



# CurbSprings

Your Reliable Partner in Urban Mobility

## Προδιαγραφές Συστήματος

### Del.1.2

Version 1.3

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](mailto:afotiadis@ece.auth.gr)

09/05/2024



## Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	<a href="mailto:asymeon@issel.ee.auth.gr">asymeon@issel.ee.auth.gr</a>
Πίπτης Γεώργιος	25	<a href="mailto:gkpittis@ece.auth.gr">gkpittis@ece.auth.gr</a>
Γουρδομιχάλης Αναστάσιος	25	<a href="mailto:anasgour@ece.auth.gr">anasgour@ece.auth.gr</a>
Τσαρναδέλης Αθανάσιος Γρηγόριος	25	<a href="mailto:atsarnad@ece.auth.gr">atsarnad@ece.auth.gr</a>
Φωτιάδης Αλέξανδρος	25	<a href="mailto:afotiadis@ece.auth.gr">afotiadis@ece.auth.gr</a>



## Πίνακας Περιεχομένων

Πίνακας Περιεχομένων .....	3
Λίστα Σχημάτων .....	6
1 Στατική Μοντελοποίηση .....	8
1.1 Πακέτο κλάσεων Feature 1: Reserve a parking spot .....	8
1.1.1 Ορισμός Κλάσεων .....	8
1.1.1.1 Ορισμός Κλάσης User .....	8
1.1.1.2 Ορισμός Κλάσης Reservation .....	9
1.1.1.3 Ορισμός Κλάσης ReservationController .....	10
1.1.1.4 Ορισμός Κλάσης MakeReservationGUI .....	10
1.1.1.5 Ορισμός Κλάσης SpotInformationGUI .....	11
1.1.1.6 Ορισμός Κλάσης DBProxy .....	11
1.1.2 Διάγραμμα κλάσεων .....	12
1.2 Πακέτο κλάσεων Feature 2: Manage Reservation .....	12
1.2.1 Ορισμός Κλάσεων .....	12
1.2.1.1 Ορισμός Κλάσης ManageReservationGUI .....	12
1.2.1.2 Ορισμός Κλάσης UpdateReservationGUI .....	13
1.2.1.3 Ορισμός Κλάσης DeleteReservationGUI .....	13
1.2.2 Διάγραμμα κλάσεων .....	14
1.3 Πακέτο κλάσεων Feature 3: Manage parking spots .....	15
1.3.1 Ορισμός Κλάσεων .....	15
1.3.1.1 Ορισμός Κλάσης ParkingSpotController .....	15
1.3.1.2 Ορισμός Κλάσης ModifySpotInformationGUI .....	15
1.3.1.3 Ορισμός Κλάσης RemoveSpotGUI .....	16
1.3.1.4 Ορισμός Κλάσης ParkingSpot .....	17
1.3.2 Διάγραμμα κλάσεων .....	18
1.4 Πακέτο κλάσεων Feature 4: View available parking spots .....	18
1.4.1 Ορισμός Κλάσεων .....	18
1.4.1.1 Ορισμός Κλάσης ShowAllAvailableSpotsGUI .....	18
1.4.1.2 Ορισμός Κλάσης HomePageGUI .....	19
1.4.1.3 Ορισμός Κλάσης StartPageGUI .....	20
1.4.1.4 Ορισμός Κλάσης DisplayController .....	21
1.4.2 Διάγραμμα κλάσεων .....	22
1.5 Πακέτο κλάσεων Feature 5: Search for available parking spots .....	22
1.5.1 Ορισμός Κλάσεων .....	22
1.5.1.1 Ορισμός Κλάσης SearchSpotByChargerAvailability .....	22
1.5.1.2 Ορισμός Κλάσης SearchSpotByType .....	23
1.5.1.3 Ορισμός Κλάσης SearchSpotByAddress .....	24



1.5.2 Διάγραμμα κλάσεων.....	25
1.6 Πακέτο κλάσεων Feature 6: Add spot owner.....	25
1.6.1 Ορισμός Κλάσεων.....	25
1.6.1.1 Ορισμός Κλάσης SpotOwner.....	25
1.6.1.2 Ορισμός Κλάσης AddSpotOwnerGUI.....	26
1.6.1.3 Ορισμός Κλάσης AdminPageGUI.....	27
1.6.1.4 Ορισμός Κλάσης SpotOwnerController.....	27
1.6.2 Διάγραμμα κλάσεων.....	28
1.7 Πακέτο κλάσεων Feature 7: Add parking spots.....	28
1.7.1 Ορισμός Κλάσεων.....	28
1.7.1.1 Ορισμός Κλάσης SpotOwnerPageGUI.....	28
1.7.1.2 Ορισμός Κλάσης AddSpotInformationGUI.....	29
1.7.2 Διάγραμμα κλάσεων.....	30
1.8 Πακέτο κλάσεων Feature 8: Make Payment.....	30
1.8.1 Ορισμός Κλάσεων.....	30
1.8.1.1 Ορισμός Κλάσης Payment.....	30
1.8.1.2 Ορισμός Κλάσης PaymentScreenGUI.....	31
1.8.1.3 Ορισμός Κλάσης PaymentProxy.....	31
1.8.2 Διάγραμμα κλάσεων.....	31
1.9 Πακέτο κλάσεων Feature 9: Register vehicle license plate.....	32
1.9.1 Ορισμός Κλάσεων.....	32
1.9.1.1 Ορισμός Κλάσης LicensePlate.....	32
1.9.1.2 Ορισμός Κλάσης ManageLicensePlateGUI.....	32
1.9.1.3 Ορισμός Κλάσης RegisterVehicleLicensePlateGUI.....	32
1.9.1.4 Ορισμός Κλάσης LicencePlateController.....	33
1.9.2 Διάγραμμα κλάσεων.....	34
1.10 Πακέτο κλάσεων Feature 10: Modify vehicle license plate.....	34
1.10.1 Ορισμός Κλάσεων.....	34
1.10.1.1 Ορισμός Κλάσης ModifyVehicleLicensePlateGUI.....	34
1.10.2 Διάγραμμα κλάσεων.....	35
2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν.....	36
2.1 Πρότυπο Proxy.....	36
2.2 Πρότυπο Facade.....	37
2.3 Πρότυπο Bridge.....	38
3 Δυναμική Μοντελοποίηση.....	39
3.1 Gherkin Scenario View Available Parking Spots (Feature 4: View available parking spots).....	39
3.2 Gherkin Scenario Reserve Parking Spot (Feature 1: Reserve a parking spot).....	39
3.3 Gherkin Scenario Add Parking Spots (Feature 7: Add parking spots).....	40



---

Παράρτημα Ι – Γλωσσάριο .....	41
Παράρτημα ΙΙ – Ανοιχτά Θέματα.....	42



## Λίστα Σχημάτων

Figure 1: Κλάση User .....	8
Figure 2: Κλάση Reservation .....	9
Figure 3:Κλάση Reservation Controller.....	10
Figure 4: Κλάση MakeReservationGUI .....	10
Figure 5: Κλάση SpotInformationGUI.....	11
Figure 6: Διάγραμμα κλάσεων feature 1 .....	12
Figure 7: Κλάση MakeReservationGUI .....	13
Figure 8: Κλάση UpdateReservationGUI .....	13
Figure 9: Κλάση DeleteReservationGUI .....	14
Figure 10: Διάγραμμα κλάσεων feature 2 .....	14
Figure 11: Κλάση ParkingSpotController.....	15
Figure 12: Κλάση ModifySpotInformationGUI .....	15
Figure 13: Κλάση RemoveSpotGUI.....	16
Figure 14: Διάγραμμα κλάσεων feature 3 .....	18
Figure 15: Κλάση ShowAllAvailableSpotsGUI.....	18
Figure 16: Κλάση HomePageGUI.....	19
Figure 17: Κλάση StartPageGUI.....	20
Figure 18: Κλάση DisplayController .....	21
Figure 19: Διάγραμμα κλάσεων feature 4 .....	22
Figure 20: Κλάση SearchByChargerAvailablility .....	22
Figure 21: Κλάση SearchSpotbyType .....	23
Figure 22: Κλάση SearchSpotByAddress .....	24
Figure 23: Διάγραμμα κλάσεων feature 5 .....	25
Figure 24: Κλάση SpotOwner .....	25
Figure 25: Κλάση AddSpotOwnerGUI .....	26
Figure 26: Κλάση AdminPageGUI .....	27
Figure 27: Κλάση SpotOwnerController.....	27
Figure 28: Διάγραμμα κλάσεων feature 6 .....	28
Figure 29: Κλάση SpotOwnerPageGUI .....	28
Figure 30: Κλάση AddSpotInformationGUI .....	29
Figure 31: Διάγραμμα κλάσεων feature 7 .....	30
Figure 32: Κλάση Payment .....	30
Figure 33: PaymentScreenGUI .....	31
Figure 34: Κλάση PaymentProxy .....	31
Figure 35: Διάγραμμα κλάσεων feature 8 .....	31
Figure 36: Κλάση LicensePlate .....	32
Figure 37: Κλάση ManageLicensePlateGUI .....	32
Figure 38: Κλάση RegisterVehicleLicensePlateGUI .....	33
Figure 39: Κλάση LicensePlateController .....	33
Figure 40: Διάγραμμα Κλάσεων feature 9 .....	34
Figure 41: Κλάση ModifyVehicleLicensePlate.....	34
Figure 42: Διάγραμμα κλάσεων feature 10 .....	35



---

Figure 43: Εφαρμογή προτύπου Proxy .....	36
Figure 44: Εφαρμογή προτύπου Facade.....	37
Figure 45: Εφαρμογή προτύπου Bridge.....	38
Figure 46: Δυναμική μοντελοποίηση για το Feature 4.....	39
Figure 47: Δυναμική μοντελοποίηση για το Feature 1.....	39
Figure 48: Δυναμική μοντελοποίηση για το Feature 7.....	40



# 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

«Entity» User
-UserID: Integer -UserName: String -UserLicensePlate: LicensePlate -UserReservation: list <Reservation>
+User(UserID: Integer, UserName: String, UserLicensePlate: LicensePlate, UserReservation: list <Reservation>): void +setUserID(UserID: Integer): void +setUserName(UserName: String): void +setUserLicensePlate(UserLicensePlate: LicensePlate): void +setUserReservation(UserReservation: list <Reservation>): void +getUserID(): Integer +getUserName(): String +getUserLicensePlate(): LicensePlate +getUserReservation(): list <Reservation>

Figure 1: Κλάση User

#### 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:Integer): 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 αντικείμενα τύπου 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 που αποτελεί την πινακίδα του χρήστη.
- getUserReservation(): list <Reservation> - Επιστρέφει μια λίστα με αντικείμενα τύπου Reservation που αποτελούν τις κρατήσεις του χρήστη.





### 1.1.1.2 Ορισμός Κλάσης Reservation

«Entity» Reservation
-ReservationID: Integer -ReservationDate: String -ReservationStartTime: String -ReservationExpectedDuration: Float -ReservationSpot: ParkingSpot -ReservationUser: User  +Reservation(ReservationID: Integer, ReservationStartTime: String, ReservationExpectedDuration: Float, ReservationSpot: ParkingSpot, ReservationUser: User, ReservationDate: String): void +setReservationStartTime(ReservationStartTime: String): void +setReservationExpectedDuration(ReservationExpectedDuration: Float): void +setReservationSpot(ReservationSpot: ParkingSpot): void +setReservationUser(ReservationUser: User): void +setReservationDate(ReservationDate: String): void +setReservationID(ReservationID: Integer): void +getReservationID(): Integer +getReservationStartTime(): String +getReservationExpectedDuration(): Float +getReservationSpot(): ParkingSpot +getReservationUser(): User +getReservationDate(): String

Figure 2: Κλάση Reservation

#### Χαρακτηριστικά κλάσης

- ReservationID: Integer - Το μοναδικό αναγνωριστικό ID της κράτησης.
- ReservationDate: String - Η ημερομηνία της κράτησης.
- ReservationStartTime: String - Η ώρα εκκίνησης της κράτησης.
- ReservationExpectedDuration: 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, ReservationExpectedDuration: 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(ReservationExpectedDuration: 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 Reservation τον χρήστη που έκανε την κράτηση.
- setReservationDate(ReservationDate: 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.
- getReservationExpectedDuration(): 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`

«Controller» <code>ReservationController</code>
<code>+ReservationController(): void</code> <code>+reserveParkingSpot(ReservationDate: String, ReservationStartTime: String, ReservationExpectedDuration: float): Reservation</code> <code>+updateReservation(ReservationDate: String, ReservationStartTime: String, ReservationExpectedDuration: float, UserReservation: Reservation): void</code> <code>+deleteReservation(UserReservation: Reservation): void</code> <code>+makePayment(UserReservation: Reservation): Payment</code> <code>+calculateCost(ReservationExpectedDuration: Float): Float</code>

Figure 3:Κλάση `Reservation Controller`

### Μέθοδοι κλάσης

- `ReservationController(): void` - Η συνάρτηση δόμησης της κλάσης.
- `reserveParkingSpot(ReservationDate: String, ReservationStartTime: String, ReservationExpectedDuration: 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(ReservationDate: String, ReservationStartTime: String, ReservationExpectedDuration: float, UserReservation: Reservation): void` - Παίρνει σαν όρισμα την ημερομηνία, την ώρα, την αναμενόμενη διάρκεια καθώς και το αντικείμενο κράτησης που θέλω να τροποποιήσω και στη συνέχεια καλεί τις κατάλληλες μεθόδους της κλάσης `Reservation`, ώστε να ανανεωθούν οι πληροφορίες της κράτησης.
- `deleteReservation(UserReservation: Reservation): void` - Διαγράφει τη κράτηση που παίρνει ως όρισμα εισόδου.
- `makePayment(UserReservation: Reservation): Payment` - Καλεί τον constructor του `payment`, δημιουργώντας ένα νέο αντικείμενο πληρωμής.
- `calculateCost(ReservationExpectedDuration: Float): Float` - Υπολογίζει το κόστος της κράτησης με βάση την αναμενόμενη διάρκεια της κράτησης.

### 1.1.1.4 Ορισμός Κλάσης `MakeReservationGUI`

#### Χαρακτηριστικά κλάσης

- `labelMakeReservation: label` - Επιγραφή του κειμένου "Make Reservation".
- `labelDate: label` - Επιγραφή του κειμένου "Date:".
- `labelStartTime: label` - Επιγραφή του κειμένου "Start time:".
- `labelExpectedDuration: label` - Επιγραφή του κειμένου "Expected duration:".
- `ReservationDate: String` - Η ημερομηνία που εισάγει ο χρήστης.
- `StartTime: String` - Η ώρα που εισάγει ο χρήστης.
- `ExpectedDuration: String` - Η διάρκεια κράτησης που εισάγει ο χρήστης.
- `buttonMakeReservation: button` - Το κουμπί "Make Reservation" για την δημιουργία νέας κράτησης.

«Boundary» <code>MakeReservationGUI</code>
<code>-labelMakeReservation: label</code> <code>-labelDate: label</code> <code>-labelStartTime: label</code> <code>-labelExpectedDuration: label</code> <code>-ReservationDate: String</code> <code>-StartTime: String</code> <code>-ExpectedDuration: String</code> <code>-buttonMakeReservation: button</code> <code>-buttonBack: button</code>
<code>+MakeReservationGUI(): void</code> <code>+buttonMakeReservationPress(): void</code> <code>+buttonBackPress(): void</code>

Figure 4: Κλάση `MakeReservationGUI`



- buttonBack: button - Το κουμπί επιστροφής στην προηγούμενη σελίδα.(βελάκι πάνω αριστερά)

#### Μέθοδοι κλάσης

- MakeReservationGUI(): void - Η συνάρτηση δόμησης του γραφικού.
- buttonMakeReservationPress(): void - Όταν πατηθεί το κουμπί “Make Reservation” καλεί την 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

##### Χαρακτηριστικά κλάσης

- labelSpotInfo: label - Επιγραφή του κειμένου “Spot Information”.
- labelAddress: label - Επιγραφή του κειμένου “Address:”.
- labelSpotType: label - Επιγραφή του κειμένου “Spot type:”.
- SpotImage: 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 “Reserve spot” για κράτηση της θέσης.

«Boundary» SpotInformationGUI
-labelSpotInfo: label -labelAddress: label -labelSpotType: label -SpotImage: image -SpotAddress: String -SpotType: String -SpotChargerAvailability: boolean -buttonBack: button -buttonReserveSpot: button
+SpotInformationGUI(): void +buttonReserveSpotPress(): void +buttonBackPress(): void

Figure 5: Κλάση SpotInformation-GUI

##### Μέθοδοι κλάσης

- SpotInformationGUI(): void - Η συνάρτηση δόμησης της κλάσης.
- buttonReserveSpotPress(): void - Όταν πατηθεί το κουμπί “Reserve spot” καλεί την συνάρτηση displayMakeReservationPage() του DisplayController και 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

##### Μέθοδοι κλάσης

- getSpot(All: Boolean, Address: String, Type: String, Charger: Boolean): <list> ParkingSpot - Συνάρτηση που παίρνει ως 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.

«Proxy» DBProxy
+getSpot(All: Boolean, Address: String, Type: String, Charger: Boolean): <list> ParkingSpot +getLicensePlate(User: User): LicensePlate +getReservation(User: User): <list> Reservation +getSpotOwner(SpotOwnerID: Integer): SpotOwner +getUser(UserID: Integer): User +putSpot(ParkingSpot: ParkingSpot): void +putLicensePlate(LicensePlate: LicensePlate): void +putReservation(Reservation: Reservation): void +putSpotOwner(SpotOwner: SpotOwner): void +putUser(User: User): void

Figure 6:Κλάση DBProxy

- `getReservation(User: User): <list>` Reservation - Συνάρτηση που παίρνει σαν όρισμα τον χρήστη και επιστρέφει μια λίστα με τις 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`.
- `putSpot(ParkingSpot: ParkingSpot): void` - Παίρνει σαν όρισμα ένα αντικείμενο θέσης πάρκινγκ και το τοποθετεί στην βάση δεδομένων.
- `putLicensePlate(LicensePlate: LicensePlate): void` - Παίρνει σαν όρισμα ένα αντικείμενο `LicensePlate` και το τοποθετεί στην βάση δεδομένων.
- `putReservation(Reservation: Reservation): void` - Παίρνει σαν όρισμα ένα αντικείμενο κράτησης και το τοποθετεί στην βάση δεδομένων.
- `putSpotOwner(SpotOwner: SpotOwner): void` - Παίρνει σαν όρισμα ένα αντικείμενο `SpotOwner` και το τοποθετεί στην βάση δεδομένων.
- `putUser(User: User): void` - Παίρνει σαν όρισμα ένα αντικείμενο χρήστη και το τοποθετεί στην βάση δεδομένων.

### 1.1.2 Διάγραμμα κλάσεων

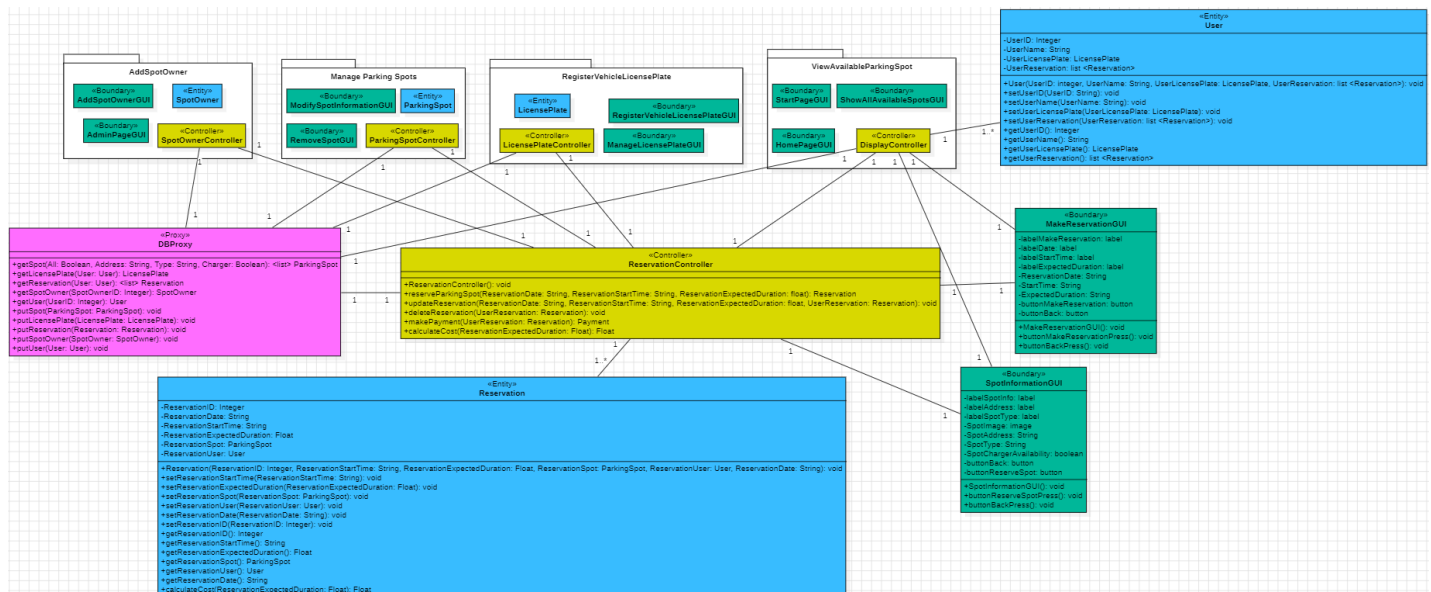


Figure 6: Διάγραμμα κλάσεων *feature 1*

## 1.2 Πακέτο κλάσεων Feature 2: Manage Reservation

### 1.2.1 Ορισμός Κλάσεων

#### 1.2.1.1 Ορισμός Κλάσης ManageReservationGUI

### Χαρακτηριστικά κλάσης

- `labelManageReservation`: `label` - Επιγραφή του κειμένου “Manage Reservation”
- `buttonUpdateReservation`: `button` - Κουμπί “Update reservation” για ενημέρωση μιας ήδη υπάρχουσας κράτησης.



- buttonDeleteReservation : button - Κουμπί "Delete Reservation" για διαγραφή μιας κράτησης.
- backButton: 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 "Update reservation" καλεί την συνάρτηση displayUpdateReservationPage() του DisplayController και ο χρήστης μεταφέρεται στη σελίδα για τροποποίηση της κράτησής του.
- buttonDeleteReservationPress(): void - Όταν πατηθεί το κουμπί "Delete Reservation" καλεί την συνάρτηση displayDeleteReservationPage() του DisplayController και ο χρήστης μεταφέρεται στη σελίδα για διαγραφή της κράτησής του.
- backButtonPress(): void - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

«Boundary» ManageReservationGUI
-labelManageReservation: label -buttonUpdateReservation: button -buttonDeleteReservation: button -buttonBack: button
+ManageReservationGUI(): void +buttonUpdateReservationPress(): void +buttonDeleteReservationPress(): void +buttonBackPress(): void

Figure 7: Κλάση MakeReservationGUI

#### 1.2.1.2 Ορισμός Κλάσης UpdateReservationGUI

##### Χαρακτηριστικά κλάσης

- labelUpdateReservation: label - Επιγραφή με το κείμενο "Update Reservation"
- labelDate: label - Επιγραφή με το κείμενο "Date:"
- labelStartTime: label - Επιγραφή με το κείμενο "Start time:"
- labelExpectedDuration: label - Επιγραφή με το κείμενο "Expected duration:"
- UpdateReservationDate: String - Η ενημερωμένη ημερομηνία κράτησης
- UpdateStartTime: String - Η ενημερωμένη ώρα έναρξης
- UpdateExpectedDuration: String - Η ενημερωμένη διάρκεια της κράτησης
- buttonSaveChanges: button – Το κουμπί "SAVE CHANGES" για την αποθήκευση των αλλαγών στις λεπτομέρειες κράτησης.
- backButton: button - Το κουμπί επιστροφής στην προηγούμενη σελίδα.

«Boundary» UpdateReservationGUI
-labelUpdateReservation: label -labelDate: label -labelStartTime: label -labelExpectedDuration: label -UpdateReservationDate: String -UpdateStartTime: String -UpdateExpectedDuration: String -buttonSaveChanges: button -buttonBack: button
+UpdateReservationGUI(): void +buttonSaveChangesPress(): void +buttonBackPress(): void

Figure 8: Κλάση UpdateReservationGUI

#### Μέθοδοι κλάσης

- UpdateReservationGUI(): void - Η συνάρτηση δόμησης της οθόνης για ενημέρωση των στοιχείων κράτησης.
- buttonSaveChangesPress(): void - Όταν πατηθεί το κουμπί "SAVE CHANGES" η εν λόγω συνάρτηση αποθηκεύει τις αλλαγές που ο χρήστης πραγματοποίησε στην κράτησή του.
- backButtonPress(): void - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

#### 1.2.1.3 Ορισμός Κλάσης DeleteReservationGUI

##### Χαρακτηριστικά κλάσης

- labelDeleteReservation: label - Επιγραφή με το κείμενο "Delete Reservation"



- labelShowCurrentReservations: label - Επιγραφή με το κείμενο “Showing your current reservations”
- labelReservationDate: label – Η ημερομηνία της κράτησης.
- labelReservationStartTime: label - Η ώρα έναρξης της κράτησης.
- labelReservationExpectedDuration: label – Η αναμενόμενη διάρκεια της κράτησης.
- ReservedSpotImage: image - Εικόνα που δείχνει την θέση πάρκινγκ που ο χρήστης ενοικιάζει.
- buttonDeleteReservation: button - Κουμπί (“κάδος” δίπλα από τη κράτηση) για διαγραφή κράτησης.
- backButton: button - Το κουμπί επιστροφής στην προηγούμενη σελίδα.
- ReservedParkingSpotsList: <list> Parking Spot - Λίστα με όλες τις κρατήσεις του χρήστη.



Figure 9: Κλάση DeleteReservationGUI

### Μέθοδοι κλάσης

- DeleteReservationGUI(): void - Η συνάρτηση δόμησης της σελίδας για διαγραφή κρατήσεων.
- buttonDeleteReservationPress(): void - Όταν πατηθεί το κουμπί διαγραφής, καλεί τη συνάρτηση deleteReservation() του ReservationController για να διαγράψει μια συγκεκριμένη κράτηση του χρήστη.
- backButtonPress(): void - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

### 1.2.2 Διάγραμμα κλάσεων

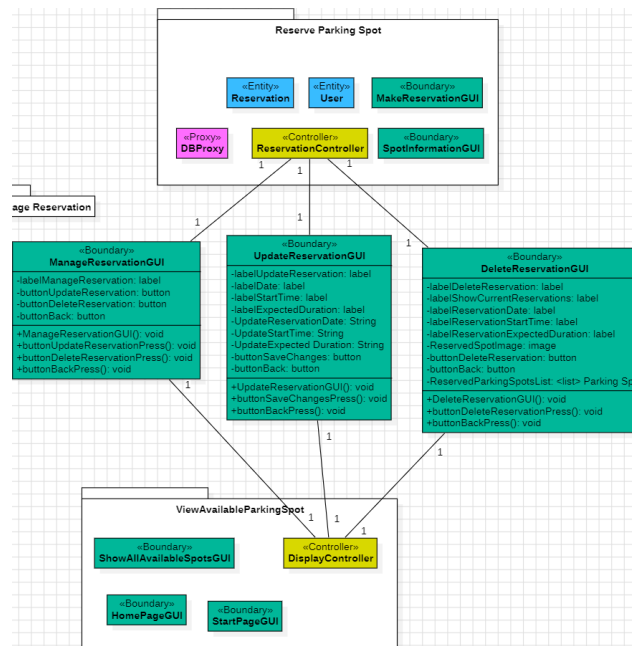


Figure 10: Διάγραμμα κλάσεων feature 2





## 1.3 Πακέτο κλάσεων Feature 3: Manage parking spots

### 1.3.1 Ορισμός Κλάσεων

#### 1.3.1.1 Ορισμός Κλάσης ParkingSpotController

«Controller» ParkingSpotController
+ParkingSpotController(): void +viewAllAvailableSpots(): list<ParkingSpot> +searchBySpotAddress(spotAddress: String): list<ParkingSpot> +searchBySpotType(spotType: String): list<ParkingSpot> +searchBySpotCharger(): list<ParkingSpot> +addParkingSpot(spotAddress: String, spotType: String, spotCharger: Boolean): ParkingSpot +modifyParkingSpot(spotAddress: String, spotType: String, spotCharger: Boolean, SpotOwnerParkingSpot: ParkingSpot): void +removeParkingSpot(SpotOwnerParkingSpot: ParkingSpot): void

Figure 11: Κλάση ParkingSpotController

#### 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 σε μια λίστα όλες τις θέσεις πάρκινγκ που διαθέτουν ηλεκτρικό φορτιστή.
- addParkingSpot(spotAddress:String, spotType:String, spotCharger:Boolean): ParkingSpot - Παίρνει σαν όρισμα την διεύθυνση, τον τύπο θέσης και την διαθεσιμότητα 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 διαθεσιμότητα ηλεκτρικού φορτιστή και το αντικείμενο θέσης που επιθυμώ να τροποποιήσω, 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.

#### 1.3.1.2 Ορισμός Κλάσης ModifySpotInformationGUI

##### Χαρακτηριστικά κλάσης

- labelModifyInfo: label - Επιγραφή του κειμένου “Modify spot information”.
- labelAddress: label - Επιγραφή του κειμένου “Address”.
- labelSpotType: label - Επιγραφή του κειμένου “Spot type”.
- labelCharger: label - Επιγραφή του κειμένου “Does the spot have a charger?”.
- textAddress: textbox - Πλαίσιο κειμένου στο οποίο ο ιδιοκτήτης εισάγει την διεύθυνση της προς ενοικίαση θέσης.

«Boundary» ModifySpotInformationGUI
-labelModifyInfo: label -labelAddress: label -labelSpotType: label -labelCharger: label -textAddress: textbox -textType: textbox -sliderCharger: slider -buttonSaveChanges: button -buttonBack: button  +ModifySpotInformationGUI(): void +buttonSaveChangesPress(): void +buttonBackPress(): void

Figure 12: Κλάση ModifySpotInformationGUI



- `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 “SAVE CHANGES” για αποθήκευση των αλλαγών στην καταχωρημένη θέση πάρκινγκ.
- `buttonBack`: `button` - Το κουμπί επιστροφής στην προηγούμενη σελίδα.

#### Μέθοδοι κλάσης

- `ModifySpotInformationGUI()`: `void` - Η συνάρτηση δόμησης της σελίδας `Modify Spot Information`.
- `buttonSaveChangesPress()`: `void` - Όταν πατηθεί το κουμπί “SAVE CHANGES”, γίνεται αποθήκευση των τροποποιημένων πληροφοριών της καταχωρημένης θέσης πάρκινγκ.
- `buttonBackPress()`: `void` - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

#### 1.3.1.3 Ορισμός Κλάσης `RemoveSpotGUI`

##### Χαρακτηριστικά κλάσης

- `labelRemoveSpot`: `label` - Επιγραφή των κειμένων “Remove spot” και “Showing your registered spots”.
- `labelSpotAddress`: `label` - Εκφράζει τη διεύθυνση της θέσης.
- `labelSpotType`: `label` - Εκφράζει τον τύπο της θέσης.
- `labelSpotCharger`: `label` - Εκφράζει τη διαθεσιμότητα φορτιστή της θέσης.
- `buttonDelete`: `button` - Το κουμπί (“κάδος” δίπλα στη θέση) 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 τον `SpotOwner`.

«Boundary» RemoveSpotGUI
-labelRemoveSpot: label -labelSpotAddress: label -labelSpotType: label -labelSpotCharger: label -buttonDelete: button -imageSpot: image -buttonBack: button -parkingSpotList: <list> ParkingSpot
+RemoveSpotGUI(): void +buttonBackPress(): void +buttonDelete(ParkingSpot: ParkingSpot): void

Figure 13: Κλάση `RemoveSpotGUI`

#### 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 καλεί την `removeParkingSpot()` του `ParkingSpotController` για να διαγράψει αυτή την θέση πάρκινγκ από το σύστημα. Στη συγκεκριμένη περίπτωση το όρισμα εισόδου της συνάρτησης `removeParkingSpot()` ταυτίζεται με εκείνο της `buttonDelete(ParkingSpot: ParkingSpot): void`.



**1.3.1.4 Ορισμός Κλάσης ParkingSpot**

«Entity» ParkingSpot
-SpotID: int -SpotOwner: SpotOwner -SpotAddress: String -SpotType: String -ChargerAvailability: Boolean
+ParkingSpot(SpotID: int, SpotOwner: SpotOwner, SpotAddress: String, SpotType: String, ChargerAvailability: Boolean): void +setSpotID(SpotID: int): void +setSpotOwner(SpotOwner: SpotOwner): void +setSpotAddress(SpotAddress: String): void +setSpotType(SpotType: String): void +setChargerAvailability(ChargerAvailability: Boolean): void +getSpotID(): int +getSpotOwner(): SpotOwner +getSpotAddress(): String +getSpotType(): String +getChargerAvailability(): Boolean

**Χαρακτηριστικά Κλάσης**

- SpotID: int – Είναι ο αριθμός ID της θέσης πάρκινγκ μέσα στο σύστημα.
- SpotOwner: SpotOwner – Είναι ο ιδιοκτήτης της θέσης πάρκινγκ.
- SpotAddress: String – Η διεύθυνση της θέσης πάρκινγκ.
- SpotType: String – Ο τύπος της θέσης πάρκινγκ.
- ChargerAvailability: Boolean – Η διαθεσιμότητα ηλεκτρικού φορτιστή της θέσης πάρκινγκ.

**Μέθοδοι κλάσης**

- ParkingSpot(SpotID:int,SpotOwner:SpotOwner,SpotAddress:String,SpotType:String,ChargerAvailability:Boolean): void- Αποτελεί την συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τον αριθμό ID της θέσης, τον ιδιοκτήτη της θέσης, την διεύθυνση της θέσης, τον τύπο της θέσης καθώς και τη διαθεσιμότητα φορτιστή της θέσης.
- setSpotID(SpotID:int): void – Αλλάζει τον αριθμό ID της θέσης σε SpotID.
- setSpotOwner(SpotOwner:SpotOwner): void – Αλλάζει τον ιδιοκτήτη της θέσης σε SpotOwner.
- setSpotAddress(SpotAddress:String): void – Αλλάζει την διεύθυνση της θέσης σε SpotAddress.
- setSpotType(SpotType:String): void – Αλλάζει τον τύπο της θέσης σε SpotType.
- setChargerAvailability(ChargerAvailability:Boolean): void – Αλλάζει τη διαθεσιμότητα φορτιστή της θέσης σε ChargerAvailability.
- getSpotID(): int – Επιστρέφει τον αριθμό ID της θέσης.
- getSpotOwner(): SpotOwner – Επιστρέφει τον ιδιοκτήτη της θέσης.
- getSpotAddress(): String – Επιστρέφει τη διεύθυνση της θέσης.
- getSpotType(): String – Επιστρέφει τον τύπο της θέσης.
- getChargerAvailability(): Boolean – Επιστρέφει τη διαθεσιμότητα φορτιστή της θέσης.

### 1.3.2 Διάγραμμα κλάσεων

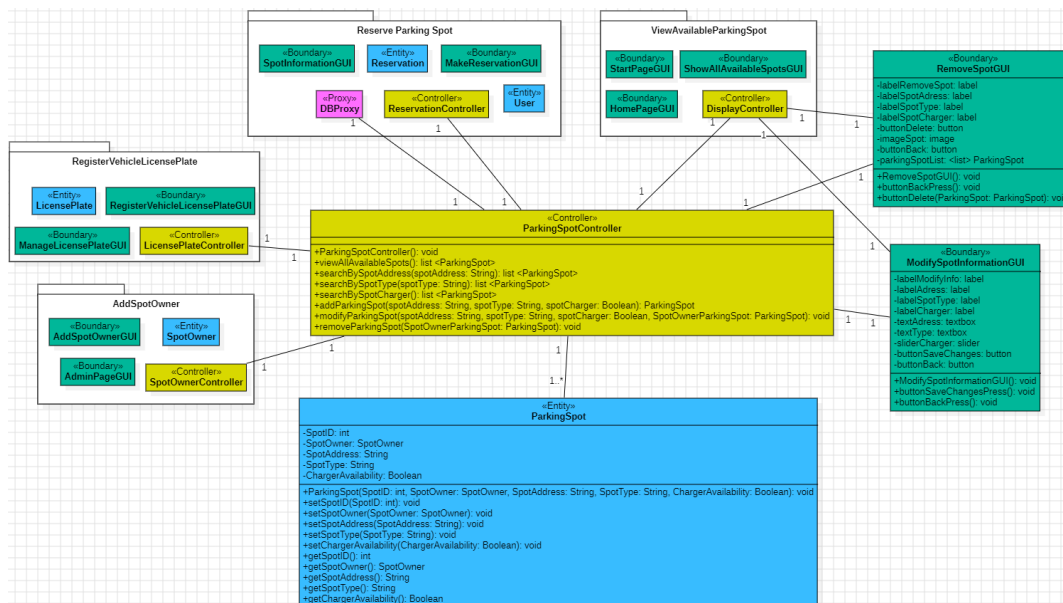


Figure 14: Διάγραμμα κλασεων feature 3

#### 1.4 Πακέτο κλάσεων Feature 4: View available parking spots

### 1.4.1 Ορισμός Κλάσεων

#### 1.4.1.1 Ορισμός Κλάσης ShowAllAvailableSpotsGUI

### Χαρακτηριστικά Κλάσης

- `labelShowAllAvailableSpots: label` - Επιγραφή του κειμένου “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 της διαθέσιμης θέσης πάρκινγκ.
  - `AvailableSpotsImage:image` – Οι εικόνες των διαθέσιμων θέσεων πάρκινγκ.
  - `buttonSelectAvailableSpot:button` – Πατάω αυτό το κουμπί (βελάκι δεξιά από κάθε θέση) για να διαλέξω θέση.
  - `buttonBack:button` - Το κουμπί για επιστροφή στην προηγούμενη σελίδα.
  - `AvailableParkingSpotsList:<list> ParkingSpot` - Είναι η λίστα όλων των διαθέσιμων θέσεων πάρκινγκ που υπάρχουν στο σύστημα. Σε αυτή περιέχονται όλες οι θέσεις που εμφανίζονται στη διεπαφή.
- ```

classDiagram
    class ShowAllAvailableSpotsGUI {
        <<boundary>>
        -labelShowAllAvailableSpots: label
        -labelAddress: label
        -labelSpotType: label
        -labelCharger: label
        -AvailableSpotsImage: image
        -buttonSelectAvailableSpot: button
        -buttonBack: button
        -AvailableParkingSpotsList: <list> Parking Spot
        +ShowAllAvailableSpotsGUI(): void
        +buttonSelectAvailableSpotPress(): void
        +buttonBackPress(): void
    }
  
```
- Figure 15: Κλάση *ShowAllAvailableSpotsGUI*

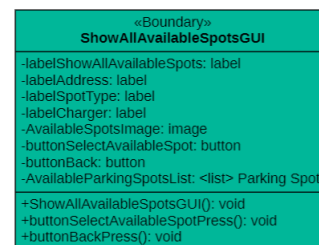


Figure 15: Κλάση ShowAllAvailableSpotsGUI

## Μέθοδοι κλάσης

- ShowAllAvailableSpotsGUI(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.



- `buttonSelectAvailableSpotPress():void` - Όταν πατηθεί το κουμπί επιλογής θέσης(βελάκι δεξιά της θέσης), καλεί τη συνάρτηση `displaySpotInformationGUI()` του `DisplayController` για να προβάλλει τις πληροφορίες της επιλεγμένης θέσης.
- `buttonBackPress():void` - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

#### 1.4.1.2 Ορισμός Κλάσης `HomePageGUI`

##### Χαρακτηριστικά Κλάσης

- `labelWelcome: label` - Επιγραφή του κειμένου "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 δει τις 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 δει τις 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.
- `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.

| «Boundary»<br>HomePageGUI                                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -labelWelcome: label<br>-labelUsername: label<br>-textSearch: textbox<br>-buttonAllSpots: button<br>-buttonSearchByType: button<br>-buttonSpotsWithCharger: button<br>-buttonMakePayment: button<br>-buttonManageReservation: button<br>-buttonManageLicensePlate: button<br>-buttonHelp: button                             |
| +HomePageGUI(User: User): void<br>+buttonSearchPress(): String<br>+buttonAllSpotsPress(): void<br>+buttonSearchByTypePress(): void<br>+buttonSpotsWithChargerPress(): void<br>+buttonMakePaymentPress(): void<br>+buttonManageReservationPress(): void<br>+buttonManageLicensePlatePress(): void<br>+buttonHelpPress(): void |

Figure 16: Κλάση `HomePageGUI`

##### Μέθοδοι κλάσης

- `HomePageGUI(User:User): void` - Αποτελεί την συνάρτηση δόμησης της κλάσης.
- `buttonSearchPress(): String` – Όταν πατηθεί ο μεγεθυντικός φακός του search bar, καλεί τη συνάρτηση `displaySearchByAddressPage()` του `DisplayController` για να προβάλλει τις διαθέσιμες θέσεις που βρίσκονται κοντά στη διεύθυνση που πληκτρολόγησε ο χρήστης. Η `buttonSearchPress()` επιστρέφει την διεύθυνση που πληκτρολόγησε ο χρήστης σε μορφή String.
- `buttonAllSpotsPress(): void` – Όταν πατηθεί το κουμπί "All available parking spots", καλεί τη συνάρτηση `displayAllAvailableSpotsPage()` του `DisplayController` για να προβάλλει όλες τις διαθέσιμες θέσεις πάρκινγκ που υπάρχουν στο σύστημα.



- `buttonSearchByTypePress():void` – Όταν πατηθεί το κουμπί “Search by type” καλεί τη συνάρτηση `displaySearchByTypePage()` του `DisplayController`, η οποία προβάλλει την οθόνη αναζήτησης διαθέσιμης θέσης με βάση τον τύπο.
- `buttonSpotsWithChargerPress(): void` – Όταν πατηθεί το κουμπί “Show only spots with charger”, καλεί τη συνάρτηση `displaySearchByChargerPage()` του `DisplayController` για να προβάλλει τις διαθέσιμες θέσεις που είναι εξοπλισμένες με ηλεκτρικό φορτιστή.
- `buttonMakePaymentPress(): void` – Όταν πατηθεί το κουμπί “Make payment” καλεί τη συνάρτηση `displayPaymentScreen()` του `DisplayController`, η οποία προβάλλει την οθόνη πληρωμής της κράτησης.
- `buttonManageReservationPress(): void` – Όταν πατηθεί το κουμπί “Manage Reservation” καλεί τη συνάρτηση `displayManageReservationPage()` του `DisplayController`, η οποία προβάλλει την οθόνη διαχείρισης κράτησης.
- `buttonManageLicensePlatePress(): void` – Όταν πατηθεί το κουμπί “Manage licence plate” καλεί τη συνάρτηση `displayManageLicensePlatePage()` του `DisplayController`, η οποία προβάλλει την οθόνη διαχείρισης πινακίδας οχήματος.
- `buttonHelpPress(): void` – Όταν πατηθεί το κουμπί βοήθειας τότε προβάλλει χρήσιμες πληροφορίες για τις βασικές λειτουργίες της εφαρμογής.

#### 1.4.1.3 Ορισμός Κλάσης `StartPageGUI`

##### Χαρακτηριστικά Κλάσης

- `imageLogo: image` – Είναι το λογότυπο της εφαρμογής.
- `labelName: label` – Επιγραφή του ονόματος της εφαρμογής (“CurbSprings”).
- `labelMotto: label` - Επιγραφή του motto της εφαρμογής (“Your reliable partner in urban mobility”).
- `buttonParkNow: button` – Είναι το κουμπί “Park now!”. 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` – Όταν πατηθεί το κουμπί “Park now!” καλεί τη συνάρτηση `displayHomePage()` του `DisplayController`, η οποία προβάλλει τη κεντρική οθόνη του χρήστη.
- `IamAdminPress(): void` – Όταν πατηθεί το κουμπί “I am an admin” καλεί τη συνάρτηση `displayAdminPage()` του `DisplayController`, η οποία προβάλλει τη κεντρική οθόνη του διαχειριστή του συστήματος.
- `IamSpotOwnerPress(): void` – Όταν πατηθεί το κουμπί “I am a spot owner” καλεί τη συνάρτηση `displaySpotOwnerPage()` του `DisplayController`, η οποία προβάλλει τη κεντρική οθόνη του ιδιοκτήτη θέσης.

| «Boundary»<br>StartPageGUI                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------|
| -imageLogo: image<br>-labelName: label<br>-labelMotto: label<br>-buttonParkNow: button<br>-buttonAdmin: button<br>-buttonSpotOwner: button |
| +StartPageGUI(): void<br>+ParkNowPress(): void<br>+IamAdminPress(): void<br>+IamSpotOwnerPress(): void                                     |

Figure 17: Κλάση `StartPageGUI`



#### 1.4.1.4 Ορισμός Κλάσης DisplayController

##### Μέθοδοι κλάσης

- DisplayController(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
- displayStartPage(): void – Προβάλλει την αρχική οθόνη της εφαρμογής.
- displayHomePage(): void – Προβάλλει 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.

| «Controller»<br>DisplayController                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| +DisplayController(): void<br>+displayStartPage(): void<br>+displayHomePage(): void<br>+displaySpotOwnerPage(): void<br>+displayAddSpotInformationPage(): void<br>+displayModifySpotInformationPage(): void<br>+displayRemoveSpotPage(): void<br>+displayMakeReservationPage(): void<br>+displayManageReservationPage(): void<br>+displayUpdateReservationPage(): void<br>+displayDeleteReservationPage(): void<br>+displayPaymentScreen(): void<br>+displayAllAvailableSpotsPage(): void<br>+displaySearchByAddressPage(): void<br>+displaySearchByTypePage(): void<br>+displaySearchByChargerPage(): void<br>+displayRegisterVehicleLicensePlatePage(): void<br>+displayModifyVehicleLicensePlatePage(): void<br>+displayManageLicensePlatePage(): void<br>+displayAdminPage(): void<br>+displayAddSpotOwnerPage(): void<br>+displaySpotInformationGUI(): void |

Figure 18: Κλάση DisplayController





## 1.4.2 Διάγραμμα κλάσεων

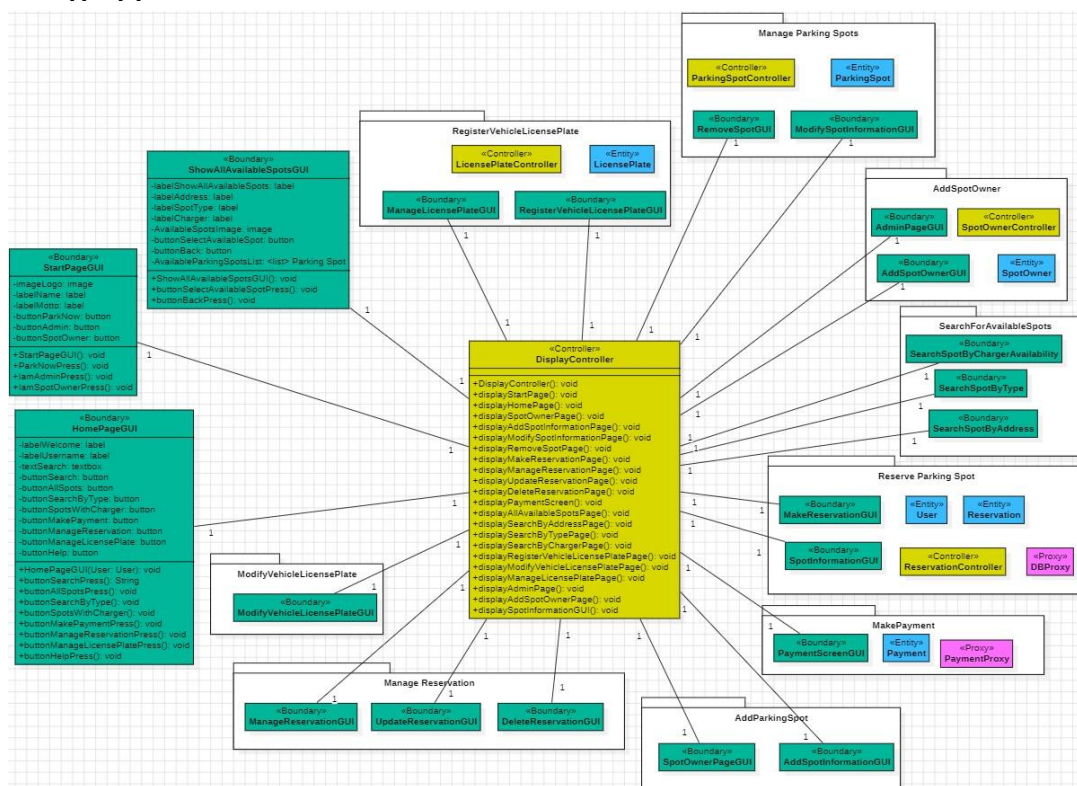


Figure 19: Διάγραμμα κλάσεων feature 4

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

### 1.5.1 Ορισμός Κλάσεων

#### 1.5.1.1 Ορισμός Κλάσης SearchSpotByChargerAvailability

##### Χαρακτηριστικά Κλάσης

- `labelSearchByCharger:label` - Επιγραφή του κειμένου "Search by Charger".
- `labelShowSpotsWithCharger:label` - Επιγραφή του κειμένου "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` - Το κουμπί για επιστροφή στην προηγούμενη σελίδα.

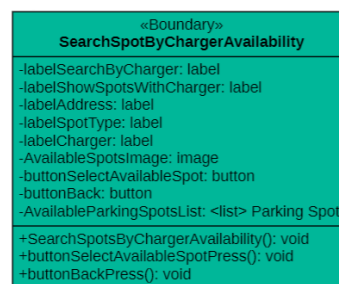


Figure 20: Κλάση SearchBy-ChargerAvailability



- AvailableParkingSpotsList:<list> ParkingSpot - 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 – Όταν πατηθεί το κουμπί επιλογής θέσης(βελάκι δεξιά της θέσης) καλεί τη συνάρτηση displaySpotInformationGUI() του DisplayController, η οποία προβάλλει τις πληροφορίες της επιλεγμένης θέσης.
- backButtonPress():void Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

#### **1.5.1.2 Ορισμός Κλάσης SearchSpotByType**

##### **Χαρακτηριστικά Κλάσης**

- labelSearchByType:label - Επιγραφή του κειμένου “Search by type”.
- labelSelectType:label - Επιγραφή του κειμένου “Select a specific type:”.
- labelFoundSpots:label - Επιγραφή του κειμένου “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.
- backButton:button - Το κουμπί για επιστροφή στην προηγούμενη σελίδα.
- AvailableParkingSpotsList:<list> ParkingSpot - Είναι η λίστα των διαθέσιμων θέσεων πάρκινγκ ενός συγκεκριμένου τύπου. Σε αυτή περιέχονται όλες οι θέσεις που εμφανίζονται στη διεπαφή.

| «Boundary»<br>SearchSpotByType                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -labelSearchByType: label<br>-labelSelectType: label<br>-labelFoundSpots: label<br>-labelAddress: label<br>-labelSpotType: label<br>-labelCharger: label<br>-AvailableSpotsImage: image<br>-buttonSelectSpotType: button<br>-buttonSelectAvailableSpot: button<br>-buttonBack: button<br>-AvailableParkingSpotsList: <list> Parking Spot |
| +SearchSpotsByType(): void<br>+buttonSelectAvailableSpotPress(): void<br>+buttonSelectSpotTypePress(): String<br>+buttonBackPress(): void                                                                                                                                                                                                |

Figure 21: Κλάση SearchSpot-byType

#### **Μέθοδοι κλάσης**

- SearchSpotsByType(): void - Αποτελεί την συνάρτηση δόμησης της κλάσης.
- buttonSelectSpotTypePress(): String – Όταν ο χρήστης επιλέξει έναν συγκεκριμένο τύπο θέσης(“Open”ή “Underground” ή “Garage”) , τότε καλεί την συνάρτηση searchBySpotType() του ParkingSpotController η οποία επιστρέφει σε μια λίστα τις διαθέσιμες θέσεις αυτού του συγκεκριμένου τύπου. Η buttonSelectSpotTypePress() επιστρέφει τον τύπο θέσης που επέλεξε ο χρήστης σε μορφή String. Αυτό το String αποτελεί όρισμα εισόδου της searchBySpotType() του ParkingSpotController.
- buttonSelectAvailableSpotPress():void – Όταν πατηθεί το κουμπί επιλογής θέσης(βελάκι δεξιά της θέσης) καλεί τη συνάρτηση displaySpotInformationGUI() του DisplayController, η οποία προβάλλει τις πληροφορίες της επιλεγμένης θέσης.



- `buttonBackPressed():void` - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

### 1.5.1.3 Ορισμός Κλάσης `SearchSpotByAddress`

#### Χαρακτηριστικά Κλάσης

- `labelSearchByAddress:label` - Επιγραφή του κειμένου "Search by address".
- `labelFoundSpots:label` - Επιγραφή του κειμένου "We found some 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` - Το κουμπί για επιστροφή στην προηγούμενη σελίδα.
- `AvailableParkingSpotsList:<list> ParkingSpot` - 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.

| «Boundary»<br>SearchSpotByAddress                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -labelSearchByAddress: label<br>-labelFoundSpots: label<br>-labelAddress: label<br>-labelSpotType: label<br>-labelCharger: label<br>-AvailableSpotsImage: image<br>-buttonSelectAvailableSpot: button<br>-buttonBack: button<br>-AvailableParkingSpotsList: <list> Parking Spot |
| +SearchSpotByAddress(): void<br>+buttonSelectAvailableSpotPress(): void<br>+buttonBackPressed(): void                                                                                                                                                                           |

Figure 22: Κλάση `SearchSpotByAddress`

#### Μέθοδοι κλάσης

- `SearchSpotByAddress(): void` - Αποτελεί την συνάρτηση δόμησης της κλάσης.
- `buttonSelectAvailableSpotPress():void` - Όταν πατηθεί το κουμπί επιλογής θέσης(βελάκι δεξιά της θέσης) καλεί τη συνάρτηση `displaySpotInformationGUI()` του `DisplayController`, η οποία προβάλλει τις πληροφορίες της επιλεγμένης θέσης.
- `buttonBackPressed():void` - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.





## 1.5.2 Διάγραμμα κλάσεων

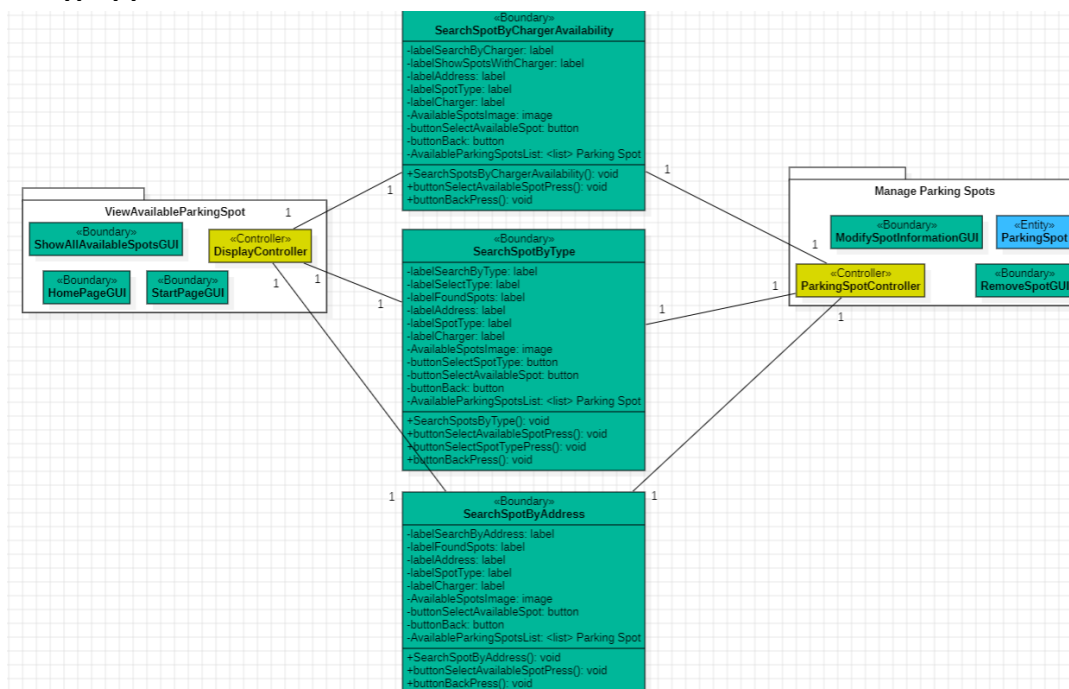


Figure 23: Διάγραμμα κλάσεων feature 5

## 1.6 Πακέτο κλάσεων Feature 6: Add spot owner

### 1.6.1 Ορισμός Κλάσεων

#### 1.6.1.1 Ορισμός Κλάσης SpotOwner

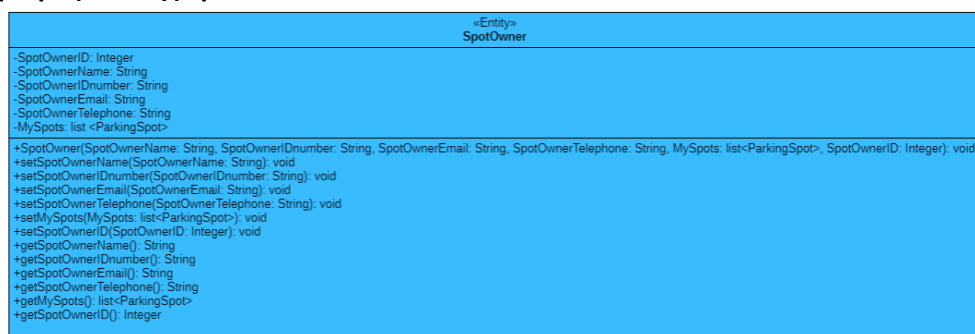


Figure 24: Κλάση SpotOwner

### Χαρακτηριστικά κλάσης

- SpotOwnerID: Integer - Αποτελεί τον αριθμό ταυτοποίησης του ιδιοκτήτη της θέσης parking.
- SpotOwnerName: String - Αποτελεί το όνομα του ιδιοκτήτη της θέσης parking.
- SpotOwnerIDnumber: String - Αποτελεί τον αριθμό ταυτότητας του ιδιοκτήτη της θέσης parking.
- SpotOwnerEmail: String - Αποτελεί το email του ιδιοκτήτη της θέσης parking.
- SpotOwnerTelephone: String - Αποτελεί το τηλέφωνο του ιδιοκτήτη της θέσης parking.
- MySpots: list <ParkingSpot> - Αποτελεί μια λίστα με όλες τις θέσεις parking που ανήκουν στον SpotOwner.

**Μέθοδοι κλάσης**

- SpotOwner(SpotOwnerName:String, SpotOwnerIDNumber:String, SpotOwnerEmail:String, SpotOwnerTelephone:String, MySpots:list<ParkingSpot>, SpotOwnerID:Integer): void - Αποτελεί τη συνάρτηση δόμησης της οντότητας SpotOwner.
- setSpotOwnerName(SpotOwnerName:String): void - Η μέθοδος αυτή θέτει το όνομα του ιδιοκτήτη της θέσης parking.
- setSpotOwnerIDNumber(SpotOwnerIDNumber:String): void - Η μέθοδος αυτή θέτει τον αριθμό ταυτότητας που έχει ο ιδιοκτήτης της θέσης parking.
- setSpotOwnerEmail(SpotOwnerEmail:String): void - Η μέθοδος αυτή θέτει το email που έχει ο ιδιοκτήτης της θέσης parking.
- setSpotOwnerTelephone(SpotOwnerTelephone:String): void - Η μέθοδος αυτή θέτει το τηλέφωνο που έχει ο ιδιοκτήτης της θέσης parking.
- setMySpots(MySpots:list<ParkingSpot>): void - Η μέθοδος αυτή θέτει τη λίστα με τις θέσεις parking που έχει ο ιδιοκτήτης.
- setSpotOwnerID(SpotOwnerID:Integer): void - Η μέθοδος αυτή θέτει τον αριθμό ταυτοποίησης που έχει ο ιδιοκτήτης της θέσης parking.
- getSpotOwnerName(): String - Η μέθοδος αυτή επιστρέφει το όνομα του ιδιοκτήτη.
- getSpotOwnerIDNumber(): String - Η μέθοδος αυτή επιστρέφει τον αριθμό ταυτότητας του ιδιοκτήτη.
- getSpotOwnerEmail(): String - Η μέθοδος αυτή επιστρέφει το email του ιδιοκτήτη.
- getSpotOwnerTelephone(): String - Η μέθοδος αυτή επιστρέφει το τηλέφωνο του ιδιοκτήτη.
- getMySpots(): list <ParkingSpot> - Η μέθοδος αυτή επιστρέφει τις θέσεις parking του ιδιοκτήτη.
- getSpotOwnerID(): Integer - Η μέθοδος αυτή επιστρέφει τον αριθμό ταυτοποίησης του ιδιοκτήτη.

**1.6.1.2 Ορισμός Κλάσης AddSpotOwnerGUI****Χαρακτηριστικά κλάσης**

- labelSpotOwnerSurname: label - Επιγραφή του κειμένου "Surname:"
- labelSpotOwnerName: label - Επιγραφή του κειμένου "Name:"
- labelSpotOwnerIDNumber: label - Επιγραφή του κειμένου "ID number:"
- labelSpotOwnerEmail: label - Επιγραφή του κειμένου "Email:"
- labelSpotOwnerPhoneNumber: label - Επιγραφή του κειμένου "Phone number:"
- labelAddSpotOwnerInfo: label - Επιγραφή του κειμένου "Add spot owner information"
- SpotOwnerName: String - Το όνομα του ιδιοκτήτη της θέσης πάρκινγκ.
- SpotOwnerSurname: String - Το επώνυμο του ιδιοκτήτη της θέσης πάρκινγκ.
- SpotOwnerIDNumber: String - Ο αριθμός ταυτότητας του ιδιοκτήτη της θέσης πάρκινγκ.
- SpotOwnerEmail: String - Το email του ιδιοκτήτη της θέσης πάρκινγκ.
- SpotOwnerPhoneNumber: String - Το τηλέφωνο του ιδιοκτήτη της θέσης πάρκινγκ.
- buttonSaveInformation: button - Κουμπί "SAVE" για την αποθήκευση των πληροφοριών.

| «Boundary»<br>AddSpotOwnerGUI                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -labelAddSpotOwnerInfo: label<br>-labelSpotOwnerName: label<br>-labelSpotOwnerSurname: label<br>-labelSpotOwnerIDNumber: label<br>-labelSpotOwnerEmail: label<br>-labelSpotOwnerPhoneNumber: label<br>-SpotOwnerName: String<br>-SpotOwnerSurname: String<br>-SpotOwnerIDNumber: String<br>-SpotOwnerEmail: String<br>-SpotOwnerPhoneNumber: String<br>-buttonSaveInformation: button<br>-buttonBack: button<br>+AddSpotOwnerGUI(): void<br>+buttonSaveInformationPress(): void<br>+buttonBackPress(): void |

Figure 25: Κλάση AddSpotOwnerGUI



- buttonBack: button – Κουμπί για επιστροφή στην προηγούμενη σελίδα.

#### **Μέθοδοι κλάσης**

- AddSpotOwnerGUI(): void - Αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- buttonSaveInformationPress(): void – Όταν πατηθεί το κουμπί “SAVE” τότε η συνάρτηση αποθηκεύει τις πληροφορίες του ιδιοκτήτη θέσεων.
- buttonBackPress(): void - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

#### **1.6.1.3 Ορισμός Κλάσης AdminPageGUI**

##### **Χαρακτηριστικά κλάσης**

- labelWelcome: label - Επιγραφή του κειμένου “Welcome”.
- labelAdminName: label - Επιγραφή του ονόματος του admin.
- labelSystemAdmin: label - Επιγραφή του κειμένου “You are a system admin”.
- buttonAddSpotOwner: button - Κουμπί “Add new spot owner” για την προσθήκη ιδιοκτήτη θέσης πάρκινγκ.

| «Boundary»<br>AdminPageGUI                                                                                |
|-----------------------------------------------------------------------------------------------------------|
| -labelWelcome: label<br>-labelAdminName: label<br>-labelSystemAdmin: label<br>-buttonAddSpotOwner: button |
| +AdminPageGUI(): void<br>+buttonAddSpotOwnerPress(): void                                                 |

Figure 26: Κλάση AdminPageGUI

##### **Μέθοδοι κλάσης**

- AdminPageGUI(): void - Αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- buttonAddSpotOwnerPress(): void - Όταν πατηθεί το κουμπί “Add new spot owner” καλεί την συνάρτηση displayAddSpotOwnerPage() του DisplayController και προβάλλει την σελίδα καταχώρησης των στοιχείων των ιδιοκτητών των θέσεων από τον διαχειριστή της εφαρμογής.

#### **1.6.1.4 Ορισμός Κλάσης SpotOwnerController**

| «Controller»<br>SpotOwnerController                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| +SpotOwnerController(): void<br>+saveSpotOwnerInfo(SpotOwnerName: String, SpotOwnerSurname: String, SpotOwnerIDNumber: String, SpotOwnerEmail: String, SpotOwnerPhoneNumber: String): SpotOwner |

Figure 27: Κλάση SpotOwnerController

##### **Μέθοδοι κλάσης**

- SpotOwnerController(): void - Αποτελεί τη συνάρτηση δόμησης του ελεγκτή
- saveSpotOwnerInfo(SpotOwnerName:String, SpotOwnerSurname:String, SpotOwnerIDNumber:String, SpotOwnerEmail:String, SpotOwnerPhoneNumber:String): SpotOwner - Παίρνει ως είσοδο τα στοιχεία του ιδιοκτήτη θέσεων πάρκινγκ που καταχώρησε ο διαχειριστής στην εφαρμογή(όνομα ,επίθετο, αριθμός ταυτότητας,email,αριθμός τηλεφώνου) και δημιουργεί ένα νέο αντικείμενο SpotOwner με αυτά τα στοιχεία.



## 1.6.2 Διάγραμμα κλάσεων

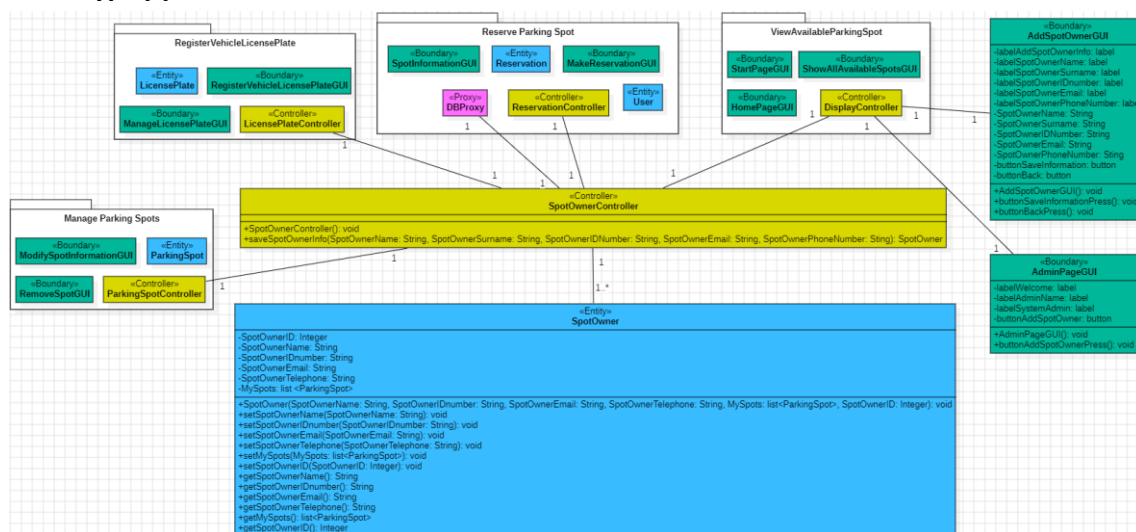


Figure 28: Διάγραμμα κλάσεων feature 6

## 1.7 Πακέτο κλάσεων Feature 7: Add parking spots

### 1.7.1 Ορισμός Κλάσεων

#### 1.7.1.1 Ορισμός Κλάσης SpotOwnerPageGUI

##### Χαρακτηριστικά κλάσης

- labelSpotOwnerName: label – Η επιγραφή του ονόματος του ιδιοκτήτη θέσης.
- labelWelcome: label - Επιγραφή με το κείμενο “Welcome”.
- labelSpotOwner: label - Επιγραφή με το κείμενο “You are a spot owner”.
- buttonAddSpot: button – Κουμπί “Add spot” για την προσθήκη θέσης πάρκινγκ στο σύστημα.
- buttonModifySpot: button - Κουμπί “Modify spot” για την επεξεργασία στοιχείων ήδη καταχωρημένης θέσης πάρκινγκ.
- buttonRemoveSpot: button - Κουμπί “Remove spot” για την αφαίρεση μιας θέσης πάρκινγκ από το σύστημα.

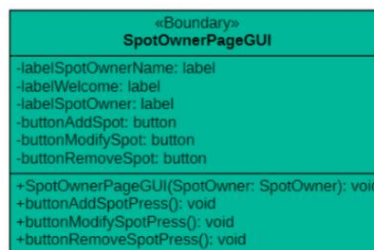


Figure 29: Κλάση SpotOwner-PageGUI

##### Μέθοδοι κλάσης

- SpotOwnerPageGUI( SpotOwner: SpotOwner): void - Η συνάρτηση δόμησης της σελίδας για την διαχείριση θέσεων. Παίρνει ως όρισμα ένα αντικείμενο τύπου SpotOwner και εμφανίζει το όνομα του SpotOwner στο επάνω μέρος της οθόνης. Επίσης, ενεργοποιεί τις δυνατότητες του SpotOwner.
- buttonAddSpotPress(): void - Όταν πατηθεί το κουμπί “Add spot”, τότε καλεί τη συνάρτηση displayAddSpotInformationPage() του DisplayController, η οποία με τη σειρά της μεταφέρει τον ιδιοκτήτη θέσης στην επόμενη σελίδα για να προσθέσει νέα θέση πάρκινγκ.



- `buttonModifySpotPress(): void` - Όταν πατηθεί το κουμπί “Modify spot”, καλεί τη συνάρτηση `displayModifySpotInformationPage()` του `DisplayController` για να μεταφέρει τον ιδιοκτήτη στη σελίδα επεξεργασίας των ήδη καταχωρημένων από αυτόν θέσεων.
- `buttonRemoveSpotPress(): void` - Όταν πατηθεί το κουμπί “Remove spot”, καλεί τη συνάρτηση `displayRemoveSpotPage()` του `DisplayController` για να μεταφέρει τον ιδιοκτήτη στη σελίδα με τις καταχωρημένες θέσεις του για να επιλέξει ποια θα διαγράψει.

### 1.7.1.2 Ορισμός Κλάσης `AddSpotInformationGUI`

#### Χαρακτηριστικά κλάσης

- `labelAddInfo: label` - Επιγραφή του κειμένου “Add spot Information”.
- `labelAdress: label` - Επιγραφή του κειμένου “Adress”.
- `labelSpotType: label` - Επιγραφή του κειμένου “Spot type”.
- `labelCharger: button` - Επιγραφή του κειμένου “Does the spot have a charger?”.
- `textAdress: textbox` - Πλαίσιο κειμένου στο οποίο εισάγεται η διεύθυνση του ιδιοκτήτη της θέσης parking.
- `textType: textbox` - Πλαίσιο κειμένου στο οποίο εισάγεται ο τύπος της θέσης parking.
- `sliderCharger: slider` - Toggle button με το οποίο επιβεβαιώνεται αν η θέση parking έχει φορτιστή.
- `buttonSave: button` - Το κουμπί “SAVE” για αποθήκευση των αλλαγών στην καταχωρημένη θέση πάρκινγκ.
- `buttonBack: button` - Το κουμπί για επιστροφή στην προηγούμενη σελίδα.

| «Boundary»<br>AddSpotInformationGUI                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -labelAddInfo: label<br>-labelAdress: label<br>-labelSpotType: label<br>-labelCharger: label<br>-textAdress: textbox<br>-textType: textbox<br>-sliderCharger: slider<br>-buttonSave: button<br>-buttonBack: button |
| +AddSpotInformationGUI(): void<br>+buttonSavePress(): void<br>+buttonBackPress(): void                                                                                                                             |

Figure 30: Κλάση `AddSpotInformationGUI`

#### Μέθοδοι κλάσης

- `AddSpotInformationGUI(): void` - Αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- `buttonSavePress(): void` - Όταν πατηθεί το κουμπί “SAVE”, γίνεται προσθήκη της νέας θέσης πάρκινγκ στο σύστημα.
- `buttonBackPress(): void` - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.



## 1.7.2 Διάγραμμα κλάσεων

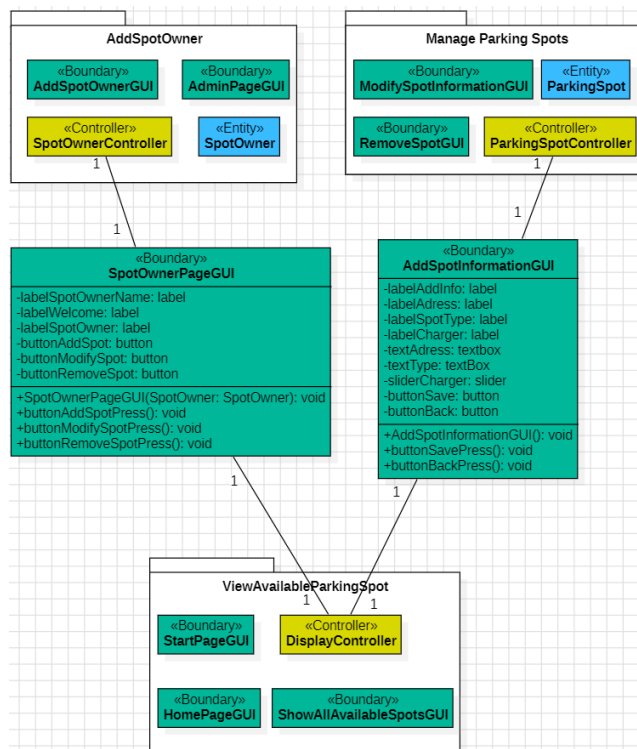


Figure 31: Διάγραμμα κλάσεων feature 7

## 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

- Payment(PaymentUser: User, PaymentAmount: Float) : void - 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 αντικείμενο User και ορίζει τον χρήστη που πραγματοποιεί την πληρωμή.
- setPaymentAmount(PaymentAmount:Float): void - Η συνάρτηση παίρνει σαν όρισμα κάποιο ποσό και το ορίζει ως ποσό πληρωμής.
- getPaymentUser(): User - Επιστρέφει τον χρήστη που πληρώνει.
- getPaymentAmount(): Float - Επιστρέφει το ποσό που πληρώνει ο χρήστης.

| «Entity»<br>Payment                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -PaymentUser: User<br>-PaymentAmount: Float                                                                                                                                                                   |
| +Payment(PaymentUser: User, PaymentAmount: Float): void<br>+setPaymentUser(PaymentUser: User): void<br>+setPaymentAmount(PaymentAmount: Float): void<br>+getPaymentUser(): User<br>+getPaymentAmount(): Float |

Figure 32: Κλάση Payment

#### 1.8.1.2 Ορισμός Κλάσης PaymentScreenGUI

### Χαρακτηριστικά κλάσης

- `labelMakePayment`: `label` - Επιγραφή του κειμένου “Make payment”.
- `labelSecureGatewayMessage`: `label` - Επιγραφή του κειμένου “Transfer to secure payment gateway”.
- `buttonPayNow`: `button` - Το κουμπί πληρωμής (“Pay Now”).
- `buttonBack`: `button` - Το κουμπί επιστροφής στην προηγούμενη σελίδα.

| «Boundary»<br>PaymentScreenGUI    |
|-----------------------------------|
| -labelMakePayment: label          |
| -labelSecureGatewayMessage: label |
| -buttonPayNow: button             |
| -buttonBack: button               |
| +PaymentScreenGUI(): void         |
| +buttonPayNowPress(): void        |
| +buttonBackPress(): void          |

Figure 33: PaymentScreen-GUI

## Μέθοδοι κλάσης

- `PaymentScreenGUI(): void` - Η συνάρτηση δόμησης του γραφικού. *GUI*
- `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 PaymentProxy

## 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.

|                                    |
|------------------------------------|
| «Proxy»<br><b>PaymentProxy</b>     |
| +makePayment(Cost: Float): Boolean |

Figure 34: Κλάση PaymentProxy

### 1.8.2 Διάγραμμα κλάσεων

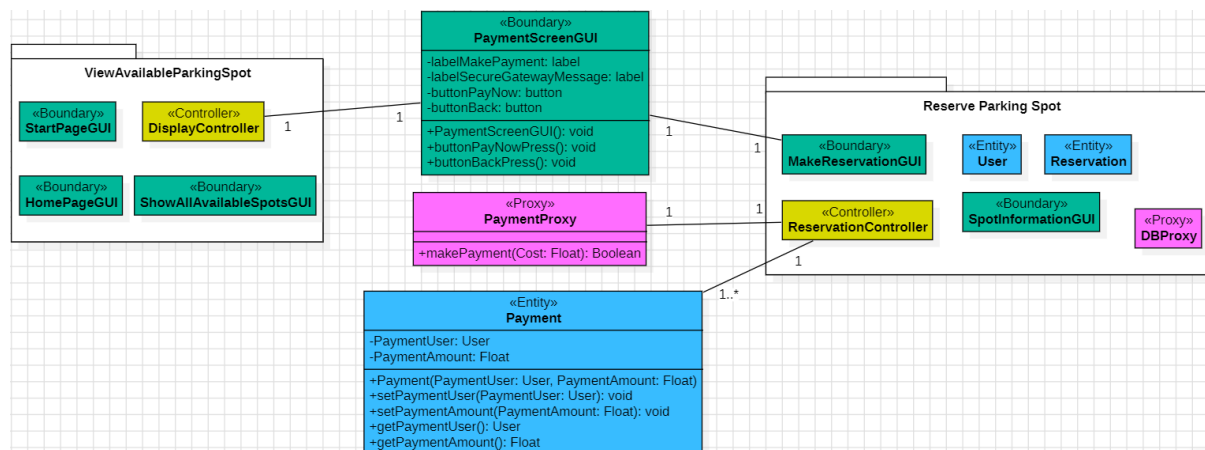


Figure 35: Διάγραμμα κλάσεων feature 8





## 1.9 Πακέτο κλάσεων Feature 9: Register vehicle license plate

### 1.9.1 Ορισμός Κλάσεων

#### 1.9.1.1 Ορισμός Κλάσης LicensePlate

##### Χαρακτηριστικά κλάσης

- PlateName:String - Αποτελεί τον αριθμό της πινακίδας.
- PlateOwner: User - Αποτελεί τον χρήστη ο οποίος είναι ιδιοκτήτης της πινακίδας.

| «Entity»<br>LicensePlate                                                                                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -PlateName: String<br>-PlateOwner: User                                                                                                                                                           |
| +LicencePlate(PlateName: String, PlateOwner: User): void<br>+setPlateName(PlateName: String): void<br>+setPlateOwner(PlateOwner: User): void<br>+getPlateName(): String<br>+getPlateOwner(): User |

Figure 36: Κλάση LicensePlate

##### Μέθοδοι κλάσης

- LicencePlate( PlateName:String, PlateOwner:User): void - Αποτελεί τη συνάρτηση δόμησης της οντότητας LicensePlate.
- setPlateName( PlateName:String): void - Η μέθοδος αυτή θέτει το όνομα της πινακίδας.
- setPlateOwner( PlateOwner:User): void - Η μέθοδος αυτή θέτει τον χρήστη που έχει την πινακίδα.
- getPlateName(): String - Η μέθοδος αυτή επιστρέφει το όνομα της πινακίδας.
- getPlateOwner(): User - Η μέθοδος αυτή επιστρέφει τον χρήστη που έχει την πινακίδα.

#### 1.9.1.2 Ορισμός Κλάσης ManageLicensePlateGUI

##### Χαρακτηριστικά κλάσης :

- labelManageLicencePlate:label - Επιγραφή του κειμένου “Manage licence plate”.
- buttonAddLicensePlate: button - Κουμπί “Add licence plate” για προσθήκη νέας πινακίδας.
- buttonModifyLicensePlate: button - Κουμπί “Modify licence plate” για τροποποίηση μιας ήδη καταχωρημένης πινακίδας.
- buttonBack: button - Κουμπί επιστροφής στην προηγούμενη οθόνη.

| «Boundary»<br>ManageLicensePlateGUI                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------|
| -labelManageLicencePlate: label<br>-buttonAddLicensePlate: button<br>-buttonModifyLicensePlate: button<br>-buttonBack: button               |
| +ManageLicensePlateGUI(): void<br>+buttonAddLicensePlatePress(): void<br>+buttonModifyLicensePlatePress(): void<br>+buttonBackPress(): void |

Figure 37: Κλάση ManageLicensePlateGUI

##### Μέθοδοι κλάσης:

- ManageLicensePlateGUI(): void - Αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- buttonAddLicensePlatePress(): void - Όταν πατηθεί το κουμπί “Add licence plate”, καλεί την συνάρτηση displayRegisterVehicleLicensePlatePage() του DisplayController για την προβολή της σελίδας καταχώρησης πινακίδας οχήματος.
- buttonModifyLicensePlatePress():void - Όταν πατηθεί το κουμπί “Modify licence plate”, καλεί την συνάρτηση displayModifyVehicleLicensePlatePage() του DisplayController για την προβολή της σελίδας τροποποίησης πινακίδας οχήματος.
- buttonBackPress(): void - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.

#### 1.9.1.3 Ορισμός Κλάσης RegisterVehicleLicensePlateGUI

##### Χαρακτηριστικά κλάσης :





- labelVehicleRegistration: label - Επιγραφή κειμένου “Vehicle Registration”.
- labelEnterRegistrationPlate: label - Επιγραφή κειμένου “Enter your vehicle’s registration plate”.
- RegistrationPlateName: String - Το όνομα της πινακίδας που καταχωρείται.
- buttonSaveRegistration: button - Κουμπί “SAVE” για την αποθήκευση της καταχώρησης.
- buttonBack: button - Κουμπί επιστροφής στην προηγούμενη οθόνη.

| «Boundary»<br>RegisterVehicleLicensePlateGUI                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -labelVehicleRegistration: label<br>-labelEnterRegistrationPlate: label<br>-RegistrationPlateName: String<br>-buttonSaveRegistration: button<br>-buttonBack: button |
| +RegisterVehicleLicensePlateGUI(): void<br>+buttonSaveRegistrationPress(): void<br>+buttonBackPress(): void                                                         |

Figure 38: Κλάση RegisterVehicleLicensePlateGUI

#### Μέθοδοι κλάσης:

- RegisterVehicleLicensePlateGUI(): void - Αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- buttonBackPress(): void - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.
- buttonSaveRegistrationPress(): void - Όταν πατηθεί το κουμπί “SAVE” , γίνεται καταχώρηση της πινακίδας μέσα στο σύστημα.

#### 1.9.1.4 Ορισμός Κλάσης LicencePlateController

| «Controller»<br>LicensePlateController                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| +LicensePlateController(): void<br>+registerVehicleLicensePlate(LicenseName: String): LicensePlate<br>+modifyVehicleLicensePlate(LicenseName: String, UserLicensePlate: LicensePlate): void |

Figure 39: Κλάση LicensePlateController

#### Μέθοδοι κλάσης

- LicensePlateController(): void - Αποτελεί τη συνάρτηση δόμησης του ελεγκτή
- registerVehicleLicensePlate( LicenseName:String): LicensePlate – Παίρνει ως είσοδο το όνομα της πινακίδας και δημιουργεί ένα νέο αντικείμενο LicensePlate με αυτό το όνομα πινακίδας. Αυτό το αντικείμενο LicensePlate το αντιστοιχίζει στον χρήστη (αντικείμενο κλάσης User) που καταχώρησε το όνομα της πινακίδας στο σύστημα.
- modifyVehicleLicensePlate(LicenseName:String, UserLicensePlate:LicensePlate): void - Παίρνει ως είσοδο το νέο όνομα της πινακίδας ή και ιδιοκτήτη του LicensePlate που θέλω να τροποποιήσω. Καλεί τη setPlateName() ή και την setPlateOwner() της κλάσης LicensePlate με όρισμα εισόδου τα νέα δεδομένα και έτσι τροποποιεί την ήδη καταχωρημένη πινακίδα του χρήστη.



## 1.9.2 Διάγραμμα κλάσεων

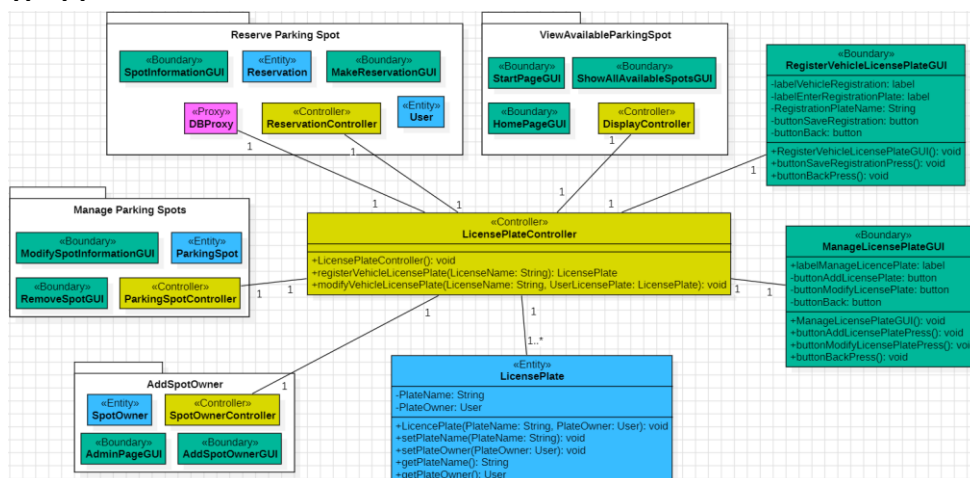


Figure 40: Διάγραμμα Κλάσεων feature 9

## 1.10 Πακέτο κλάσεων Feature 10: Modify vehicle license plate

### 1.10.1 Ορισμός Κλάσεων

#### 1.10.1.1 Ορισμός Κλάσης ModifyVehicleLicensePlateGUI

##### Χαρακτηριστικά κλάσης

- labelModifyLicencePlate: label - Επιγραφή κειμένου “Modify license plate”.
- labelModifyRegistrationPlate: label - Επιγραφή κειμένου “Enter your vehicle’s registration plate”.
- ModifiedPlateName: String - Το νέο όνομα της πινακίδας.
- buttonSaveChanges: button - Κουμπί “SAVE CHANGES” για την αποθήκευση των αλλαγών.
- buttonBack: button - Κουμπί επιστροφής στην προηγούμενη οθόνη.

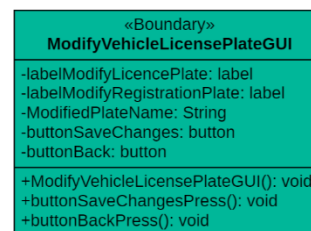


Figure 41: Κλάση ModifyVehicleLicensePlate

##### Μέθοδοι κλάσης

- ModifyVehicleLicensePlateGUI():void - Αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- buttonSaveChangesPress(): void - Όταν πατηθεί το κουμπί “SAVE CHANGES”, γίνεται αποθήκευση της τροποποιημένης πινακίδας του χρήστη μέσα στο σύστημα.
- buttonBackPress(): void - Όταν πατηθεί το κουμπί ο χρήστης επιστρέφει στην προηγούμενη σελίδα.



### 1.10.2 Διάγραμμα κλάσεων

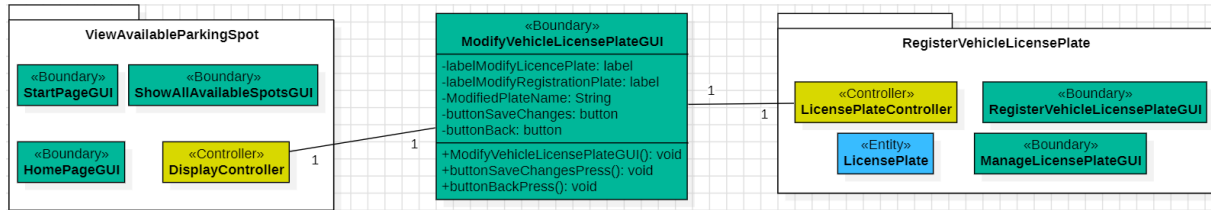


Figure 42: Διάγραμμα κλάσεων feature 10



## 2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

### 2.1 Πρότυπο Proxy

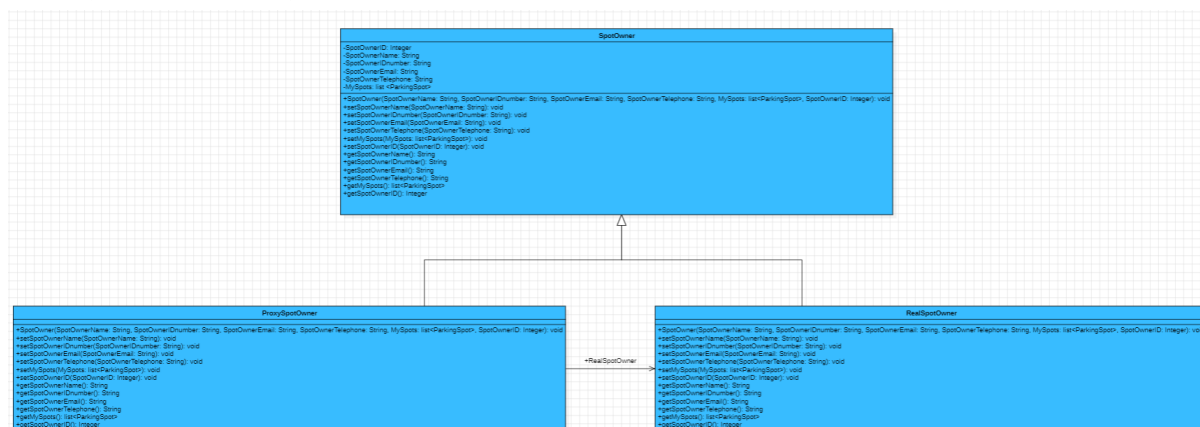


Figure 43: Εφαρμογή προτύπου Proxy

Για να καλύψουμε την ΜΛΑ-1 “Το σύστημα πρέπει να προστατεύει τα προσωπικά δεδομένα των χρηστών” χρησιμοποιούμε το πρότυπο proxy. Το πρότυπο proxy είναι ένα δομικό πρότυπο με το οποίο μπορούμε να εξασφαλίσουμε την ασφάλεια και την αξιοπιστία του συστήματός μας, και πιο συγκεκριμένα να προστατεύσουμε τα προσωπικά δεδομένα των χρηστών κατά τον Γενικό Κανονισμό Προστασίας Δεδομένων (GDPR) της Ευρωπαϊκής ένωσης. Χρησιμοποιούμε ένα protection proxy στην κλάση SpotOwner (η οποία έχει ως μεταβλητές πολλά προσωπικά δεδομένα των χρηστών (SpotOwners) και ελέγχουμε την πρόσβαση σε αντικείμενα της. Επιτρέπουμε δηλαδή μόνο σε ταυτοποιημένους χρήστες (SpotOwners) να έχουν πρόσβαση σε αυτά τα αντικείμενα, ενώ οι υπόλοιποι χειρίζονται ψεύτικα αντίγραφα αυτού (ProxySpotOwners).

Πέρα από την ασφάλεια, η χρήση του προτύπου proxy είναι ιδιαίτερα βοηθητική για την ομαλότερη εμπειρία του χρήστη αφού μπορούμε με την χρήση ενός virtual proxy να διαχειριστούμε περιπτώσεις που το πραγματικό αντικείμενο δεν φορτώνεται, και αντ’ αυτού να χρησιμοποιήσουμε το proxy αυτού ως place holder, έως ότου αντικατασταθεί από το πραγματικό αντικείμενο όταν φορτωθεί. Ο χρήστης δεν μπορεί να καταλάβει ότι δουλεύει με το proxy αντικείμενο και έτσι δεν δυσανασχετεί με το ουσιώδες πρόβλημα φόρτωσης που υπάρχει.

Πηγές για το πρότυπο Proxy: [Wikipedia](#) [Refactoring Guru](#)



## 2.2 Πρότυπο Facade

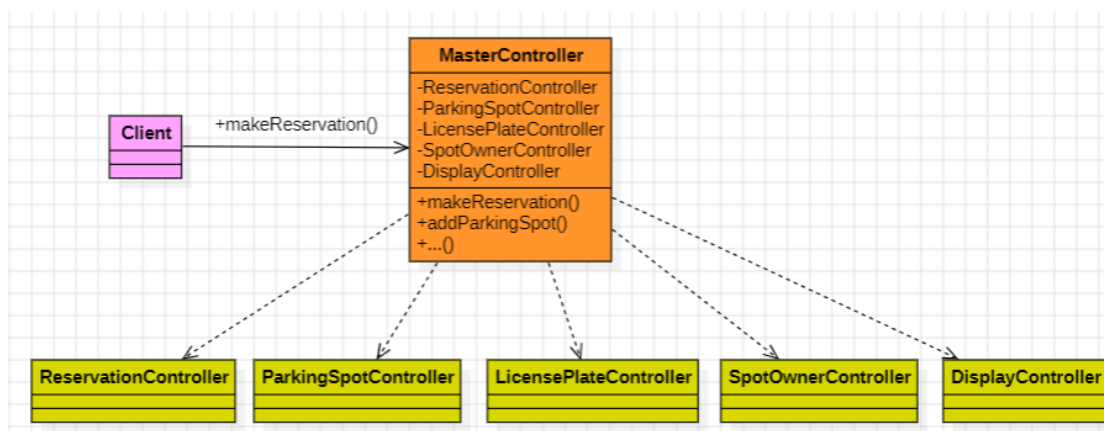


Figure 44: Εφαρμογή προτύπου Facade

Για την ΜΛΑ-2 “Το σύστημα πρέπει να είναι επεκτάσιμο” χρησιμοποιούμε το πρότυπο facade. Το πρότυπο facade είναι ένα δομικό πρότυπο με το οποίο μπορούμε να έχουμε έναν κεντρικό controller για όλους τους controllers του συστήματος. Αυτός ο controller μπορεί να διαχειριστεί και πολλαπλά instances των ίδιων controllers που τρέχουν σε διαφορετικά μηχανήματα. Έτσι μπορώ να μεγαλώσω οριζόντια το σύστημά μου, αυξάνοντας τα instances των controllers, αλλά με μοναδικό Master Controller. Μπορώ και μοιράζω έτσι τα φορτία ισόποσα και έχω ένα επεκτάσιμο σύστημα. Για το συγκεκριμένο παράδειγμα, ο client (που μπορεί να είναι κάποιο boundary αντικείμενο ή άλλη κλάση του συστήματος), καλεί την makeReservation του MasterController, που καλεί μεθόδους των υποκλάσεων του, δηλαδή άλλων controller. Εδώ αναφέρονται δύο μέθοδοι στο MasterController, ενώ στην πραγματικότητα θα ήταν πολλές περισσότερες, και παραλείπονται για λόγους συντομίας.

Πηγές για το πρότυπο Façade: [Wikipedia](#) [Refactoring Guru](#)



## 2.3 Πρότυπο Bridge

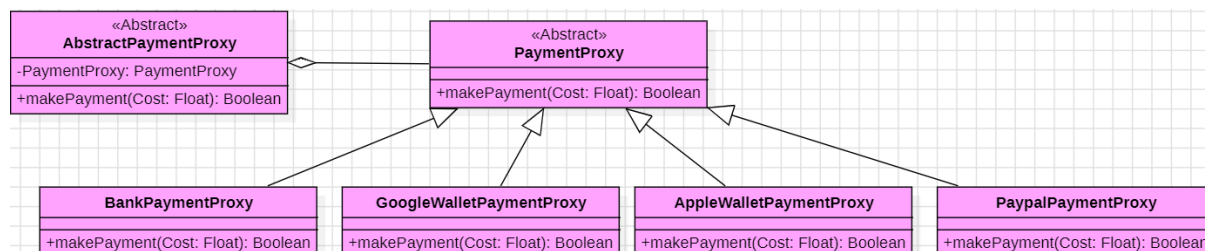


Figure 45: Εφαρμογή προτύπου Bridge

Για την ΜΛΑ “Το σύστημα πρέπει να υποστηρίζει διαφορετικούς τύπους ηλεκτρονικών συναλλαγών” χρησιμοποιούμε το πρότυπο bridge. Το πρότυπο bridge είναι ένα δομικό πρότυπο το οποίο ικανοποιεί την σχεδιαστική ανάγκη του να έχουμε ανεξαρτησία μεταξύ της υλοποίησης και της αφαίρεσης(του βασικού μοντέλου δηλαδή που χρησιμοποιούμε και δεν έχει πολλές λεπτομέρειες, ούτε υλοποίηση).

Ταιριάζει σε ΜΛΑ που αφορούν κλάσεις που μπορούν να χρησιμοποιήσουν σε μεγάλο βαθμό την κληρονομικότητα για να επιτευχθεί η διαφορετικότητα για τις υλοποιήσεις. Έτσι και στην περίπτωση μας, που θέλουμε να υποστηρίξουμε διαφορετικούς τρόπους πληρωμής(μέσω τράπεζας, μέσω GoogleWallet, μέσω AppleWallet, μέσω PayPal) εφαρμόζουμε το παραπάνω πρότυπο για την γενικότερη αφηρημένη έννοια της πληρωμής(abstract κλάση PaymentProxy) με εξειδίκευση στην υλοποίηση τους 4 παραπάνω τρόπους. Με το πρότυπο γέφυρας, το οποίο το επιλέγουμε επιπρόσθετα επειδή γνωρίζουμε εκ των προτέρων τις διαφορετικές υλοποιήσεις που θα έχουμε, κερδίζουμε τόσο σε απλότητα αφού μπορούμε να απομονώνουμε και να ελέγχουμε τις επιμέρους ξεχωριστές υλοποιήσεις, σε ευκολία αφού θα έχουμε επαναχρησιμοποίηση κώδικα για τις υλοποιήσεις που θα ακολουθούν το ίδιο abstraction, και το κυριότερο προσφέρουμε ευελιξία στον χρήστη για να διαλέξει μια από τις διαφορετικές υλοποιήσεις.



### 3 Δυναμική Μοντελοποίηση

#### 3.1 Gherkin Scenario View Available Parking Spots (Feature 4: View available parking spots)

Στο συγκεκριμένο σενάριο, ο χρήστης βρίσκεται στην αρχική σελίδα της εφαρμογής και πατάει το κουμπί All available parking spots για να δει όλες τις διαθέσιμες θέσεις που υπάρχουν στο σύστημα. Το HomePageGUI καλεί την μέθοδο DisplayAllAvailableSpotsPage() του DisplayController, ο οποίος με την σειρά του καλεί την μέθοδο viewAllAvailableSpots του ParkingSpotController. Ο τελευταίος ζητάει από το DBProxy όλες τις διαθέσιμες θέσεις που υπάρχουν στο σύστημα. Το DBProxy χρησιμοποιεί την μέθοδο getSpot(All=True) για να επιστρέψει όλες τις διαθέσιμες θέσεις στο ParkingSpotController σε μορφή λίστας. Αφού ο ParkingSpotController αποκτήσει την λίστα με τις διαθέσιμες θέσεις, καλεί τον DisplayController για να εμφανίσει τις θέσεις αυτές στον χρήστη. Ο DisplayController χρησιμοποιεί την μέθοδο DisplayAllAvailableSpotsPage() για να καλέσει τον constructor του ShowAllAvailableSpotsGUI και με αυτόν τον τρόπο να δημιουργήσει τη σελίδα στην οποία θα προβληθεί η λίστα με όλες τις διαθέσιμες θέσεις. Τέλος εμφανίζεται το μήνυμα Showing all available spots.

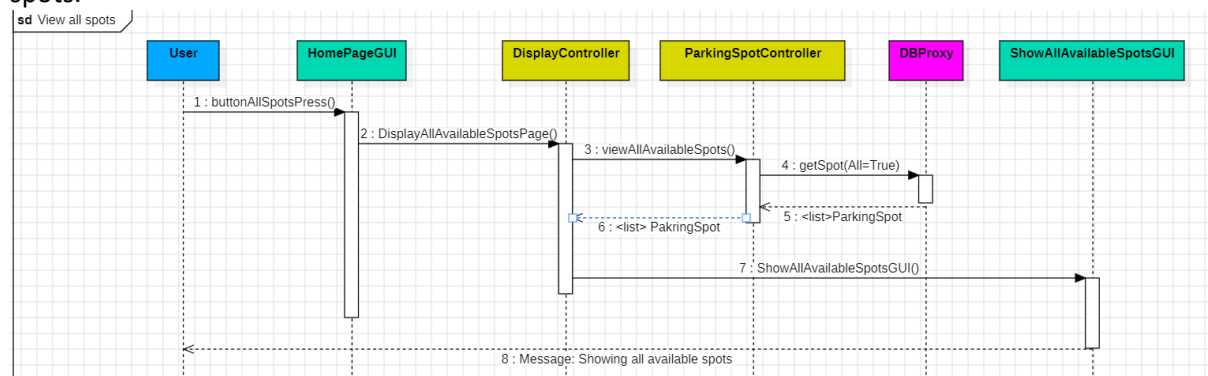


Figure 46: Δυναμική μοντελοποίηση για το Feature 4

#### 3.2 Gherkin Scenario Reserve Parking Spot (Feature 1: Reserve a parking spot)

Για το συγκεκριμένο σενάριο, ο χρήστης βρίσκεται στην σελίδα με τις πληροφορίες της θέσης, και πατάει το κουμπί Reserve spot. Το SpotInformationGUI καλεί την μέθοδο του DisplayController και αυτός με την σειρά του δημιουργεί το MakeReservationGUI. Ο χρήστης συμπληρώνει τις απαραίτητες πληροφορίες και πατάει το κουμπί Make Reservation. Καλείται η μέθοδος του reserveParkingSpot του ReservationController, που δημιουργεί ένα αντικείμενο Reservation. Το αντικείμενο επιστρέφεται στον controller, που καλεί την μέθοδο putReservation του DBProxy για να αποθηκεύσει την κράτηση στην βάση. Τέλος καλείται η μέθοδος displayHomePage του DisplayController, φαίνεται μήνυμα Information saved και ο χρήστης επιστρέφει στην κεντρική σελίδα.

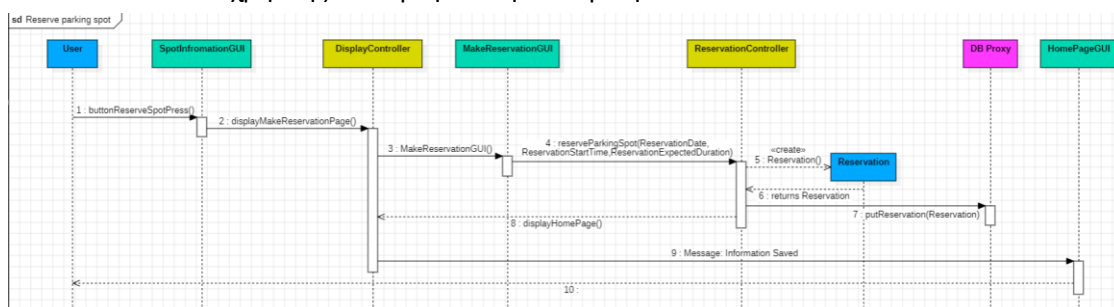


Figure 47: Δυναμική μοντελοποίηση για το Feature 1



### 3.3 Gherkin Scenario Add Parking Spots (Feature 7: Add parking spots)

Για το συγκεκριμένο σενάριο, ο Spot Owner βρίσκεται στην διαμορφωμένη για τον ρόλο αυτό σελίδα και πατάει το κουμπί Add Spot καλώντας τη μέθοδο `buttonAddSpotPress` του `SpotOwnerPageGUI`. Το GUI αυτό, με τη σειρά του, επικοινωνεί με τον `ParkingSpotController` καλώντας την μέθοδό του `addParkingSpot`. Στη συνέχεια, ο `ParkingSpotController` καλεί την μέθοδο του `DisplayController` για να μεταφερθεί ο `SpotOwner` στην οθόνη που θα καταχωρήσει τις πληροφορίες της νέας θέσης πάρκινγκ. Έπειτα δημιουργείται το `AddSpotInformationGUI` και όταν πατηθεί το κουμπί για αποθήκευση αλλαγών, το GUI δημιουργεί ένα αντικείμενο της Entity κλάσης `ParkingSpot` το οποίο μετά επιστρέφει στον `ParkingSpotController` με την ομώνυμη μέθοδο. Στη συνέχεια ο `ParkingSpotController` καταχωρεί στο `DBProxy` την νέα θέση με την χρήση της μεθόδου της `putParkingSpot`. Αφού γίνει η καταχώρηση, ο `SpotOwner` μεταβαίνει στο `HomePageGUI`, μέσω της μεθόδου του `DisplayController` `displayHomePage` και του εμφανίζεται το μήνυμα `Spot added successfully`.

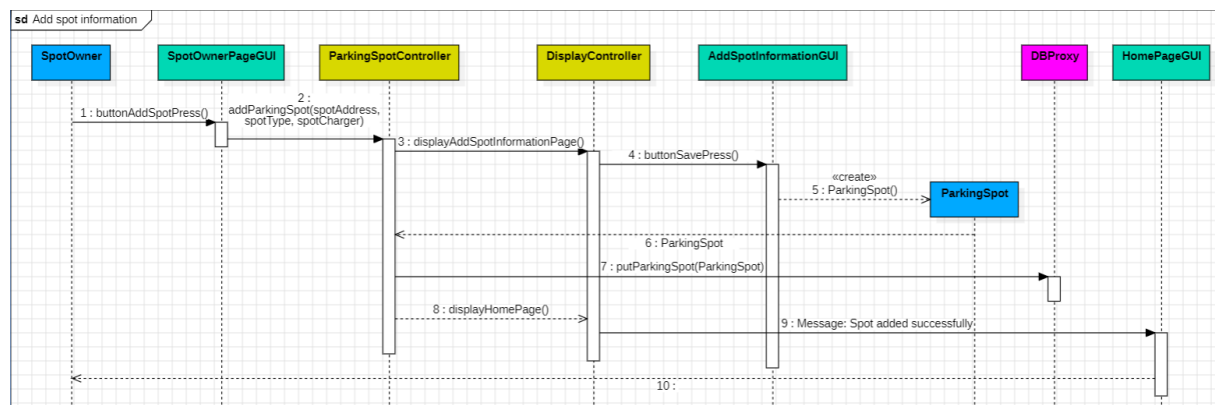


Figure 48: Δυναμική μοντελοποίηση για το Feature 7





## Παράρτημα Ι – Γλωσσάριο

|          |                                                                                                                                               |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| ΛΑ-xxxx  | Λειτουργική Απαίτηση xxxx                                                                                                                     |
| ΜΛΑ-xxxx | Μη Λειτουργική Απαίτηση xxxx                                                                                                                  |
| ΟΑ       | Ομάδα Ανάπτυξης                                                                                                                               |
| GUI      | Graphical user interface, γραφικό περιβάλλον διεπαφής του χρήστη.                                                                             |
| ID       | Μοναδικό αναγνωριστικό πχ. Το UserID είναι το μοναδικό αναγνωριστικό του χρήστη.                                                              |
| DB       | DataBase, Βάση δεδομένων                                                                                                                      |
| GDPR     | General Data Protection Regulation, Αφορά τον Κανονισμό της Ευρωπαϊκής Ένωσης 2016/679 για την επεξεργασία των δεδομένων Προσωπικού Χαρακτήρα |



## Παράρτημα II – Ανοιχτά Θέματα

- Το κομμάτι της δυναμικής μοντελοποίησης δεν καλύπτει όλα τα Gherkin scenarios που έχουν οριστεί στο πρώτο παραδοτέο.
- Το κομμάτι των design patterns δεν καλύπτει όλες τις ΜΛΑ που έχουν οριστεί στο πρώτο παραδοτέο.