Δώστε ένα συνοπτικό διάγραμμα κλάσεων το οποίο περιλαμβάνει μόνο τις κλάσεις που προκύπτουν από την εφαρμογή του προτύπου στις κλάσεις που χρειάστηκε. Συμβουλευτείτε τις διαφάνειες του μαθήματος και του εργαστηρίου)*

**Σημείωση:** Ανεξάρτητα από το εάν απαιτούνται παραπάνω πρότυπα σχεδίασης για το σύστημά σας, περιοριστείτε σε τρία. Εάν καμία από τις ΜΛΑ που επιλέξατε δεν ικανοποιείται από κάποιο πρότυπο σχεδίασης, καταγράψτε δύο υποθετικές σχεδιαστικές ανάγκες (π.χ. επιπλέον ΜΛΑ ή ΛΑ) και επιλέξτε τρία πρότυπα σχεδίασης για να τις ικανοποιήσετε.

3 Δυναμική Μοντελοποίηση

**Οδηγία**: Σχεδιάστε 3 διαγράμματα ακολουθιών που αντιστοιχούν στα 3 πιο σημαντικά Gherkin Scenarios (όχι features) του συστήματός σας. Εισάγετε μία ενότητα όπως η παρακάτω για **κάθε Gherkin Scenario** για το οποίο θα φτιάξετε διάγραμμα ακολουθιών.

3.1 Gherkin Scenario X (Feature Y)

**Οδηγία:** Δώστε μία σύντομη περιγραφή για τον τρόπο με τον οποίο αλληλεπιδρούν οι (εμπλεκόμενες στο συγκεκριμένο Gherkin Scenario X) κλάσεις που προδιαγράψατε στην ενότητα 1) ούτως ώστε να παρέχουν τη λειτουργικότητα που περιγράψατε στο Gherkin Scenario X του Gherkin Feature Y.

Ακολούθως εισάγετε εικόνα του UML διαγράμματος ακολουθιών.

Παράρτημα I – Γλωσσάριο

Το σετ των ακρωνυμίων που χρησιμοποιείτε στο έγγραφο

ΛΑ-*xxxx* Λειτουργική Απαίτηση *xxxx*

ΟΑ Ομάδα Εργασίας

Παράρτημα IΙ – Ανοιχτά Θέματα

<Μια δυναμική λίστα με ανοιχτά θέματα απαιτήσεων, θα δημιουργηθεί στο παράρτημα Γ>