



# Σύστημα *ThessPark*

We spot your parking place.

## Απαιτήσεις Λογισμικού (Προδιαγραφές Λογισμικού)

### Del.2.1

Version 1.1

Μουρούζης Χρήστος, [chrimour@auth.gr](mailto:chrimour@auth.gr)  
Τσακάλης Κωνσταντίνος, [tsakalis@auth.gr](mailto:tsakalis@auth.gr)  
Μιχαήλ Μάριος, [mamichai@auth.gr](mailto:mamichai@auth.gr)  
Αγρότης Ιωάννης, [iagrotis@auth.gr](mailto:iagrotis@auth.gr)

15/6/2015



## Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
A. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers <sup>1♣</sup> και του M. Smialek's.	0.1
A. Συμεωνίδης	29/3/2014	Προσαρμογή του εγγράφου.	0.1.3
Χ. Ζολώτας	2/4/2015	Προσθήκη διαγράμματος καταστάσεων	0.1.4
Κ. Τσακάλης, Μιχαήλ Μ.	25/4/2015	Προσθήκη Πακέτων Κλάσεων	0.2
Μουρούζη Χ, Αγρότης Γ.	1/5/2015	Προσθήκη Διαγραμμάτων Ακολουθιών	0.5
ΟΑ	3/5/2015	Διόρθωση εγγράφου	1.0
ΟΑ	15/6/2015	Διόρθωση εγγράφου	1.1

## Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Θ.Μαυρίδης	*	<a href="mailto:themis.mavridis@issel.ee.auth.gr">themis.mavridis@issel.ee.auth.gr</a>
Κωνσταντίνος Τσακάλης	27	<a href="mailto:tsakalis@auth.gr">tsakalis@auth.gr</a>
Μιχαήλ Μάριος	27	<a href="mailto:mamichai@auth.gr">mamichai@auth.gr</a>
Μουρούζη Χρήστος	27	<a href="mailto:chrimour@auth.gr">chrimour@auth.gr</a>
Αγρότης Ιωάννης	27	<a href="mailto:iagrotis@auth.gr">iagrotis@auth.gr</a>

<sup>1</sup> ♣ Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document. Original template is available at: <http://www.processimpact.com/>



## ΠΕΡΙΕΧΟΜΕΝΑ

Λίστα Σχημάτων .....	3
1 Εισαγωγικά .....	4
1.1 Στόχος του Εγγράφου .....	4
1.2 Τυπογραφικές Παραδοχές του Εγγράφου .....	4
1.3 Αναγνωστικό κοινό και τρόπος ανάγνωσης .....	4
1.4 Σκοπός του Έργου .....	4
2 Στατική Μοντελοποίηση .....	6
2.1 Πακέτο Βασικής Διεπαφής Χρήστη .....	6
2.1.1 Ορισμοί .....	6
2.1.1.1 Πακέτο Users .....	6
2.1.1.2 Πακέτο GUI .....	10
2.2 Πακέτο Utilities .....	18
2.2.1 Πακέτο User Utilities .....	19
2.2.2 Πακέτο ParkingAdmin Utilities .....	23
2.2.3 Πακέτο Supervisor Utilities .....	24
2.3 Πακέτο SystemControl .....	26
2.4 Πακέτο DataBase .....	30
2.5 Πακέτο API .....	33
2.6 Διαγράμματα κλάσεων .....	35
3 Απαιτήσεις Λογισμικού .....	39
3.1 Δυναμική μοντελοποίηση του συστήματος .....	39
3.1.1 Πακέτο Αρχική Οθόνη και Είσοδος χρήστη .....	39
3.1.1.1 Αφήγηση Σεναρίου: .....	39
3.1.2 Πακέτο Αναβάθμισης Βάσης Δεδομένων Parkings .....	41
3.1.2.1 Αφήγηση Σεναρίου: .....	41
3.1.3 Πακέτο Λειτουργιών Απλού Χρήστη .....	42
3.1.3.1 Αφήγηση Σεναρίου: .....	42
3.1.4 Πακέτο Κράτησης θέσης .....	44
3.1.4.1 Αφήγηση Σεναρίου: .....	44
3.1.5 Πακέτο Τρέχουσας Χρέωσης .....	45
3.1.5.1 Αφήγηση Σεναρίου: .....	45
3.2 Διάγραμμα Καταστάσεων .....	46
3.3 Μη λειτουργικές απαιτήσεις .....	47
3.3.1 Απαιτήσεις επίδοσης .....	47
3.3.2 Απαιτήσεις ασφάλειας (Security) .....	47
3.3.3 Απαιτήσεις Χρηστικότητας (Usability) .....	47
3.3.4 Απαιτήσεις Αξιοπιστίας (Reliability) .....	47
Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας .....	48
Παράρτημα ΙΙ – Ανοιχτά Θέματα .....	49



## Λίστα Σχημάτων

Σχήμα 1: Διάγραμμα Πακέτου Users .....	6
Σχήμα 2: Οντότητα User .....	7
Σχήμα 3: Οντότητα Parking Admin .....	8
Σχήμα 4: Οντότητα Supervisor .....	9
Σχήμα 5: Διάγραμμα Πακέτου GUI .....	10
Σχήμα 6: Διάγραμμα Κλάσης Log In GUI .....	11
Σχήμα 8: Διάγραμμα Κλάσης Create Account GUI .....	14
Σχήμα 9: Διάγραμμα Κλάσης Supervisor GUI .....	15
Σχήμα 11: Διάγραμμα Πακέτου Utilities .....	18
Σχήμα 12: Διάγραμμα Πακέτου Users Utilities .....	19
Σχήμα 13: Διάγραμμα Κλάσης Position .....	19
Σχήμα 14: Διάγραμμα Κλάσης Booking .....	20
Σχήμα 15: Διάγραμμα Κλάσης Cost .....	21
Σχήμα 16: Διάγραμμα Κλάσης Parking .....	22
Σχήμα 17: Διάγραμμα Πακέτου Parking Administrator Utilities .....	23
Σχήμα 18: Διάγραμμα Κλάσης Prices .....	23
Σχήμα 19: Διάγραμμα Κλάσης Availability .....	24
Σχήμα 20: Διάγραμμα Πακέτου Supervisor Utilities Package .....	24
Σχήμα 21: Διάγραμμα Κλάσης System Maintenance .....	25
Σχήμα 22: Διάγραμμα Κλάσης Status .....	25
Σχήμα 23: Διάγραμμα Πακέτου System Control .....	26
Σχήμα 24: Διάγραμμα Κλάσης System Overview .....	27
Σχήμα 25: Διάγραμμα Κλάσης Booking .....	27
Σχήμα 26: Διάγραμμα Κλάσης API Status .....	28
Σχήμα 27: Διάγραμμα Κλάσης UserUtilControl .....	28
Σχήμα 28: Διάγραμμα Κλάσης LoginControl .....	29
Σχήμα 29: Διάγραμμα Κλάσης UpdateDBControl .....	30
Σχήμα 30: Διάγραμμα Πακέτου DB .....	31
Σχήμα 31: Διάγραμμα Κλάσης Users .....	31
Σχήμα 32: Διάγραμμα Κλάσης Parkings .....	32
Σχήμα 33: Διάγραμμα Πακέτου API .....	33
Σχήμα 34: Διάγραμμα Κλάσης GoogleMaps .....	33
Σχήμα 35: Διάγραμμα Κλάσεων .....	35
Σχήμα 36: Διάγραμμα Κλάσεων Πακέτου Βασικής Διεπαφής Απλού Χρήστη .....	36
Σχήμα 37: Διάγραμμα Κλάσεων Πακέτου διεπαφής Διαχειριστή Χώρου Στάθμευσης .....	37
Σχήμα 38: Διάγραμμα Κλάσεων Πακέτου διεπαφής Διαχειριστή Συστήματος .....	38
Σχήμα 39: Διάγραμμα ακολουθιών προβολής αρχικής οθόνης – εισόδου χρήστη .....	40
Σχήμα 40: Διάγραμμα ακολουθιών Αναβάθμισης Βάσης Δεδομένων Parkings .....	41
Σχήμα 41: Διάγραμμα ακολουθιών Λειτουργιών Απλού χρήστη .....	43
Σχήμα 42: Διάγραμμα ακολουθιών Κράτησης θέσης .....	44
Σχήμα 43: Διάγραμμα ακολουθιών τρέχουσας χρέωσης .....	45
Σχήμα 45: Διάγραμμα καταστάσεων .....	46



# 1 Εισαγωγικά

## 1.1 Στόχος του Εγγράφου

Το παρόν έγγραφο <<Απαιτήσεις Λογισμικού>> αποτελεί συνέχεια του πρώτου παραδοτέου <<Απαιτήσεις Χρηστών>>. Στόχος του εγγράφου είναι η μοντελοποίηση όλων εκείνων των απαιτήσεων χρηστών από τη πλευρά των λειτουργιών συστήματος όπως αυτές έχουν περιγραφεί στο πρώτο παραδοτέο.

Ο κύριος κορμός του εγγράφου είναι το πως θα λειτουργεί ακριβώς το σύστημα, τι δεδομένα θα διαχειρίζεται και ποια άλλα λεπτομερή χαρακτηριστικά θα πρέπει να έχει. Συγκεκριμένα, αναπτύσσονται λεπτομερή σενάρια και αφηγήσεις σχετικά με τη δυναμικότητα του περιβάλλοντος διεπαφής και τη συμπεριφορά του συστήματος, λεπτομερείς περιγραφές των τύπων δεδομένων που ανταλλάσσονται με τους χρήστες και άλλα συστήματα, των μενού και των φορμών μέσω των οποίων το σύστημα αλληλεπιδρά με τους χρήστες και λοιπές λεπτομερείς περιγραφές των μη-λειτουργικών χαρακτηριστικών και περιορισμών.

Η ανάλυση που επιχειρούμε διαχωρίζει το σύστημα σε διακριτά μέρη, όπως αυτά περιγράφονται από τις κλάσεις οντοτήτων, τις κλάσεις ελέγχου και τις οριακές κλάσεις. Ακόμη παρουσιάζεται και η δυναμική μοντελοποίηση του συστήματος. Συγκεκριμένα, επιχειρούμε την τεκμηρίωση της ορθότητας και πληρότητας των κλάσεων ως προς τα σενάρια χρήσης που αναπτύχθηκαν στο έγγραφο απαιτήσεων χρηστών. Αυτό επιτυγχάνεται μέσω των διαγραμμάτων ροών που περιγράφουν πλήρως τη συμπεριφορά του συστήματος κατά την εκτέλεση των διαφόρων λειτουργιών του και κατά την αλληλεπίδρασή του με τους χρήστες και τα εξωτερικά συστήματα.

## 1.2 Τυπογραφικές Παραδοχές του Εγγράφου

Το κείμενο του εγγράφου τούτου είναι γραμμένο με γραμματοσειρά Callibri, μεγέθους 11pt. Οι οδηγίες και οι παρατηρήσεις σημειώνονται με αγκύλες (<>). Οι απαιτήσεις στο κεφάλαιο 3 πρέπει να ονομάζονται και να αριθμούνται κατάλληλα και να συντάσσονται με την κατάλληλη μορφή. Για τις λεζάντες των διαγραμμάτων χρησιμοποιείται γραμματοσειρά Calibri, μεγέθους 11pt και πλάγια γραφή.

## 1.3 Αναγνωστικό κοινό και τρόπος ανάγνωσης

Το έγγραφο αυτό ως κομμάτι του συνολικού έργου απευθύνεται σε όποιον ενδιαφερόμενο σχετικά με αυτό. Καθώς όμως τα ζητήματα τα οποία αναλύονται είναι πιο λεπτομερή και έχουν να κάνουν με την υλοποίηση και τη λειτουργία του ίδιου του συστήματος, ένας απλός αναγνώστης ίσως δυσκολευτεί στην παρακολούθηση της αφήγησης. Έτσι, το έγγραφο απευθύνεται περισσότερο στους μηχανικούς λογισμικού, σχεδιασμού και ανάπτυξης οι οποίοι θα ασχοληθούν με την υλοποίηση του προγράμματος, καθώς επίσης και στους πιθανούς ενδιαφερόμενους επενδυτές οι οποίοι μπορεί να θέλουν πλήρη προσωπική εποπτεία του προγράμματος και ίσως να προτείνουν κάποιες αλλαγές στην ανάπτυξη της εφαρμογής.

## 1.4 Σκοπός του Έργου

Τα τελευταία χρόνια, παράλληλα με την εξέλιξη του λογισμικού των κινητών τηλεφώνων έχει αναπτυχθεί μία πληθώρα εφαρμογών γενικής χρήσης. Άλλες από αυτές απευθύνονται σε παιδιά, άλλες



σε ενήλικες, άλλες σε εξειδικευμένους κλάδους, άλλες σε επιχειρηματίες και ούτω καθ' εξής. Πηγή έμπνευσης της δικής μας εφαρμογής ήταν ο σχεδιασμός ενός έργου το οποίο θα έχει απήχηση στο ευρύ κοινό. Αναλογιζόμενοι ποιο θα μπορούσε να είναι ένα τέτοιο έργο, το σκεπτικό μας βασίστηκε στην εύρεση μιας ιδέας απλής και λειτουργικής. Μιας εφαρμογής η οποία θα αποτελεί εργαλείο και βοηθό του χρήστη στην καθημερινότητά του. Μια τέτοια περιοχή εργασίας είναι αδιαμφισβήτητα τα κοινωνικά προβλήματα.

Εστιάζοντας στο πεδίο της αστικής μετακίνησης της πόλης μας, παρατηρήθηκε το μείζον πρόβλημα της εύρεσης χώρου στάθμευσης μέσα στην πόλη. Αν κάποιος θα ήθελε να ασχοληθεί με την αντιμετώπιση αυτού του προβλήματος τότε η πόλη της Θεσσαλονίκης αποτελεί σίγουρα το ιδανικό παράδειγμα. Εκτός απ τα κυκλοφοριακά προβλήματα που αντιμετωπίζει καθημερινά ο μέσος πολίτης (δυσκολία κατά τη μετακίνηση, συνωστισμός στα μέσα μαζικής μεταφοράς, κυκλοφοριακή συμφόρηση στους κεντρικούς δρόμους της πόλης) η χρήση του αυτοκινήτου ιδιωτικής χρήσης φαντάζει απαγορευτική κατά τις πρωινές και μεσημβρινές ώρες. Δεν είναι άλλος ο λόγος από τη δυσκολία εύρεσης θέσης στάθμευσης. Κατά τις ώρες αιχμής τις εργάσιμες ημέρες καθώς και τα Σαββατοκύριακα, όλη η πόλη γίνεται ένα απέραντο parking (χώρος στάθμευσης) και πολλοί θα έλεγαν ότι οι πολίτες της Θεσσαλονίκης κερδίζουν το βραβείο εφευρετικότητας στο συγκεκριμένο θέμα. Αυτό έχει ως αποτέλεσμα να παρατηρούνται σταθμεύσεις επάνω σε διαβάσεις πεζών, μπροστά σε ράμπες διέλευσης αναπηρικών αμαξιδίων από άτομα με ειδικές ανάγκες δυσχεραίνοντας έτσι ακόμη περισσότερο τη μετακίνησή τους, διπλοπαρκαρίσματα και η λίστα δεν έχει τέλος. Αυτά τα παραδείγματα καθιστούν εμφανή την ανάγκη για ένα νέο ενιαίο σύστημα το οποίο θα συμβάλλει ενεργά και λειτουργικά στην καταπολέμησή τους και στην αποσυμφόρηση των δρόμων και πεζοδρομίων από τα παράνομα σταθμευμένα οχήματα. Το ρόλο αυτού του συστήματος φιλοδοξεί να εκπληρώσει το ThessPark.



## 2 Στατική Μοντελοποίηση

### 2.1 Πακέτο Βασικής Διεπαφής Χρήστη

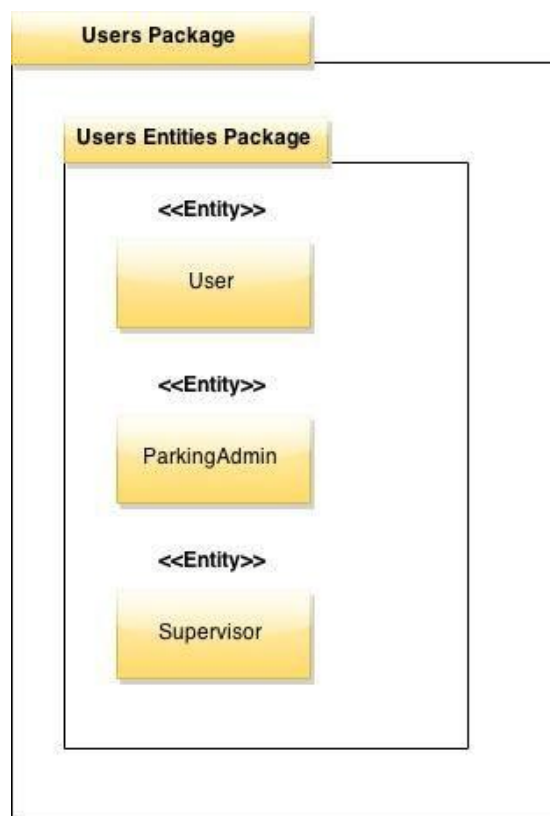
Κεντρικό κομμάτι του πακέτου αυτού είναι η βασική διεπαφή και οι επιλογές στις οποίες έχει πρόσβαση μέσω αυτής ο χρήστης. Παρακάτω ορίζονται οι κλάσεις που απαιτούνται για την υλοποίηση των δυνατοτήτων αυτών και δίνεται λεκτική περιγραφή για τη λειτουργικότητα του κάθε πακέτου, κλάσης καθώς και των χαρακτηριστικών και των μεταβλητών τους.

#### 2.1.1 Ορισμοί

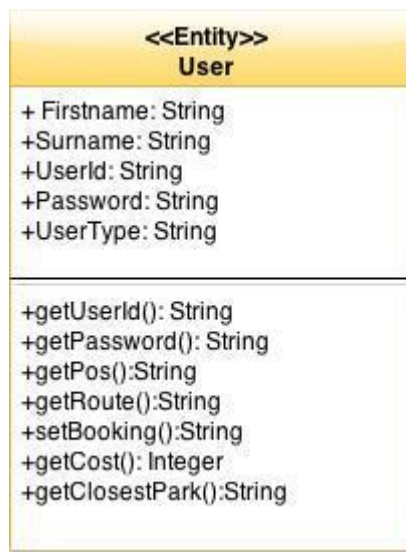
Στη συνέχεια παρουσιάζονται οι ακριβείς λεξικογραφικοί ορισμοί για κάθε κλάση.

##### 2.1.1.1 Πακέτο Users

Ο χρήστης μετά το άνοιγμα του προγράμματος αντιμετωπίζεται ως guest όπου του δίνεται η δυνατότητα να επιλέξει ως προς ποια ιδιότητα επιθυμεί να εισέλθει στο σύστημα δίνοντας του τα απαραίτητα στοιχεία που έχουν οριστεί για κάθε οντότητα. Οι κατηγορίες των χρηστών αυτών μετά την είσοδο τους στο σύστημα αντιπροσωπεύονται από τις κλάσεις User, ParkingAdmin καθώς και Supervisor. Η κλάση User είναι η βασική κλάση η οποία κληρονομείται από όλες τις υπόλοιπες.



Σχήμα 1: Διάγραμμα Πακέτου Users

**Entity User**

Σχήμα 2: Οντότητα User

Είναι η βασική κλάση χρηστών. Αποτελείται από όλες τις κοινές μεταβλητές και συναρτήσεις για όλους τους απλούς χρηστών του συστήματος.

Τα χαρακτηριστικά της οντότητας αυτής είναι:

- ☐ Firstname: Το μικρό όνομα του χρήστη.
- ☐ Surname: Το επίθετο του χρήστη.
- ☐ UserId: Το ψευδώνυμο του κάθε χρήστη, μοναδικό για κάθε άτομο.
- ☐ Password: Ο κωδικός πρόσβασης του χρήστη.
- ☐ UserType: Περιέχει τον τύπο του χρήστη.

Οι μέθοδοι της οντότητας αυτής είναι:

- ☐ getUserId(): Επιστρέφει τα στοιχεία του χρήστη.
- ☐ getPassword(): Επιστρέφει ένα καινούργιο κωδικό στο χρήστη.
- ☐ getPos(): Επιστρέφει την τρέχουσα θέση του χρήστη.
- ☐ getRoute(): Επιστρέφει την διαδρομή που πρέπει να ακολουθήσει ο χρήστης για να φθάσει στο προορισμό του.
- ☐ setBooking(): Ενημερώνει το χρήστη για την αίτηση κράτησης μιας θέσεως και πραγματοποιεί εάν είναι δυνατό τη κράτηση αυτή.
- ☐ getCost(): Επιστρέφει την τρέχουσα χρέωση του χρήστη.
- ☐ getClosestPark(): Επιστρέφει το κοντινότερο χώρο στάθμευσης του χρήστη, βάση της τοποθεσίας που βρίσκεται.



**Entity ParkingAdmin**

Σχήμα 3: Οντότητα Parking Admin

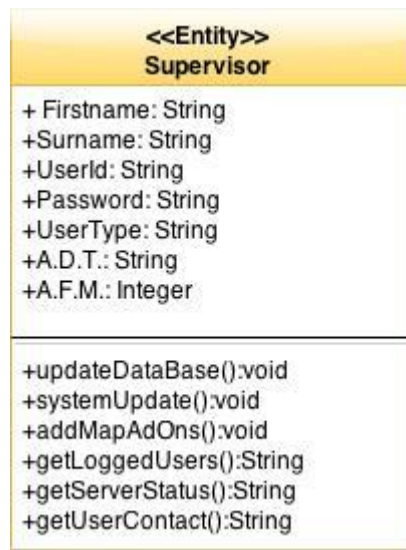
Αντιπροσωπεύει το χρήστη διαχειριστή Parking. Στο χρήστη αυτό δίνονται διαφορετικές δυνατότητες από αυτές του απλού χρήστη. Μερικές από τις δυνατότητες αυτές είναι η ενημέρωση των εναπομεινάντων θέσεων στάθμευσης, η ενημέρωση για τυχών προσφορών που είναι διαθέσιμες, καθώς και τις τρέχουσες χρεώσεις του χώρου στάθμευσης.

Τα χαρακτηριστικά της οντότητας αυτής είναι:

- ☐ Firstname: Το μικρό όνομα του διαχειριστή.
- ☐ Surname: Το επίθετο του διαχειριστή
- ☐ UserId: Το ψευδώνυμο του διαχειριστή.
- ☐ Password: Ο κωδικός πρόσβασης του διαχειριστή.
- ☐ UserType: Περιέχει τον τύπο του χρήστη.
- ☐ A.D.T.: Ο αριθμός δελτίου ταυτότητας του χρήστη.

Οι μέθοδοι της οντότητας αυτής είναι:

- ☐ updateOffers(): Ενημερώνει τις τρέχουσες προσφορές που υπάρχουν στο χώρο στάθμευσης
- ☐ updatePrices(): Ενημέρωση των τιμών χρέωσης του χώρου.
- ☐ updateTotParkPlaces(): Ενημέρωση για την διαθεσιμότητα του χώρου στάθμευσης.

**Entity Supervisor**

Σχήμα 4: Οντότητα Supervisor

Αντιπροσωπεύει το χρήστη Supervisor. Ο Supervisor είναι ο χρήστης που έχει πλήρη πρόσβαση σε όλα τα μέρη του προγράμματος. Βασικές λειτουργίες του είναι η διαχείριση του συστήματος, η δυνατότητα αναβάθμισης του, η δημιουργία χρηστών, η επεξεργασία των στοιχείων τους καθώς και η διαγραφή τους.

Τα χαρακτηριστικά της οντότητας αυτής είναι:

- ☐ Firstname: Το μικρό όνομα του supervisor.
- ☐ Surname: Το επίθετο του supervisor.
- ☐ UserId: Το ψευδώνυμο του supervisor.
- ☐ Password: Ο κωδικός εισόδου του supervisor.
- ☐ UserType: Περιέχει τον τύπο του χρήστη.
- ☐ A.D.T.: Ο αριθμός δελτίου ταυτότητας του supervisor.
- ☐ A.F.M.: Ο αριθμός φορολογικής δήλωσης του supervisor.

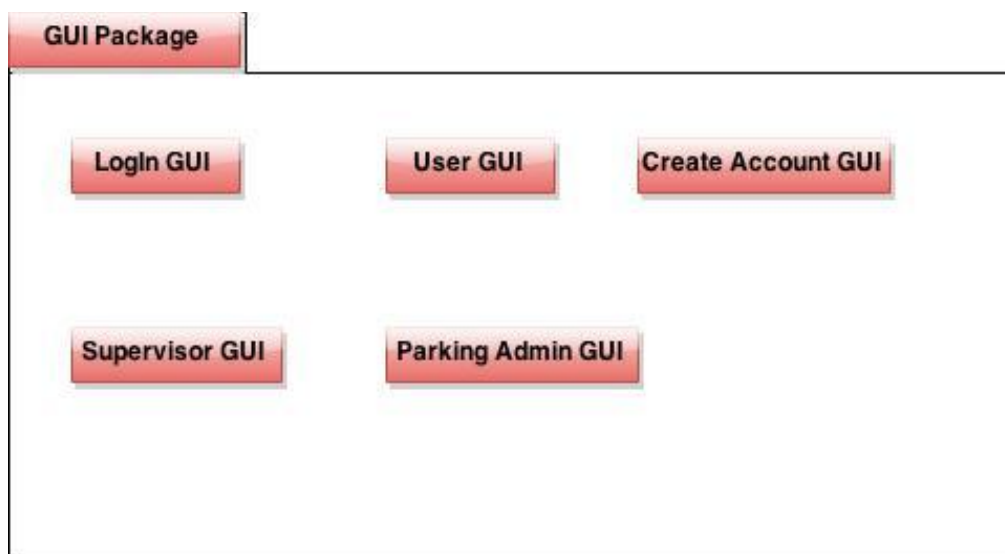
Οι μέθοδοι της οντότητας αυτής είναι:

- ☐ updateDatabase(): Ενημέρωση της βάσης Δεδομένων της εφαρμογής.
- ☐ systemUpdate(): Γενική Εποπτεία και συντήρηση του συστήματος.
- ☐ addMapAdOns(): Ενημέρωση Διαδραστικών Στοιχείων στο χάρτη.
- ☐ getLoggedUsers(): Εμφάνιση των χρηστών που βρίσκονται στο σύστημα.
- ☐ getServerStatus(): Εμφάνιση ιστορικού κατάστασης του server.
- ☐ getUserContact(): Εμφάνιση στοιχείων επικοινωνίας διαχειριστών χώρων στάθμευσης.



### 2.1.1.2 Πακέτο GUI

Το συγκεκριμένο πακέτο αναφέρεται στο GUI (Graphical User Interface), δηλαδή στο γραφικό περιβάλλον διεπαφής του χρήστη με το σύστημα προς υλοποίηση. Ως εκ τούτου, τα αντικείμενα των κλάσεων που ανήκουν στο εν λόγω πακέτο θεωρούνται οριακών συνθηκών (<<boundary>>), μιας και έχουμε την άμεση επαφή του συστήματος με τον χρήστη. Συγκεκριμένα, για κάθε κατηγορία του συστήματος διεπαφής, έχουμε και μια ξεχωριστή κλάση. Ύστερα από θεωρητική ανάλυση, η ανάγκη δημιουργίας των εξής κλάσεων: Login GUI, User GUI, ParkingAdmin GUI, Supervisor GUI, Create Account GUI. Παρακάτω, παρουσιάζεται μια οπτικοποίηση του πακέτου και των κλάσεων που περιλαμβάνει.



Σχήμα 5: Διάγραμμα Πακέτου GUI

**Boundary Class Log In GUI**

Σχήμα 6: Διάγραμμα Κλάσης Log In GUI

Τα χαρακτηριστικά της κλάσης αυτής είναι:

- ❑ Logo: Το logo του ThessPark με το πάτημα του οποίου ο χρήστης θα μεταβαίνει στην αρχική σελίδα της εφαρμογής.
- ❑ User: Πεδίο επιλογής του χρήστη για εισαγωγή στο σύστημα ως απλός χρήστης.
- ❑ Parking Administrator: Πεδίο επιλογής του χρήστη για εισαγωγή στο σύστημα ως διαχειριστής κάποιου parking.
- ❑ Supervisor: Πεδίο επιλογής του χρήστη για εισαγωγή στο σύστημα ως γενικός διαχειριστής.
- ❑ Continue: Κουμπί με το πάτημα του οποίου γίνεται εισαγωγή του χρήστη ανάλογα με τα χαρακτηριστικά που έχει επιλέξει.
- ❑ Terms of Use: Κουμπί με το πάτημα του οποίου εμφανίζονται στην οθόνη σε μορφή απλού κειμένου οι όροι χρήσης της εφαρμογής ThessPark.
- ❑ UserID: Πεδίο εισαγωγής κειμένου στο οποίο ο χρήστης πληκτρολογεί το ψευδώνυμό του (με ελάχιστο μέγεθος 6 χαρακτήρες). Θα χρησιμοποιηθεί για την ταυτοποίηση του χρήστη με τη βάση δεδομένων.
- ❑ Password: Πεδίο εισαγωγής κειμένου (καθώς ο κωδικός του χρήστη πρέπει να αποτελείται από τουλάχιστον 1 κεφαλαίο γράμμα και οποιοδήποτε συνδιασμό γραμμάτων και αριθμών) στο οποίο ο χρήστης πληκτρολογεί τον κωδικό του. Θα χρησιμοποιηθεί για την ταυτοποίηση του χρήστη με τη βάση δεδομένων.

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ refresh(): Συνάρτηση η οποία καλείται με το πάτημα του κουμπιού logo η οποία είναι υπεύθυνη να ανανεώσει την αρχική σελίδα που βρίσκεται ο χρήστης.



- ❑ termsOfUse(): Συνάρτηση η οποία καλείται με το πάτημα του κουμπιού termofuse και είναι υπεύθυνη να εμφανίσει στο χρήστη σε μορφή λίστας κειμένου όλους τους όρους χρήσης της εφαρμογής μας.
- ❑ signIn(userType,userId,password): Δέχεται ως ορίσματα το πεδίο ιδιότητας που επέλεξε ο χρήστης, το ψευδώνυμο και τον κωδικό του. Η συνάρτηση αυτή καλείται με το πάτημα του κουμπιού continue και είναι υπεύθυνη να επικοινωνήσει με την βάση δεδομένων για ταυτοποίηση των στοιχείων που έχει εισάγει ο χρήστης, καθώς και σε επιτυχή ταυτοποίηση να οδηγήσει το χρήστη στο επόμενο μενού επιλογών που αντιστοιχεί σ αυτόν.

### Boundary Class User GUI



Σχήμα 7: Διάγραμμα Κλάσης User GUI

Τα χαρακτηριστικά της κλάσης αυτής είναι:

- ❑ Logo: Το logo του ThessPark με το πάτημα του οποίου ο χρήστης θα μεταβαίνει στην αρχική σελίδα της εφαρμογής.
- ❑ Username: Πεδίο εμφάνισης κειμένου όπου εμφανίζεται στο χρήστη το username του (ψευδώνυμο) που έχει επιλέξει.
- ❑ ClosestPark: Κουμπί με την ενεργοποίηση του οποίου εμφανίζονται στο χρήστη όλοι οι κοντινοί χώροι στάθμευσης βάση της τοποθεσίας που βρίσκεται.
- ❑ Drivemethere: Κουμπί με την ενεργοποίηση του οποίου εμφανίζεται στο χάρτη που υπάρχει η βέλτιστη διαδρομή που πρέπει να ακολουθήσει για να φθάσει στον προορισμό του.
- ❑ Online Park-o-meter: Κουμπί με την ενεργοποίηση του οποίου εμφανίζεται στην οθόνη του χρήστη (σε μορφή μηνύματος) η τρέχουσα χρέωση του.
- ❑ BookOnline: Κουμπί με την ενεργοποίηση του οποίου ο χρήστης κάνει αίτηση για κράτηση μιας θέσεως στάθμευσης σε χώρο της επιλογής του.



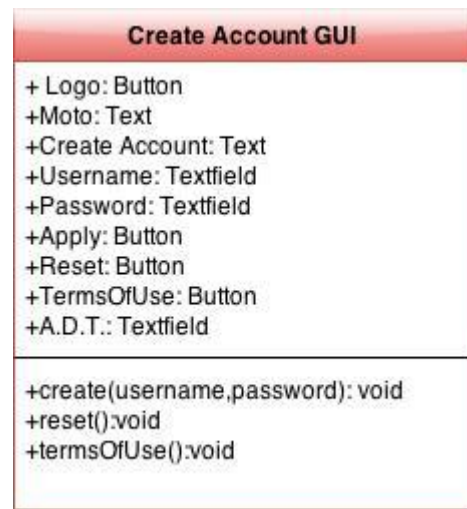
- ❑ SortBy: Λίστα με την ενεργοποίηση της οποίας ο χρήστης επιλέγει το τρόπο με τον οποίο θα εμφανίζονται οι χώροι στάθμευσης στο χάρτη (ψηφινότερος-κοντινότερος).
- ❑ MyPosition: Κουμπί με την ενεργοποίηση του οποίου εμφανίζεται στο χάρτη η τρέχουσα θέση του χρήστη.
- ❑ Map: Ο διαδραστικός χάρτης που εμφανίζεται στην οθόνη του χρήστη.
- ❑ Delete: Κουμπί με την επιλογή του οποίου ο χρήστης μπορεί να διαγράψει τον λογαριασμό του απ το σύστημα.

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ refresh(): Συνάρτηση η οποία καλείται με το πάτημα του κουμπιού logo η οποία είναι υπεύθυνη να ανανεώσει την αρχική σελίδα που βρίσκεται ο χρήστης.
- ❑ makeBook(desPos,userId): Συνάρτηση η οποία δέχεται ως όρισμα την επιθυμητή θέση του χρήστη καθώς επίσης και το userId, και πραγματοποιεί μια κράτηση θέσης αφού πρώτα επικοινωνήσει με την αντίστοιχη βάση Δεδομένων των χώρων στάθμευσης.
- ❑ sortBy(type): Συνάρτηση η οποία ταξινομεί τους χώρους στάθμευσης που εμφανίζονται στον χρήστη ανάλογα με τον τύπο που έχει επιλέξει ο ίδιος μέσω από ένα μενού επιλογών το οποίο εμφανίζεται επίσης από την συγκεκριμένη συνάρτηση στην οθόνη του χρήστη.
- ❑ getCost(userId): Συνάρτηση η οποία είναι υπεύθυνη να εμφανίσει μέσω του userID ορίσματος της την τρέχουσα χρέωση του χρήστη. Η συνάρτηση επικοινωνά με τη βάση δεδομένων και αφού πάρει τα απαραίτητα στοιχεία εκτελεί τους υπολογισμούς που χρειάζονται.
- ❑ GoogleMaps(LatLong): Συνάρτηση η οποία αναλαμβάνει να επικοινωνήσει με το GoogleMaps API. Συγκεκριμένα κάνει κλήση των υπηρεσιών (interfaces) της ανάλογα με το συγκεκριμένες επιλογές του χρήστη. Με το πάτημα του κουμπιού Myposition η συνάρτηση καλεί την μέθοδο *GoogleMap.OnMyLocationButtonClickListener* η οποία είναι υπεύθυνη για τον εντοπισμό της τρέχουσα θέσης του χρήστη, με το πάτημα του κουμπιού drivemethere καλείται η μέθοδος *GoogleMap.OnMapClickListener* η οποία είναι υπεύθυνη να εντοπίσει την επιθυμητή θέση προορισμού την οποία έχει επιλέξει ο χρήστης από την οθόνη του και μέσω της μεθόδου *locationsource*, η οποία εμφανίζει πληροφορίες της επιθυμητής θέσης, εμφανίζει στον χρήστη την διαδρομή που πρέπει να ακολουθήσει. Τέλος με κλήση της μεθόδου *LocationSource.OnLocationChangedListener* πραγματοποιείται η συνεχής ενημέρωση της τρέχουσας θέσης του χρήστη.
- ❑ delete(userId): Η μέθοδος αυτή επικοινωνεί με τη Βάση Δεδομένων του συστήματος και διαγράφει τον λογαριασμό του χρήστη userId.



### Boundary Class Create Account GUI



Σχήμα 8: Διάγραμμα Κλάσης Create Account GUI

Τα χαρακτηριστικά της κλάσης αυτής είναι:

- ❑ **Logo:** Το logo του ThessPark με το πάτημα του οποίου ο χρήστης θα μεταβαίνει στην αρχική σελίδα της εφαρμογής.
- ❑ **Create Account:** Πεδίο εμφάνισης κειμένου όπου εμφανίζεται στο χρήστη για να τον ενημερώσει ότι βρίσκεται στο πεδίο δημιουργίας λογαριασμού.
- ❑ **Username:** Πεδίο καταχώρησης κειμένου όπου ο χρήστης καταγράφει το username(ψευδώνυμο) που επιθυμεί να έχει στο λογαριασμό του. Για λόγους ασφαλείας κάθε username πρέπει να είναι μοναδικό.
- ❑ **Password:** Πεδίο καταχώρησης κειμένου όπου ο χρήστης καταγράφει το κωδικό που επιθυμεί να έχει στο λογαριασμό του. Για λόγους ασφαλείας ο κωδικός εμφανίζεται με τη μορφή αστερίσκου (\*) έτσι ώστε να μην είναι ορατό σε άλλα πρόσωπα.
- ❑ **A.D.T.:** Πεδίο εισαγωγής του Α.Δ.Τ. Απαιτείται μόνο από τους ιδιοκτήτες χώρου στάθμευσης. Για τους απλούς χρήστες είναι κενό.
- ❑ **Apply:** Κουμπί με την ενεργοποίηση του οποίου καταχωρείται η αίτηση του χρήστη για δημιουργία λογαριασμού στο σύστημα.
- ❑ **Reset:** Κουμπί με την ενεργοποίηση του οποίου καθαρίζονται όλα τα πεδία καταχώρησης κειμένου στη σελίδα (username & password).
- ❑ **TermsOfUse:** Κουμπί με την ενεργοποίηση του οποίου εμφανίζονται στο χρήστη σε μορφή μηνύματος όλοι οι κανονισμοί και προϋποθέσεις για την δημιουργία λογαριασμού στο σύστημα.

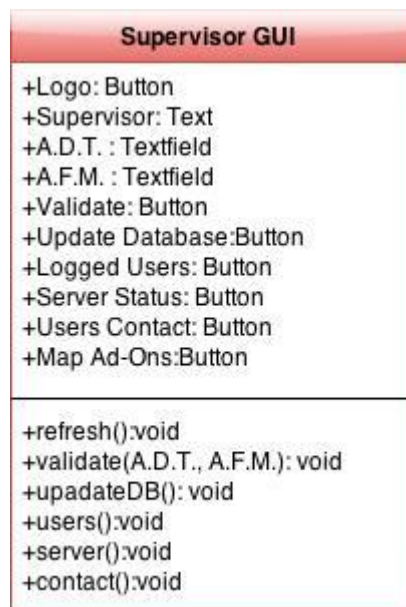
Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ **refresh():** Συνάρτηση η οποία καλείται με το πάτημα του κουμπιού logo η οποία είναι υπεύθυνη να ανανεώσει την αρχική σελίδα που βρίσκεται ο χρήστης.



- ❑ create(username,password, A.D.T.): Συνάρτηση η οποία επικοινωνεί με τον κατάλληλο controller ώστε να δημιουργηθεί στη Βάση Δεδομένων νέος λογαριασμός με στοιχεία username και password τα οποία έχει εισάγει ο χρήστης. Εάν έχει εισαχθεί κάποιος αριθμός στην μεταβλητή A.D.T. τότε ο χρήστης καταχωρείται ως parking administrator. Καλείται με το πάτημα του κουμπιού Apply.
- ❑ reset(): Η συνάρτηση αυτή καλείται με το πάτημα του κουμπιού Reset και απλώς καθαρίζει όλα τα textfield από τις μέχρι τώρα καταχωρήσεις του χρήστη.
- ❑ termsOfUse(): Συνάρτηση η οποία καλείται με το πάτημα του κουμπιού termofuse και είναι υπεύθυνη να εμφανίσει στο χρήστη σε μορφή λίστας κειμένου όλους τους όρους χρήσης της εφαρμογής μας.

### Boundary Class Supervisor GUI



Σχήμα 9: Διάγραμμα Κλάσης Supervisor GUI

Τα χαρακτηριστικά της κλάσης αυτής είναι:

- ❑ Logo: Το logo του ThessPark με το πάτημα του οποίου ο χρήστης θα μεταβαίνει στην αρχική σελίδα της εφαρμογής.
- ❑ Supervisor: Πεδίο εμφάνισης χαρακτήρων για ενημέρωση του supervisor σε ποίο τμήμα της εφαρμογής βρίσκεται.

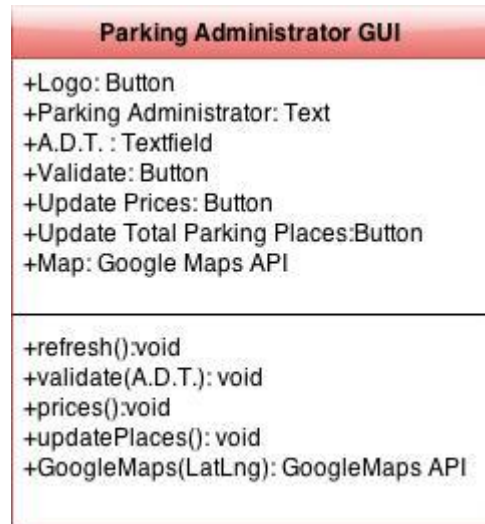




- ❑ A.D.T: Πεδίο καταχώρησης χαρακτήρων στο οποίο ο supervisor, καλείται να καταχωρήσει τον αριθμό του δελτίου ταυτότητας του.
- ❑ A.F.M: Πεδίο καταχώρησης χαρακτήρων στο οποίο ο supervisor, καλείται να συμπληρώσει το αριθμό της φορολογικής του δήλωσης.
- ❑ Validate: Κουμπί με την ενεργοποίηση του οποίου, εάν ο supervisor έχει συμπληρώσει σωστά τα απαραίτητα στοιχεία που χρειάζονται, μεταφέρει τον supervisor στο μενού επιλογών που διαθέτει.
- ❑ Update Database: Κουμπί με την ενεργοποίηση του οποίου ο supervisor, μεταφέρεται σε καινούργιο μενού έχοντας τη δυνατότητα να ανανεώσει οτιδήποτε θέλει στη βάση Δεδομένων του συστήματος.
- ❑ Logged Users: Κουμπί με την ενεργοποίηση του οποίου ο supervisor, βλέπει ιστορικό με τη κίνηση των χρηστών μέσα στο σύστημα για διάστημα 2 εβδομάδων.
- ❑ Server Status: Κουμπί με την ενεργοποίηση του οποίου ο supervisor, βλέπει ιστορικό με την κατάσταση του server, για καλύτερη αντιμετώπιση τυχών προβλημάτων ή βελτίωση του για αποφυγή μελλοντικών προβλημάτων.
- ❑ Users Contact: Κουμπί με την ενεργοποίηση του οποίου, εμφανίζεται μια λίστα στην οποία υπάρχουν τα στοιχεία επικοινωνίας των διάφορων χώρων στάθμευσης για τυχών διευθετήσεων που πρέπει να γίνουν μεταξύ των δύο.
- ❑ Map Ad-ons: Κουμπί με την ενεργοποίηση του οποίου ο supervisor, μπορεί να τροποποιήσει το τι εμφανίζεται στο διαδραστικό χάρτη,

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ refresh(): Συνάρτηση η οποία καλείται με το πάτημα του κουμπιού logo η οποία είναι υπεύθυνη να ανανεώσει την αρχική σελίδα που βρίσκεται ο χρήστης.
- ❑ validate(A.D.T,A.F.M): Συνάρτηση η οποία δέχεται σαν όρισμα την ταυτότητα του supervisor και τον αριθμό του φορολογικού του μητρώου και αφού τον επικυρώσει τον οδηγεί στο αντίστοιχο μενού επιλογών.
- ❑ updateDB(): Συνάρτηση η οποία είναι υπεύθυνη να διαικπεραιώσει τις διάφορες εργασίες του συστήματος όσον αφορά την αναβάθμιση των βάσεων Δεδομένων του. Ενεργοποιείται με το πάτημα του αντίστοιχου κουμπιού.
- ❑ users(): Συνάρτηση η οποία είναι υπεύθυνη να εμφανίσει στον χρήστη, το ιστορικό κινήσεως των χρηστών μέσα στο σύστημα, αφού ενεργοποιηθεί απο τον supervisor το κουμπί loggedUSers.
- ❑ server(): Συνάρτηση η οποία ενεργοποιείται με το πάτημα του κουμπιού serverstatus και είναι υπεύθυνη να εμφανίσει στην οθόνη του χρήστη το ιστορικό του server.
- ❑ contact(): Συνάρτηση η οποία ενεργοποιείται με το πάτημα του κουμπιού User Contact και είναι υπεύθυνη να εμφανίσει στην οθόνη του χρήστη ένα κατάλογο με τα στοιχεία επικοινωνίας όλων των χώρων στάθμευσης του συστήματος.

**Boundary Class ParkingAdmin GUI**

Σχήμα 10: Διάγραμμα Κλάσης Parking Administator GUI

Τα χαρακτηριστικά της κλάσης αυτής είναι:

- ❑ Logo: Το logo του ThessPark με το πάτημα του οποίου ο χρήστης θα μεταβαίνει στην αρχική σελίδα της εφαρμογής.
- ❑ ParkingAdministator: Πεδίο εμφάνισης χαρακτήρων για ενημέρωση του ParkingAdmin σε ποίο τμήμα της εφαρμογής βρίσκεται.
- ❑ A.D.T: Πεδίο καταχώρησης χαρακτήρων στο οποίο ο ParkingAdmin, καλείται να καταχωρήσει τον αριθμό του δελτίου ταυτότητας του.
- ❑ Validate: Κουμπί με την ενεργοποίηση του οποίου, εάν ο ParkingAdmin έχει συμπληρώσει σωστά τα απαραίτητα στοιχεία που χρειάζονται, μεταφέρει τον ParkingAdmin στο μενού επιλογών που διαθέτει.
- ❑ Update Prices: Κουμπί με την ενεργοποίηση του οποίου δίνεται η δυνατότητα στον ParkingAdmin να ανανεώσει τις χρεώσεις που έχει ο χώρος στάθμευσης του.
- ❑ UpdateTotalParkingPlaces: Κουμπί με την ενεργοποίηση του οποίου δίνεται η δυνατότητα στον ParkingAdmin να ανανεώσει τις συνολικά διαθέσιμες θέσεις που έχει ο χώρος στάθμευσης του.
- ❑ Map: Ο διαδραστικός χάρτης που εμφανίζεται στην οθόνη του ParkingAdmin, για τυχόν έλεγχο πληροφοριών διπλανών γειτονικών χώρων και έτσι πιθανώς καλλιέργεια υγιούς ανταγωνισμού

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ refresh(): Συνάρτηση η οποία καλείται με το πάτημα του κουμπιού logo η οποία είναι υπεύθυνη να ανανεώσει την αρχική σελίδα που βρίσκεται ο χρήστης.
- ❑ validate(A.D.T): Συνάρτηση η οποία δέχεται σαν όρισμα την ταυτότητα του parkingadmin και αφού τον επικυρώσει τον οδηγεί στο αντίστοιχο μενού επιλογών.



- ❑ prices(): Συνάρτηση η οποία ενεργοποιείτε με το πάτημα του κουμπιού updateprices και είναι υπεύθυνη να ενημερώσει τις τιμές χρέωσης στην βάση Δεδομένων των χώρων στάθμευσης.
- ❑ updatePlaces(): Συνάρτηση η οποία είναι υπεύθυνη να ενημερώσει τις διαθέσιμες χώρου στάθμευσης του εκάστοτε χώρου. Ενεργοποιείται με το πάτημα του κουμπιού updateTotalParkingPlaces.
- ❑ googleMaps(LatLong): Συνάρτηση η οποία είναι υπεύθυνη καλέσει την μέθοδο LocationSource της εφαρμογής googlemaps η οποία βάση την θέση του χώρου στάθμευσης θα εμφανίζει στο χάρτη του parkingadmin όλους τους γειτονικούς χώρους στάθμευσης.

## 2.2 Πακέτο Utilities

Τα συγκεκριμένα πακέτα αναφέρονται στα Utilities, δηλαδή στις διάφορες εργασίες (ουσιαστικά στο ποιές συναρτήσεις πρέπει να καλέσει), που πρέπει να κάνει το σύστημα ανάλογα με τις επιλογές του εκάστοτε χρήστη, Parking Admin και Supervisor, για να πραγματοποιήσει. Ως εκ τούτου, τα αντικείμενα των κλάσεων που ανήκουν στο εν λόγω πακέτο – αναπαριστούν μόνιμη πληροφορία η οποία ανιχνεύεται από το σύστημα (<<entities>>). Συγκεκριμένα, για κάθε κατηγορία του συστήματος Utilities, έχουμε και μια ξεχωριστή κλάση.

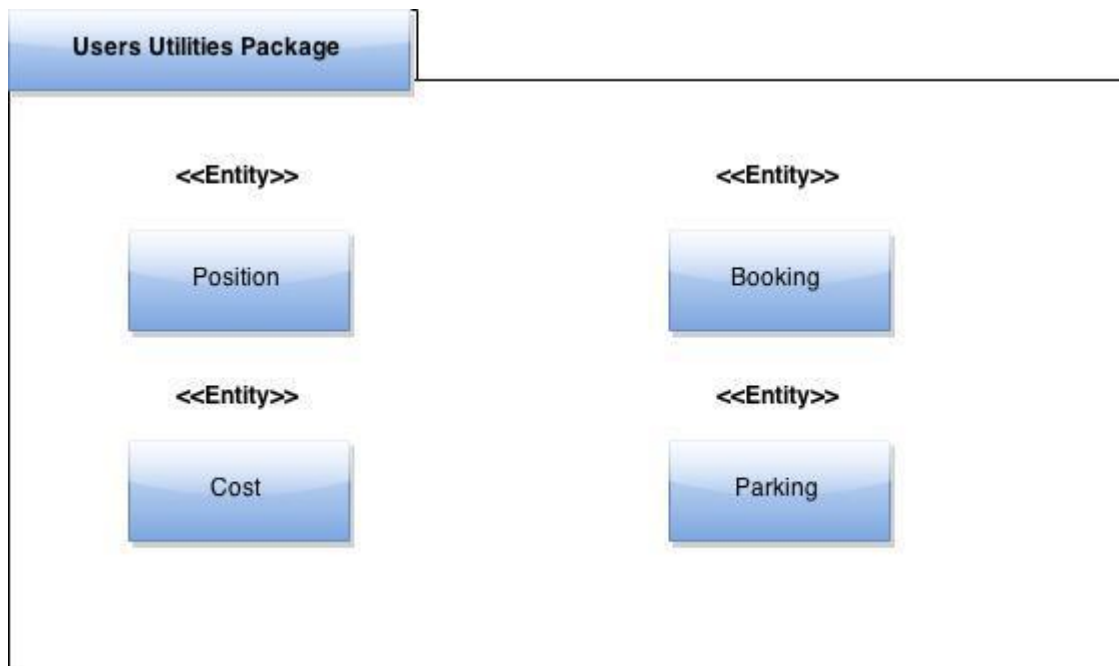


Σχήμα 11: Διάγραμμα Πακέτου Utilities



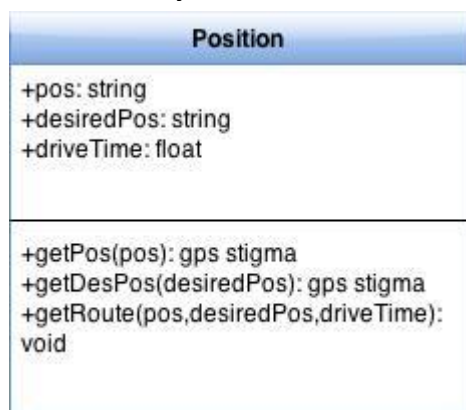
### 2.2.1 Πακέτο User Utilities

Πακέτο που αφορά αποκλειστικά τα Utilities, που έχουν να κάνουν με τον απλό χρήστη του συστήματος. Μετά από εκτενή ανάλυση του πακέτου αυτού προέκυψαν οι εξής κλάσεις οντοτήτων: position, booking, cost και Parking.



Σχήμα 12: Διάγραμμα Πακέτου Users Utilities

#### Entity Class Position



Σχήμα 13: Διάγραμμα Κλάσης Position



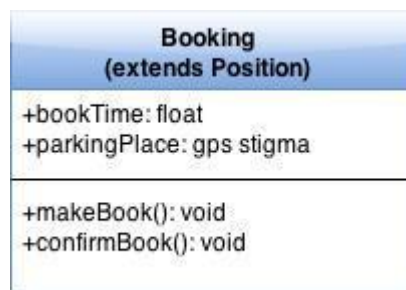
Τα χαρακτηριστικά της κλάσης αυτής είναι:

- ❑ pos: Μεταβλητή τύπου String η οποία επιστρέφει ένα στίγμα gps, το οποίο είναι η τρέχουσα θέση του χρήστη. (Ο όρος gps stigma αναφέρεται στις συντεταγμένες “γεωγραφικό μήκος και πλάτος”)
- ❑ desiredPos: Μεταβλητή τύπου String η οποία επιστρέφει ένα στίγμα gps, το οποίο είναι η επιθυμητή θέση που θέλει να πάει ο χρήστης.
- ❑ driveTime: Μεταβλητή τύπου float. Είναι ο χρόνος ο οποίος που χρειάζεται ο χρήστης, για να μεταβεί από την θέση στην οποία βρίσκεται στην επιθυμητή του.

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ getPos(pos): Συνάρτηση η οποία αναλαμβάνει να βρει την τρέχουσα θέση του χρήστη σε συνεργασία με το googleMaps. Επιστρέφει ένα gps stigma στο διαδραστικό χάρτη στην οθόνη του χρήστη.
- ❑ getDesPos(desiredPos): Συνάρτηση η οποία αναλαμβάνει να βρει την επιθυμητή θέση που θέλει να πάει ο χρήστης σε συνεργασία με το googleMaps. Επιστρέφει ένα gps stigma στο διαδραστικό χάρτη στην οθόνη του χρήστη.
- ❑ getRoute(pos,desiredPos,driveTime): Συνάρτηση η οποία είναι υπεύθυνη, να εμφανίσει στο διαδραστικό χάρτη την βέλτιστη διαδρομή από την τρέχουσα θέση του χρήστη στην επιθυμητή του, σύμφωνα με τη διαδρομή που θα υπολογίσει κάνοντας κλήση της μεθόδου *GoogleMap.Route(Location LatLang)* . Επίσης υπολογίζει προσεγγιστικά και επιστρέφει στο χρήστη τον εκτιμώμενο χρόνο που θα χρειαστεί μέσω της μεταβλητής driveTime.

### Entity Class Booking



Σχήμα 14: Διάγραμμα Κλάσης Booking

Τα χαρακτηριστικά της κλάσης αυτής είναι :

- ❑ bookTime: Μεταβλητή η οποία παίρνει ως τιμή την χρονική στιγμή κατά την οποία πραγματοποιείται η κράτηση απ τον χρήστη.
- ❑ parkingPlace: Μεταβλητή η οποία παίρνει σαν τιμή το στίγμα gps του χώρου στάθμευσης στο οποίο γίνεται η κράτηση.



Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ makeBook(userId,desiredPos): Η συνάρτηση η οποία αναλαμβάνει να διεκπεραιώσει την κράτηση του χρήστη στο σύστημα. Ουσιαστικά ελέγχει στη βάση δεδομένων τη διαθεσιμότητα του συγκεκριμένου χώρου στάθμευσης και ανάλογα προβαίνει στην κράτηση ή όχι της θέσης.
- ❑ confirmBook(bookTime,userId): Η συνάρτηση η οποία ενημερώνει την μεταβλητή bookTime με την τρέχουσα χρονική στιγμή κατά την οποία πραγματοποιείται η κράτηση και εμφανίζει μήνυμα στο χρήστη σχετικά με την κατάσταση της κράτησής του (επιτυχής ή όχι).

### Entity Class Cost



Σχήμα 15: Διάγραμμα Κλάσης Cost

Τα χαρακτηριστικά της κλάσης αυτής είναι :

- ❑ currentTime: Μεταβλητή τύπου float. Είναι η χρονική στιγμή βάση της οποίας θα υπολογισθεί η τρέχουσα χρέωση του χρήστη.

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ getTime(): Συνάρτηση η οποία υπολογίζει την τρέχουσα χρονική στιγμή βάση της οποίας θα χρεωθεί ο χρήστης λαμβάνοντας υπ' όψην τη χρονική διαφορά μεταξύ αυτής και της bookTime δηλαδή του χρόνου που έγινε η κράτηση και την οποία κληρονομεί από την κλάση Booking.
- ❑ getCost(): Συνάρτηση η οποία υπολογίζει και επιστρέφει το κόστος χρέωσης, βάση της διάρκειας του χρόνου στάθμευσης στην οποία καλείται να πληρώσει ο χρήστης.



### Entity Class Parking



Σχήμα 16: Διάγραμμα Κλάσης Parking

Τα χαρακτηριστικά της κλάσης αυτής είναι :

- ❑ available: Μεταβλητή τύπου Integer. Είναι τα συνολικά διαθέσιμα parkings κάθε χώρου στάθμευσης.
- ❑ type: Μεταβλητή τύπου String. Είναι η κατηγορία (Ιδιωτικό ή Δημόσιο) στην οποία ανήκει ο χώρος στάθμευσης.
- ❑ parkPos: Μεταβλητή τύπου gps stigma. Είναι το longitude και latitude που χαρακτηρίζει την θέση του χώρου στάθμευσης στο χάρτη.

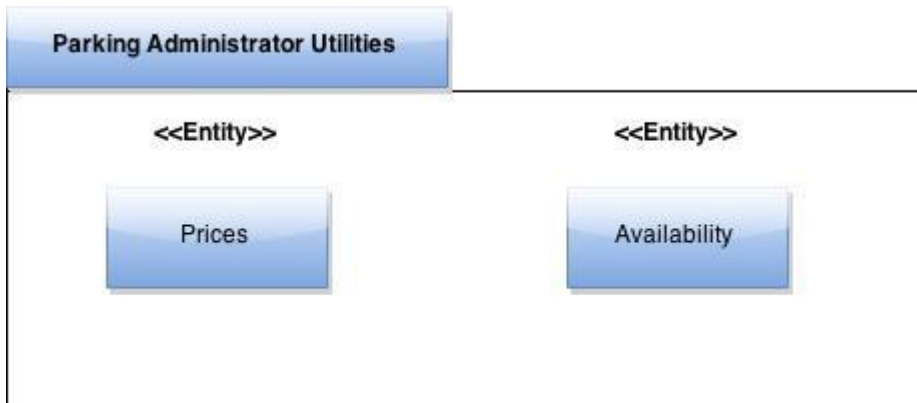
Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ getAvailable(): Συνάρτηση η οποία είναι υπεύθυνη για την συνεχή ενημέρωση από την βάση δεδομένων, της μεταβλητής available, καθώς και την εμφάνιση του αριθμού αυτού στο διαδραστικό χάρτη για κάθε χώρο στάθμευσης.
- ❑ getParkPos(): Συνάρτηση η οποία υπολογίζει για κάθε καινούργιο χώρο στάθμευσης, το gps stigma στο οποίο βρίσκεται και το αποθηκεύει στη Βάση Δεδομένων. Επίσης ενημερώνει την μεταβλητή parkPos για κάθε χώρο ξεχωριστά.



## 2.2.2 Πακέτο ParkingAdmin Utilities

Πακέτο που αφορά αποκλειστικά τα Utilities, που έχουν να κάνουν με τον parkingAdmin του συστήματος. Μετά από εκτενή ανάλυση του πακέτου αυτού προέκυψαν οι εξής κλάσεις οντοτήτων: prices και availability.



Σχήμα 17: Διάγραμμα Πακέτου Parking Administrator Utilities



Σχήμα 18: Διάγραμμα Κλάσης Prices

Τα χαρακτηριστικά της κλάσης αυτής είναι :

- ❑ priceperHour: Μεταβλητή τύπου float. Είναι η τιμή που αντιστοιχεί κάθε ώρα στάθμευσης.

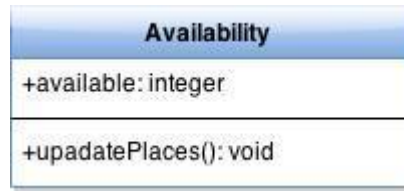
Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ setPrices(): Συνάρτηση η οποία είναι υπεύθυνη για την ενημέρωση των τιμών χρέωσης του χώρου, της βάσεως δεδομένων του συστήματος, καθώς και για την εμφάνιση του αριθμού αυτού στο διαδραστικό χάρτη για κάθε χώρο στάθμευσης.





### Entity Class Availability



Σχήμα 19: Διάγραμμα Κλάσης Availability

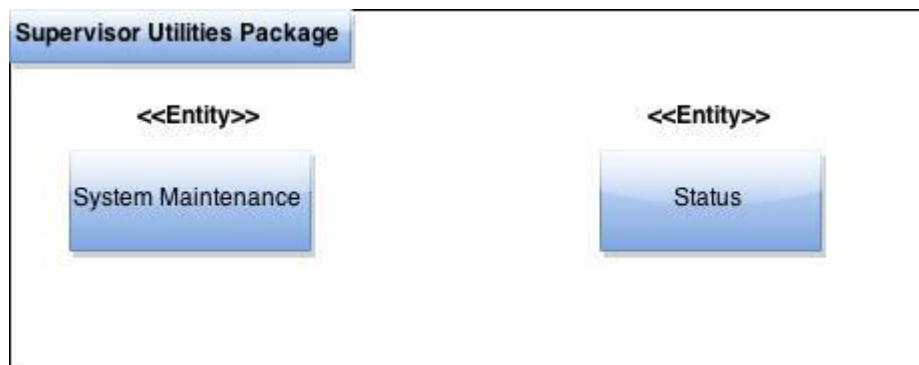
Τα χαρακτηριστικά της κλάσης αυτής είναι :

- ❑ available: Μεταβλητή τύπου integer . Είναι η τιμή που αντιστοιχεί στα ελεύθερα parkings κάθε χώρου στάθμευσης.

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ updatePlaces(): Συνάρτηση η οποία είναι υπεύθυνη για την ενημέρωση των ελεύθερων parkings του χώρου, της βάσεως δεδομένων του συστήματος.

### 2.2.3 Πακέτο Supervisor Utilities

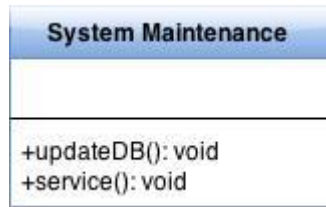


Σχήμα 20: Διάγραμμα Πακέτου Supervisor Utilities Package

Πακέτο που αφορά αποκλειστικά τα Utilities, που έχουν να κάνουν με τον Supervisor του συστήματος. Μετά από εκτενή ανάλυση του πακέτου αυτού προέκυψαν οι εξής κλάσεις οντοτήτων: SystemMaintenance, DBmaintenance και Status.



### Entity Class SystemMaintenance

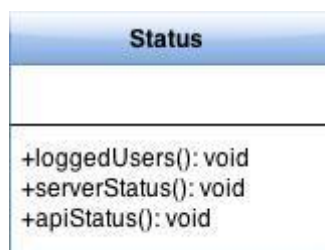


Σχήμα 21: Διάγραμμα Κλάσης System Maintenance

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ service(): Συνάρτηση η οποία είναι υπεύθυνη για την επαναφορά του συστήματος σε περίπτωση σφάλματος, αστοχίας δικτύου κλπ. Συγκεκριμένα, μέσω αυτής ο supervisor μπορεί να επαναφέρει τον server σε κατάσταση λειτουργίας σε περίπτωση “κατάρρευσης” αυτού. Σε κάθε περίπτωση είναι η αρμόδια συνάρτηση για επικοινωνία με τους controllers οι οποίοι θα υλοποιήσουν τις αλλαγές του χρήστη.
- ❑ updateDB(): Μέσω αυτής της συνάρτησης ο supervisor μπορεί να ανανεώσει τις Βάσεις Δεδομένων του συστήματος. Η συνάρτηση επικοινωνεί με τον controller updateDBcontrol.

### Entity Class Status



Σχήμα 22: Διάγραμμα Κλάσης Status

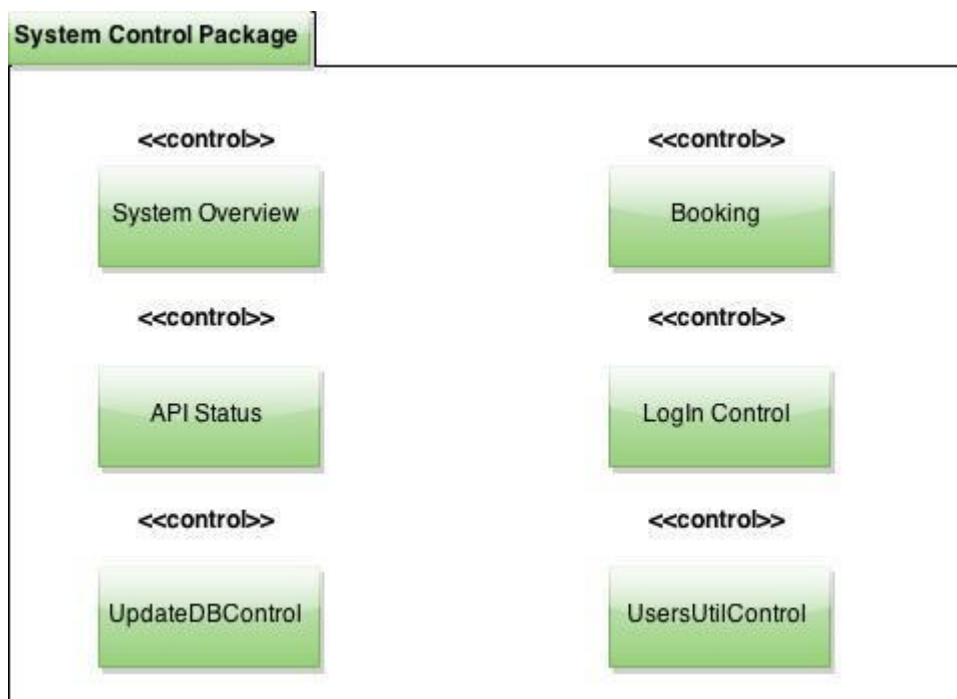
Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ loggedUsers(): Συνάρτηση η οποία είναι υπεύθυνη να επιστρέψει στον supervisor το ιστορικό των online users της εφαρμογής για διάστημα 2 εβδομάδων.
- ❑ serverStatus(): Συνάρτηση η οποία είναι υπεύθυνη να επιστρέψει στον supervisor το ιστορικό του server της εφαρμογής.
- ❑ apiStatus(): Συνάρτηση η οποία είναι υπεύθυνη να επιστρέψει στον supervisor το ιστορικό για την επικοινωνία μεταξύ της εφαρμογής μαζί με την εφαρμογή του googleMaps.



### 2.3 Πακέτο SystemControl

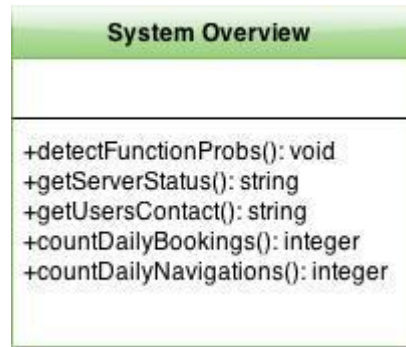
Πακέτο το οποίο αναφέρεται σε αντικείμενα ελέγχου του συστήματος, αφού εδώ ουσιαστικά έχουμε να κάνουμε τον έλεγχο (control) του συστήματός μας. Συγκεκριμένα, περιλαμβάνει μια σειρά από κλάσεις που η κάθε μια συμβάλλει διακριτά στην εύρυθμη λειτουργία του συστήματος. Στο πακέτο αυτό υπάρχει μια κλάση με το όνομα “SystemOverview” που μας δίνει γενικές πληροφορίες για την τρέχουσα κατάσταση του συστήματος αλλά και στατιστικά για την χρήση του το τελευταίο διάστημα. Επίσης, υπάρχει η κλάση “Booking”, η οποία με τη σειρά της είναι υπεύθυνη στο να ρυθμίζει, σε περίπτωση που εκτελούνται πολλές κρατήσεις μαζί, ποιες από αυτές θα έχουν μεγαλύτερη βαρύτητα, ώστε να τους παρέχεται προτεραιότητα ως προς την εκτέλεσή τους. Τέλος, υπάρχει και η κλάση “API status”, η οποία ελέγχει ανά πάσα στιγμή τη συνεργασία του συστήματος με εξωτερικές διεπαφές, συγκεκριμένα με την εφαρμογή του googlemaps η οποία είναι απαραίτητη για τις διάφορες λειτουργίες που αφορούν το χρήστη. Μια οπτικοποίηση του πακέτου και των κλάσεων που περιλαμβάνει παρουσιάζονται παρακάτω



Σχήμα 23: Διάγραμμα Πακέτου System Control



### SystemOverview Controller Class



Σχήμα 24: Διάγραμμα Κλάσης System Overview

Οι μέθοδοι της κλάσης αυτής είναι:

- ❑ detectFunctionProbs(): Συνάρτηση η οποία είναι υπεύθυνη να καταγράφει όλα τα προβλήματα που παρουσιάζονται στον server και εμποδίζουν την εύρυθμη λειτουργία του συστήματος.
- ❑ getServerStatus(): Συνάρτηση η οποία είναι υπεύθυνη να καταγράφει όλα τα στοιχεία του server (κίνηση, fly hours, last service), και με εντολή από τον supervisor επιστρέφει όλα αυτά τα στοιχεία σε μορφή ιστορικού.
- ❑ getUsersContact(): Συνάρτηση η οποία είναι υπεύθυνη να ενημερώνει μηνιαίως όλα τα στοιχεία επικοινωνίας κάθε χώρου στάθμευσης, καθώς και μετά από αίτημα να τα εμφανίζει στην οθόνη του supervisor σε μορφή λίστας.
- ❑ countDailyBookings(): Συνάρτηση η οποία είναι υπεύθυνη να καταγράφει τις καθημερινές κρατήσεις που γίνονται από το σύστημα και να τις αποθηκεύει για ευρύτερη χρήση.
- ❑ countDailyNavigations(): Συνάρτηση η οποία είναι υπεύθυνη να καταγράφει τις καθημερινές πλοηγήσεις που γίνονται από το σύστημα και να τις αποθηκεύει για ευρύτερη χρήση.

### Booking Controller Class



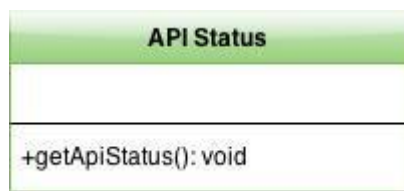
Σχήμα 25: Διάγραμμα Κλάσης Booking



Οι μέθοδοι αυτής της κλάσης είναι:

- ❑ getBookingStatus(): Συνάρτηση η οποία διαχειρίζεται την εκάστοτε κράτηση που είναι υπό επεξεργασία και ενημερώνει το σύστημα για την επιτυχία της. Επίσης είναι υπεύθυνη για την ομαλή κατανομή των κρατήσεων σε περίπτωση πολλαπλών αιτήσεων.
- ❑ checkTimeLimit(): Συνάρτηση η οποία παρακολουθεί όλες τις τρέχουσες κρατήσεις και ανιχνεύει τις περιπτώσεις που ο χρήστης ξεπέρασε το χρονικό όριο κράτησης όπως αυτό έχει οριστεί στο έγγραφο Απαιτήσεων Χρηστών.
- ❑ createBooking(): Συνάρτηση η οποία είναι υπεύθυνη για την δημιουργία μιας κράτησης θέσης.

### API Controller Class



Σχήμα 26: Διάγραμμα Κλάσης API Status

Η μέθοδος της κλάσης αυτής είναι :

- ❑ getApiStatus(): Συνάρτηση η οποία ενημερώνει το σύστημα για την κατάσταση της συνεργαζόμενης με αυτό διεπαφής, δηλαδή το Google Maps. Η σημασία της κλάσης αυτής είναι πολύ σημαντική καθώς η όλη εφαρμογή βασίζεται στο διαδραστικό χάρτη με τον οποίο ο χρήστης έχει τη δυνατότητα εποπτείας όλων των διαθέσιμων επιλογών του.

### UserUtilControl Class



Σχήμα 27: Διάγραμμα Κλάσης UserUtilControl



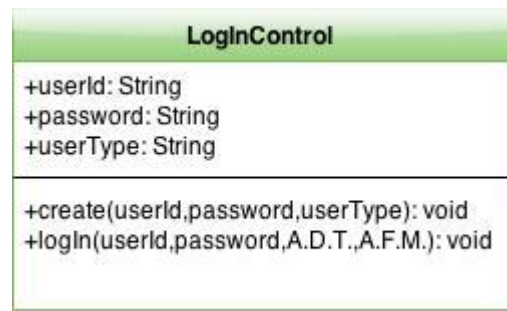
Οι μεταβλητές της κλάσης είναι :

- ☐ parkPos: Μεταβλητή η οποία περιέχει τη θέση του χώρου στάθμευσης.
- ☐ userPos: Μεταβλητή η οποία περιέχει τη θέση του χρήστη.

Οι μέθοδοι της κλάσης είναι :

- ☐ getParkPos(): Μέθοδος η οποία επιστρέφει την θέση του χώρου στάθμευσης.
- ☐ getUserPos(): Μέθοδος η οποία επιστρέφει την τρέχουσα θέση του χρήστη.
- ☐ getClosestParks(): Μέθοδος η οποία επιστρέφει τον κοντινότερο χώρο στάθμευσης με βάση την τοποθεσία του χρήστη.
- ☐ getRoute(): Μέθοδος η οποία επιστρέφει την διαδρομή που πρέπει να ακολουθήσει ο χρήστης.

### LoginControl Class



Σχήμα 28: Διάγραμμα Κλάσης LoginControl

Οι μεταβλητές της κλάσης είναι :

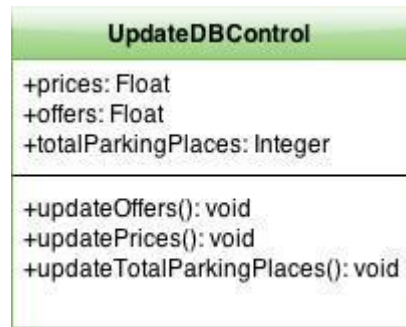
- ☐ userId: Μεταβλητή η οποία περιέχει την ταυτότητα του χρήστη.
- ☐ password: Μεταβλητή η οποία περιέχει τον κωδικό πρόσβασης του χρήστη.
- ☐ userType: Μεταβλητή η οποία περιέχει τον τύπο του χρήστη (απλός, parking admin, supervisor)

Οι μέθοδοι της κλάσης είναι :

- ☐ create(userId,password,userType): Μέθοδος η οποία δημιουργεί ένα χρήστη ανάλογα με τα στοιχεία που εισάγει.
- ☐ login(userId,password,A.D.T.,A.F.M.): Μέθοδος η οποία εκτελεί την είσοδο του χρήστη. Οι μεταβλητές A.D.T. και A.F.M. χρησιμοποιούνται μόνο στην περίπτωση που πρόκειται για είσοδο ενός parking administrator ή supervisor.



### UpdateDBcontrol Class



Σχήμα 29: Διάγραμμα Κλάσης UpdateDBControl

Οι μεταβλητές της κλάσης είναι :

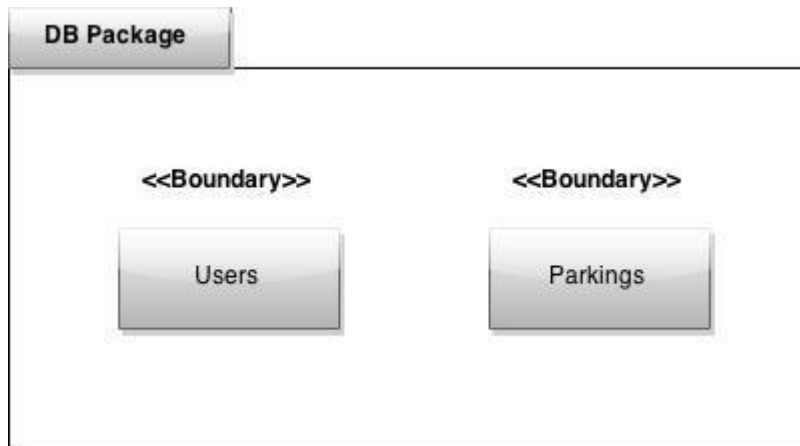
- ☐ prices: Μεταβλητή η οποία περιέχει τις χρεώσεις του χώρου στάθμευσης.
- ☐ offers: Μεταβλητή η οποία περιέχει τις προσφορές του χώρου στάθμευσης.
- ☐ totalParkingPlaces: Μεταβλητή η οποία περιέχει το συνολικό αριθμό των διαθέσιμων χώρων στάθμευσης.

Οι μέθοδοι της κλάσης είναι :

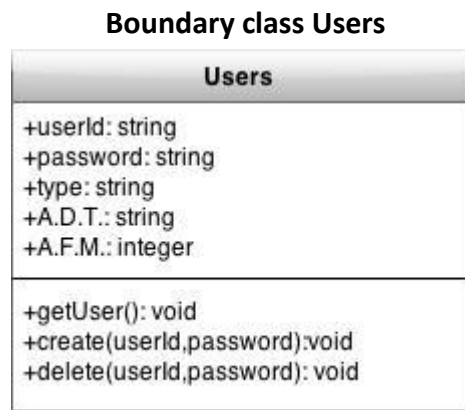
- ☐ updateOffers(): Μέθοδος η οποία είναι υπεύνη να ανανεώσει τις προσφορές του χώρου στάθμευσης.
- ☐ updatePrices(): Μέθοδος η οποία είναι υπεύνη να ανανεώσει τις τιμές του χώρου στάθμευσης.
- ☐ updateTotalParkingPlaces(): Μέθοδος η οποία είναι υπεύνη να ανανεώσει τις συνολικά διαθέσιμες θέσεις του χώρου στάθμευσης.

## 2.4 Πακέτο DataBase

Πακέτο το οποίο αναφέρεται στη βάση δεδομένων του συστήματος και περιγράφει τις κλάσεις απ τις οποίες αποτελείται. Συγκεκριμένα το πακέτο εμπεριέχει τις οριακές κλάσεις Users και Parkings. Η πρώτη έχει όλα τα στοιχεία τα οποία χρειάζονται για την ταυτοποίηση ενός χρήστη (είτε είναι απλός χρήστης είτε parking administrator είτε supervisor) κατά την διαδικασία εισαγωγής του στο σύστημα (log-in process), σε αυτήν αποθηκεύονται οι αλλαγές οι οποίες μπορεί να προκύψουν για κάποιο χρήστη καθώς και κάθε νέα εγγραφή χρήστη στο σύστημα. Η κλάση Parkings υλοποιεί όλα τα χαρακτηριστικά τα οποία καθορίζουν ένα χώρο στάθμευσης ο οποίος συμμετέχει στην εφαρμογή και επιτρέπει την ανανέωσή τους απ τους εκάστωτε αρμόδιους. Παρακάτω αναλύονται λεπτομερώς τα περιεχόμενα αυτών των κλάσεων.



Σχήμα 30: Διάγραμμα Πακέτου DB



Σχήμα 31: Διάγραμμα Κλάσης Users

Τα χαρακτηριστικά της κλάσης αυτής είναι :

- ☐ userId: Μεταβλητή τύπου String . Είναι το όνομα που έχει επιλέξει ο κάθε χρήστης.
- ☐ password: Μεταβλητή τύπου Integer. Είναι ο κωδικός που έχει επιλέξει ο κάθε χρήστης.
- ☐ type: Μεταβλητή τύπου String. Είναι καταγεγραμμένος ο τύπος κάθε χρήστη.
- ☐ A.D.T: Μεταβλητή τύπου String. Είναι ο αριθμός του δελτίου ταυτότητας . Αφορά μόνο τους χρήστες ParkingAdmin και Supervisor.
- ☐ A.F.M: Μεταβλητή τύπου Integer. Είναι ο αριθμός της φορολογικής δήλωσης . Αφορά μόνο τον χρήστη Supervisor.

Οι μέθοδοι της κλάσης αυτής είναι:

- ☐ getUser(): Συνάρτηση η οποία είναι υπεύθυνη για την ενημέρωση των χαρακτηριστικών της κλάσης και για την επιστροφή των στοιχείων της μετά από αίτημα του συστήματος.





- ❑ create(userId,password): Συνάρτηση η οποία καταχωρεί στη Βάση Χρηστών τον χρήστη με στοιχεία userId και password.
- ❑ delete(userId,password): Συνάρτηση η οποία διαγράφει από τη Βάση Χρηστών τον χρήστη με στοιχεία userId και password.

### Boundary class Parkings



Σχήμα 32: Διάγραμμα Κλάσης Parkings

Τα χαρακτηριστικά αυτής της κλάσης είναι:

- ❑ type: Μεταβλητή η οποία περιέχει τον τύπο του εκάστοτε parking (δημόσιο ή ιδιωτικό).
- ❑ location: Μεταβλητή η οποία περιέχει το στίγμα gps του εκάστοτε parking ώστε αυτό να εμφανίζεται σωστά στο διαδραστικό χάρτη.
- ❑ charges: Μεταβλητή η οποία περιέχει τις χρεώσεις τις οποίες ορίζει ο parking administrator σύμφωνα με την αρεσκεία του.
- ❑ offers: Μεταβλητή η οποία περιέχει τις προσφορές τις οποίες μπορεί να θέλει να κάνει ο parking administrator σε μορφή κειμένου και αυτές θα εμφανίζονται στον διαδραστικό χάρτη.
- ❑ totalParkingPlaces: Μεταβλητή η οποία περιέχει τον συνολικό αριθμό χώρων στάθμευσης για το εκάστοτε parking.
- ❑ availablePlaces: Μεταβλητή η οποία περιέχει τον τρέχον αριθμό ελεύθερων χώρων στάθμευσης του εκάστοτε parking. Η μεταβλητή αυτή ανανεώνεται με κάθε προσθήκη/ αποχώρηση οχήματος από το parking.

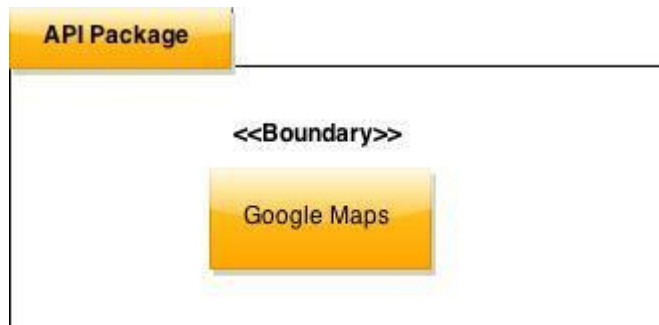
Η μέθοδος της κλάσης αυτής είναι :

- ❑ getParking(): Συνάρτηση η οποία είναι υπεύθυνη για την ενημέρωση των χαρακτηριστικών της κλάσης και για την επιστροφή των στοιχείων της μετά από αίτημα του συστήματος.



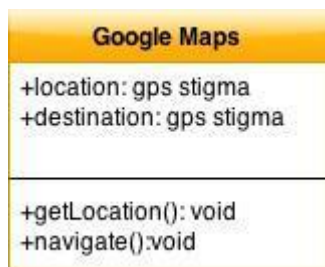
## 2.5 Πακέτο API

Το τελευταίο πακέτο προς ανάπτυξη αφορά τη διεπαφή του κυρίου συστήματος με εξωτερικές εφαρμογές. Στο σύστημα μας η βασική και μόνη εξωτερική εφαρμογή είναι η εφαρμογή του googleMaps. Και σε αυτή τη περίπτωση θα αναφερθούμε σε οριακά αντικείμενα (<>). Η κλάση που απαρτίζει το πακέτο είναι η GoogleMaps. Το τι πραγματεύεται είναι εμφανές από το όνομά της, αλλά θα αναλυθεί διεξοδικά στη συνέχεια. Κάτωθι, παρουσιάζεται το πακέτο στο σύνολό του, μαζί με την κλάση που το απαρτίζει.



Σχήμα 33: Διάγραμμα Πακέτου API

### Boundary class GoogleMaps



Σχήμα 34: Διάγραμμα Κλάσης GoogleMaps

Τα χαρακτηριστικά της κλάσης είναι :

- ❑ location: Κατά την εγκατάσταση της εφαρμογής ThessPark στο κινητό τηλέφωνο του χρήστη η διεπαφή GoogleMaps αιτείται της αδειοδότησης σε αυτήν εκ μέρους του χρήστη όπως αυτή να μπορεί να χρησιμοποιεί το στίγμα gps της συσκευής για να μπορεί να καθορίζει τη θέση του χρήστη στο διαδραστικό χάρτη. Η μεταβλητή αυτή περιέχει το προαναφερθέν στίγμα.
- ❑ destination: Μεταβλητή η οποία περιέχει το στίγμα του χώρου στάθμευσης προς πλοήγηση που επιλέγει ο χρήστης με το πάτημά του πάνω στον διαδραστικό χάρτη.

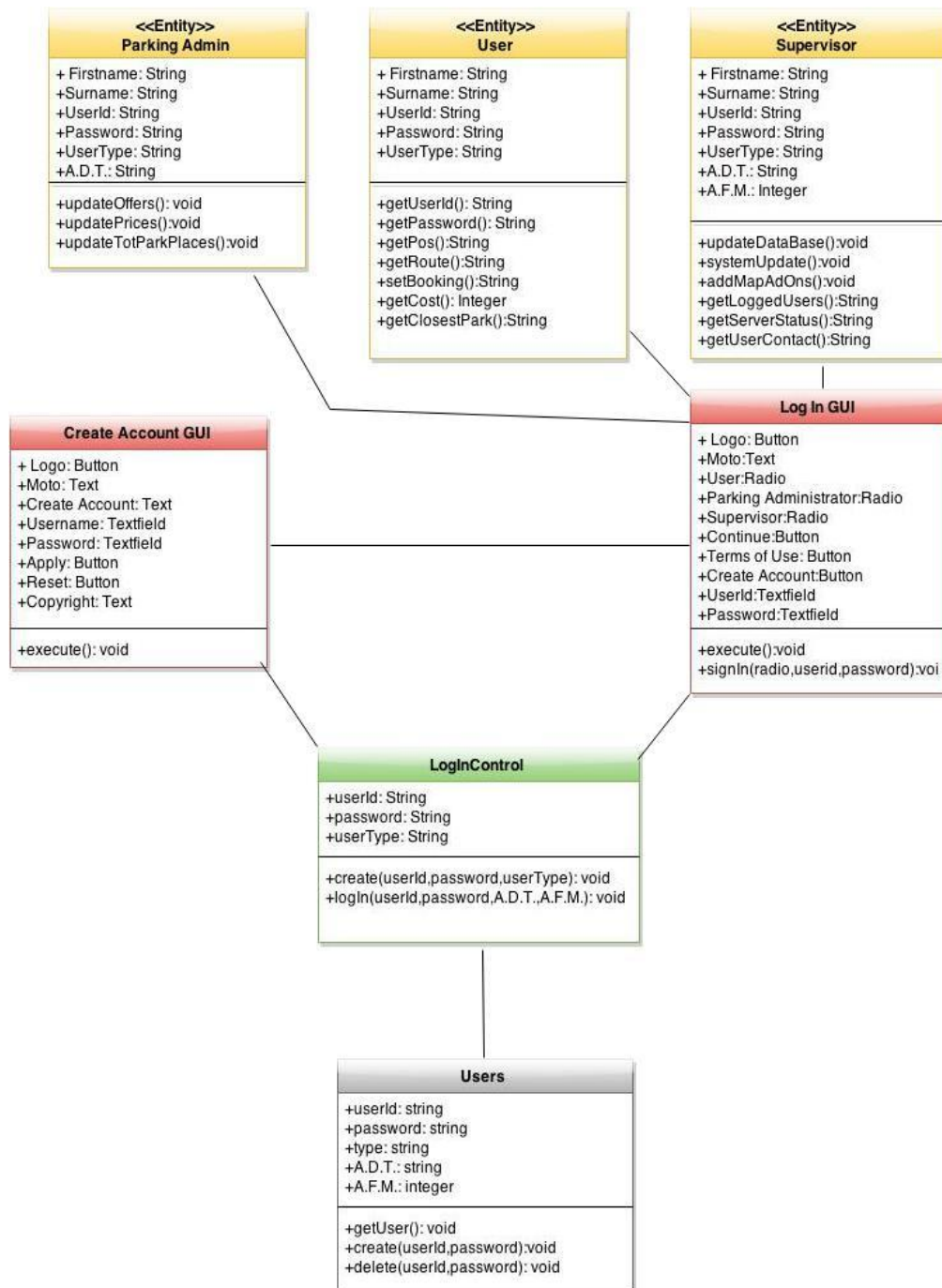


Οι μέθοδοι της κλάσης αυτής είναι:

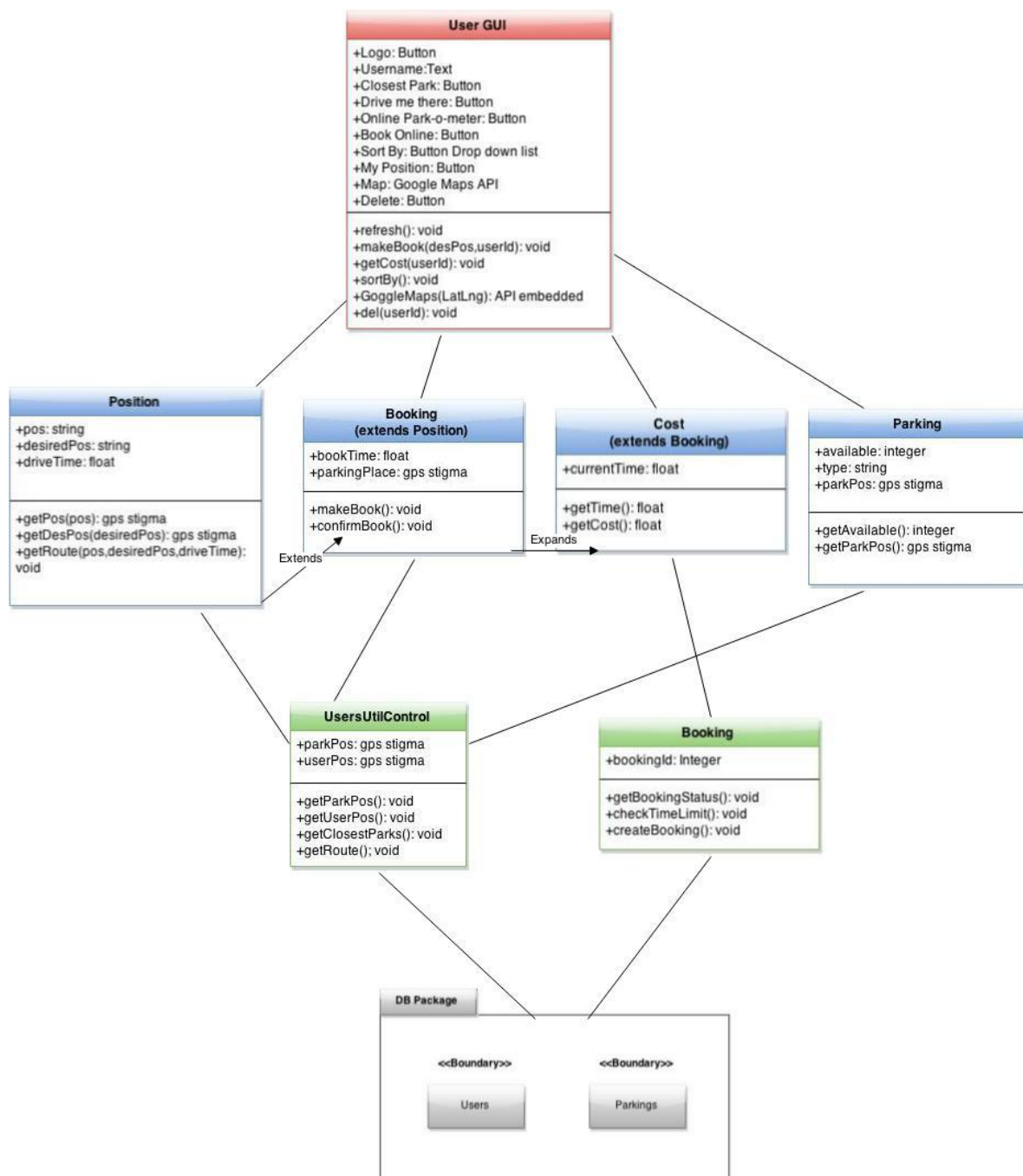
- ❑ getLocation(): Συνάρτηση η οποία επικοινωνεί με τον πομποδέκτη gps της συσκευής και επιστρέφει το στίγμα αυτής.
- ❑ navigate(): Συνάρτηση η οποία δέχεται ως ορίσματα τις μεταβλητές location και destination και αναλαμβάνει να πλοηγήσει τον χρήστη στον επιθυμητό χώρο στάθμευσης εμφανίζοντας την διαδρομή στον διαδραστικό χάρτη.



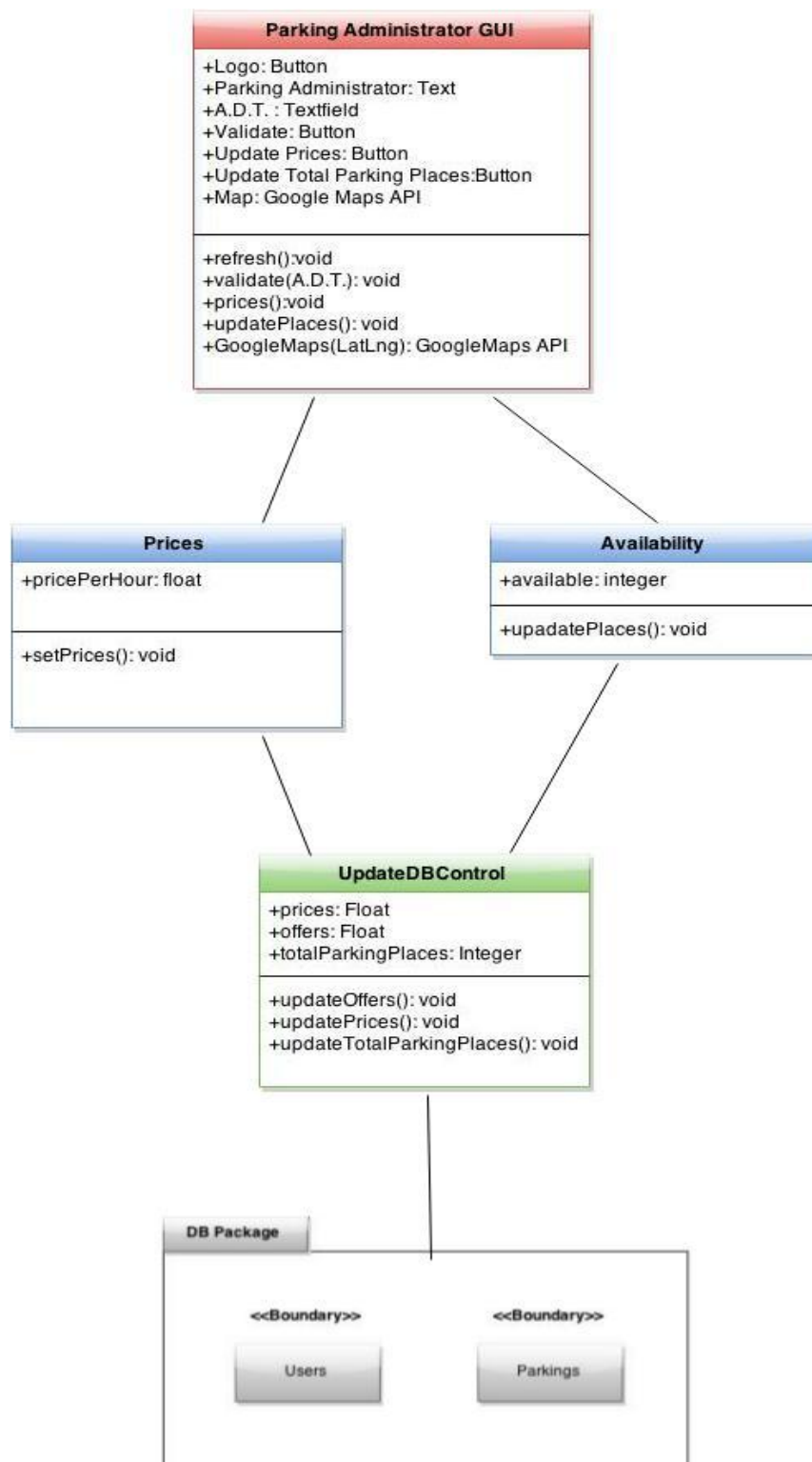
## 2.6 Διαγράμματα κλάσεων



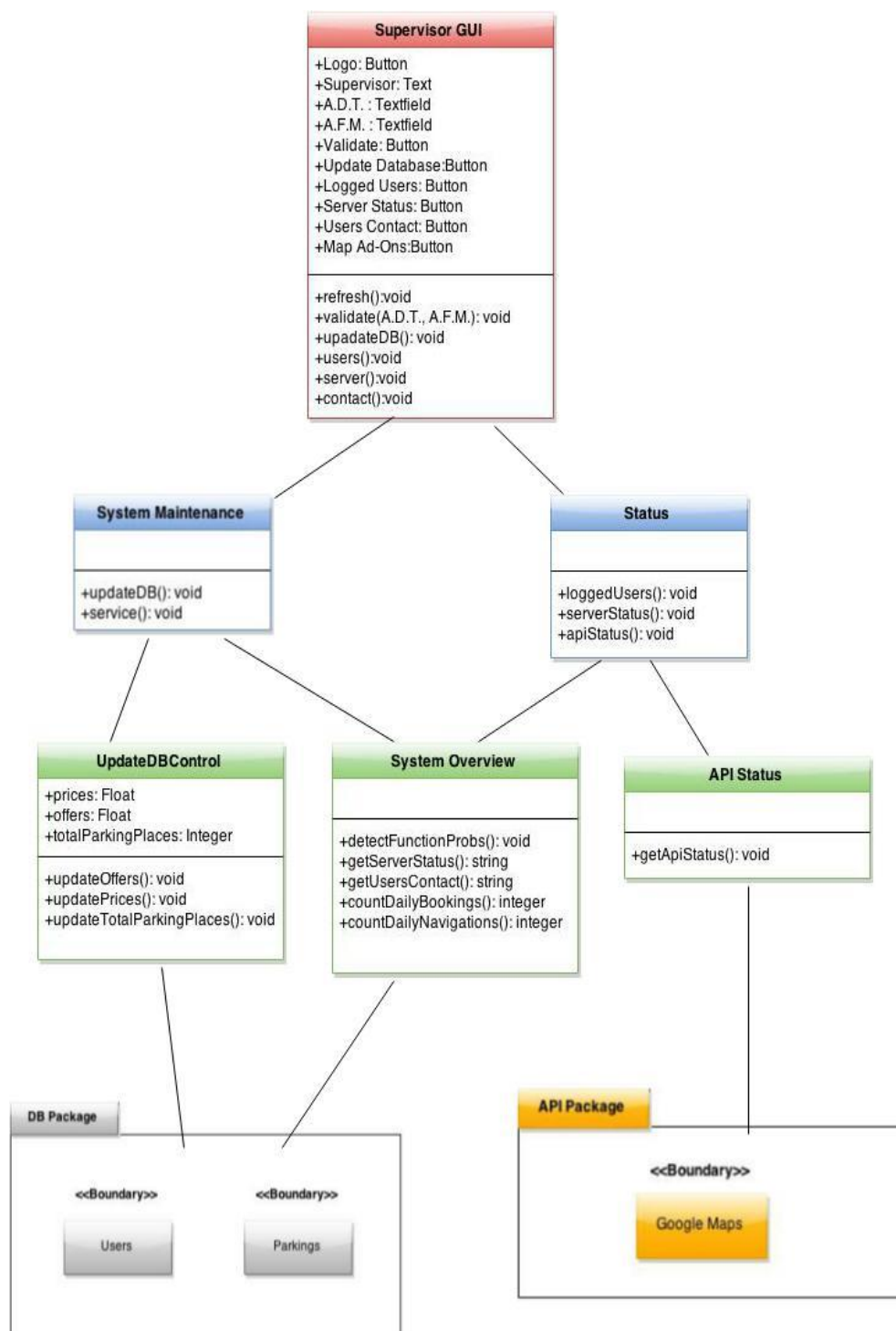
Σχήμα 35: Διάγραμμα Κλάσεων



Σχήμα 36: Διάγραμμα Κλάσεων Πακέτου Βασικής Διεπαφής Απλού Χρήστη



Σχήμα 37: Διάγραμμα Κλάσεων Πακέτου διεπαφής Διαχειριστή Χώρου Στάθμευσης



Σχήμα 38: Διάγραμμα Κλάσεων Πακέτου διεπαφής Διαχειριστή Συστήματος



## 3 Απαιτήσεις Λογισμικού

### 3.1 Δυναμική μοντελοποίηση του συστήματος

#### 3.1.1 Πακέτο Αρχική Οθόνη και Είσοδος χρήστη.

Το πακέτο αυτό αφορά την πρώτη επαφή του χρήστη με την εφαρμογή και την είσοδό του στο σύστημα. Αρχικά με το πρώτο άνοιγμα της εφαρμογής εμφανίζεται το παράθυρο επιλογής δημιουργίας λογαριασμού ή εισόδου χρήστη. Μετά από τη δημιουργία λογαριασμού ή εισόδου του χρήστη στο σύστημα εμφανίζεται η βασική διεπαφή της εφαρμογής.

Οι λειτουργικές απαιτήσεις που αντιστοιχίζονται σε αυτό το πακέτο αφήγησης είναι:

- <ΛΑ-1> Ο διαχειριστής του συστήματος πρέπει να μπορεί να κάνει LOGIN στο σύστημα.
- <ΛΑ-2> Ο διαχειριστής ιδιόκτητου parking πρέπει να μπορεί να κάνει LOGIN στο σύστημα.
- <ΛΑ-4> Ο απλός χρήστης πρέπει να μπορεί να κάνει LOGIN στο σύστημα.

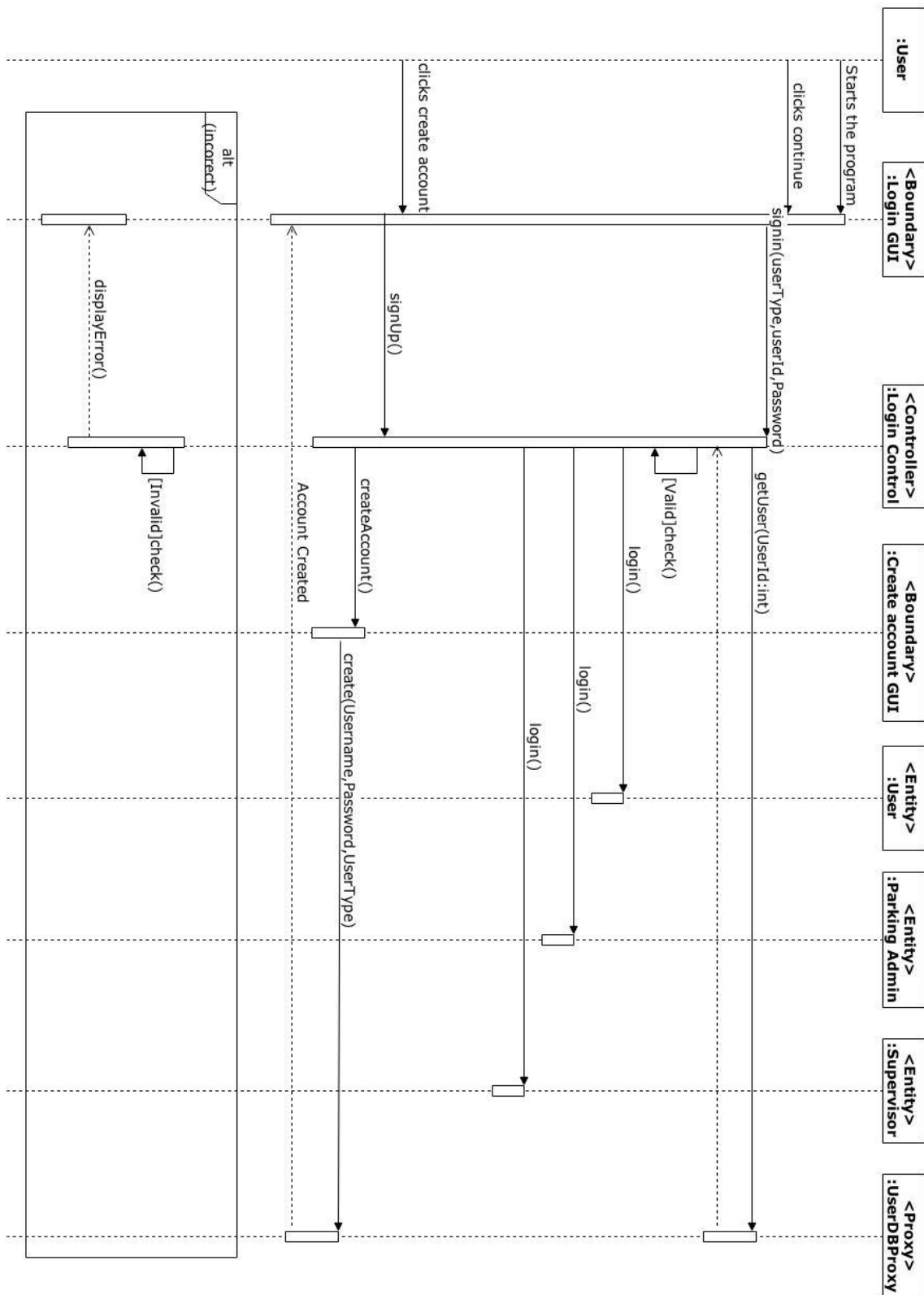
##### 3.1.1.1 Αφήγηση Σεναρίου:

Ο χρήστης ανοίγει την εφαρμογή. Αρχικά, του εμφανίζεται το γραφικό περιβάλλον του πακέτου Log In GUI. Ο χρήστης επιλέγει την ιδιότητα του (Supervisor, Parking Admin, απλός) ή Create Account. Μετά από την εισαγωγή των επιλογών από το χρήστη, με το πάτημα του κουμπιού continue εκτελείται η `signin(UserType, userID, Password)`.

Αν ο χρήστης επέλεξε ιδιότητα (Supervisor, Parking Admin, απλός) και καταχώρησε username και password τότε ο `LoginControl` ελέγχει τα στοιχεία που εισήχθησαν καλώντας την `check()` με την εξωτερική βάση δεδομένων `UserDBProxy`. Αν τα στοιχεία είναι σωστά δημιουργείται η οντότητα του χρήστη (Supervisor, Parking Admin, απλός) μέσω της συνάρτησης `login()` και εμφανίζεται η βασική διεπαφή Supervisor GUI, Parking Admin GUI, User GUI ανάλογα με την επιλογή. Σε περίπτωση εισαγωγής λανθασμένων στοιχείων ο `LoginControl()` επιστρέφει σφάλμα και το Log In GUI αναλαμβάνει να εμφανίσει μήνυμα λάθους καλώντας την `displayError()`.

Σε περίπτωση που ο χρήστης πατήσει το κουμπί Create Account τότε εκτελείται η `signup()` και ο `LoginControl` μέσω της `create()` εμφανίζει την διεπαφή Create Account GUI όπου ο χρήστης έχει δυνατότητα δημιουργίας λογαριασμού. Τέλος ο λογαριασμός δημιουργείται μέσω της συνάρτησης `create(Username, Password, UserType)` και η βάση δεδομένων ενημερώνεται.





Σχήμα 39: Διάγραμμα ακολουθιών προβολής αρχικής οθόνης – εισόδου χρήστη.



### 3.1.2 Πακέτο Αναβάθμισης Βάσης Δεδομένων Parkings.

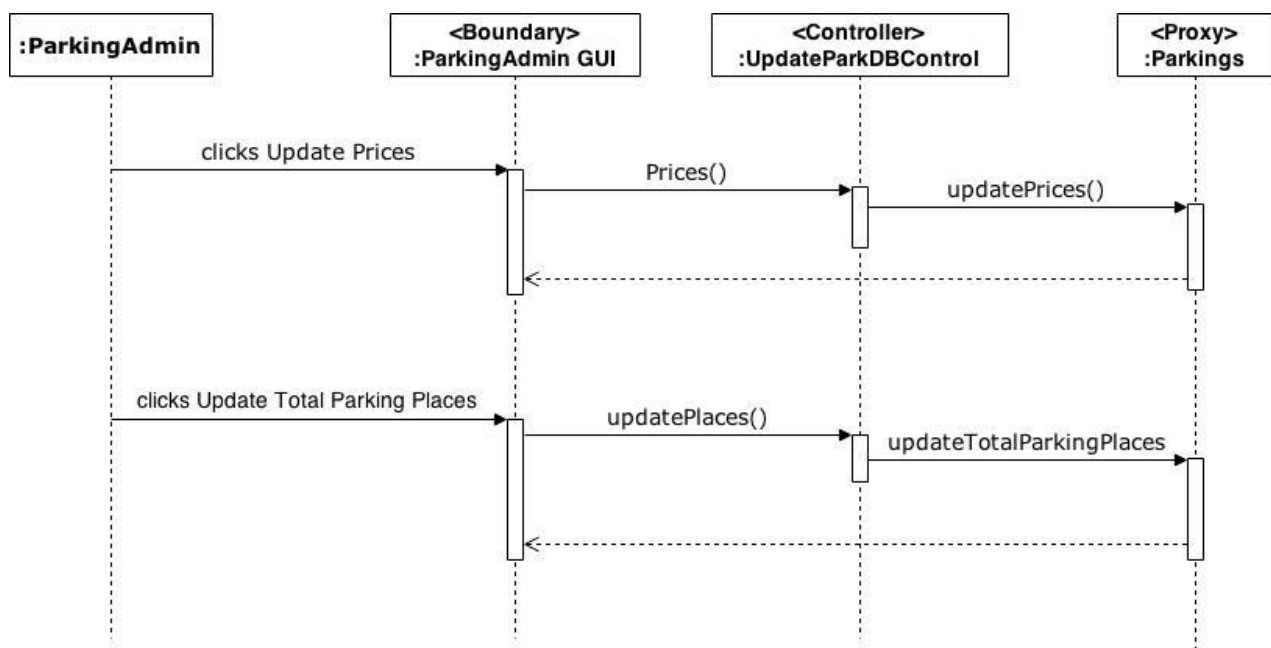
Το πακέτο αυτό αφορά την αναβάθμιση της βάσης δεδομένων των χώρων στάθμευσης από τον χρήστη ParkingAdmin. Συγκεκριμένα μπορούν να αναβαθμιστούν οι τιμές, οι συνολικοί χώροι στάθμευσης και οι προσφορές.

Οι λειτουργικές απαιτήσεις που αντιστοιχίζονται σε αυτό το πακέτο αφήγησης είναι:

- **<ΛΑ-3>** Ο διαχειριστής ιδιόκτητου parking πρέπει να μπορεί να ανανεώνει τη βάση δεδομένων που αντιστοιχεί στο parking του.

#### 3.1.2.1 Αφήγηση Σεναρίου:

Ο ParkingAdmin μετά την επιτυχή είσοδο του στο σύστημα μεταφέρεται στο γραφικό περιβάλλον Parking Admin GUI. Επιλέγοντας το κουμπί UpdatePrices ή UpdateTotalParkingPlaces εκτελείται η Prices() ή η UpdatePlaces() και ο UpdateParkDBControl ανανεώνει ανάλογα τη βάση δεδομένων Parkings.



Σχήμα 40:Διάγραμμα ακολουθιών Αναβάθμισης Βάσης Δεδομένων Parkings



### 3.1.3 Πακέτο Λειτουργιών Απλού Χρήστη.

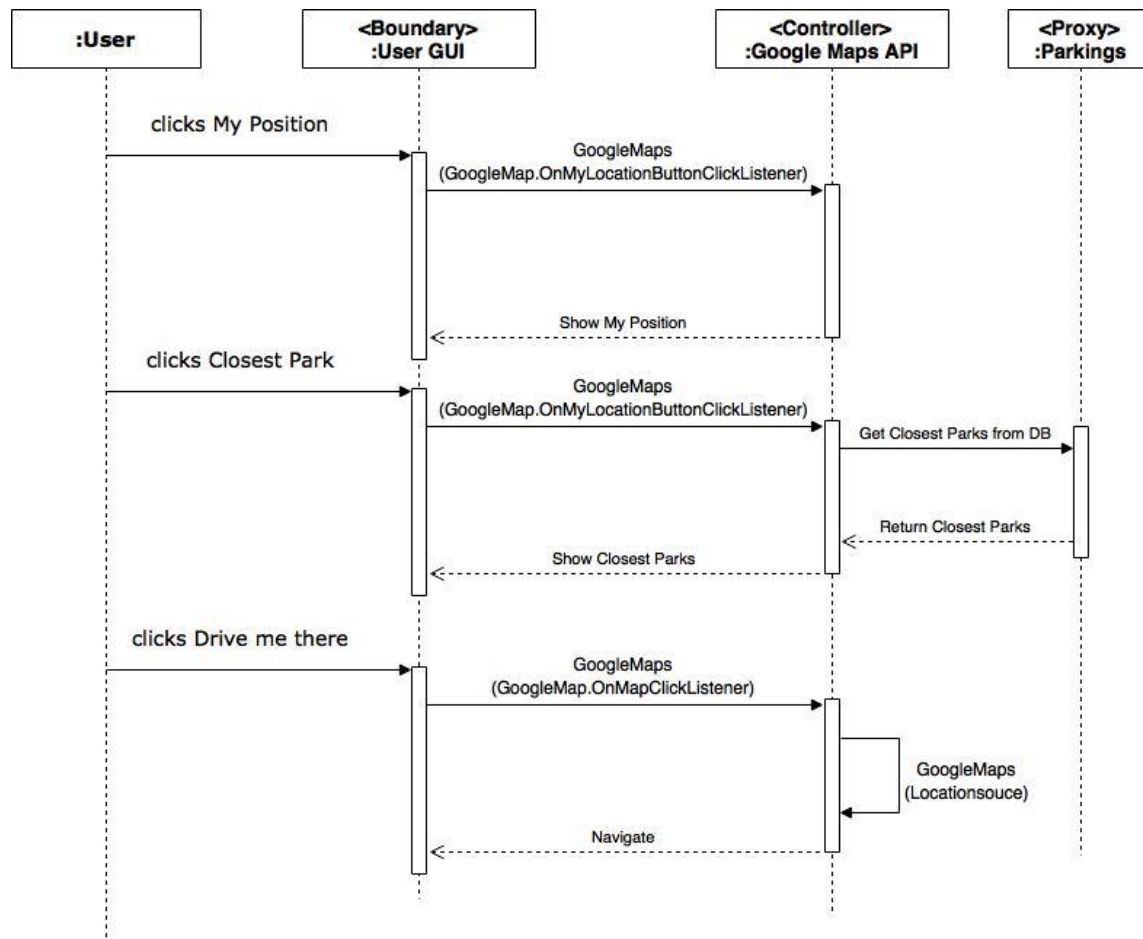
Το πακέτο αυτό αφορά όλες τις λειτουργίες που μπορεί να εκτελέσει ο απλός χρήστης, δηλαδή να βλέπει όλους τους διαθέσιμους χώρους στάθμευσης, να επιλέγει τον προορισμό του και να πλοηγείται σε αυτόν μέσω του διαδραστικού χάρτη.

Οι λειτουργικές απαιτήσεις που αντιστοιχίζονται σε αυτό το πακέτο αφήγησης είναι:

- <ΛΑ-5> Ο απλός χρήστης πρέπει να μπορεί να βλέπει όλα τα διαθέσιμα parkings.
- <ΛΑ-6> Ο απλός χρήστης πρέπει να μπορεί να βλέπει τη θέση του στο χάρτη.
- <ΛΑ-7> Ο απλός χρήστης πρέπει να μπορεί να προσδιορίσει τον προορισμό του.
- <ΛΑ-8> Ο απλός χρήστης πρέπει να μπορεί να προσδιορίσει την κατηγορία του parking.
- <ΛΑ-10> Ο απλός χρήστης πρέπει να μπορεί να πλοηγηθεί στον προορισμό του.

#### 3.1.3.1 Αφήγηση Σεναρίου:

Ο User μέσω του User GUI επιλέγει το κουμπί My Position. Η τοποθεσία του χρήστη φαίνεται στο χάρτη στο User GUI αφού πρώτα έχει βρεθεί μέσω του controller Google Maps API. Ο User επιλέγει το κουμπί Closest Park. Ο Google Maps API παίρνει της συντεταγμένες των κοντινότερων χώρων στάθμευσης από τη θέση του χρήστη από τη βάσης δεδομένων Parkings και έπειτα τους εμφανίζει στο χάρτη με τα στοιχεία τους. Μετά ο χρήστης επιλέγει το χώρο στάθμευσης που επιθυμεί και με την επιλογή του κουμπιού Drive me There ο Google Maps API Controller πλοηγεί το χρήστη στο τελικό του προορισμό μέσω του χάρτη.



Σχήμα 41: Διάγραμμα ακολουθιών Λειτουργιών Απλού χρήστη



### 3.1.4 Πακέτο Κράτησης Θέσης.

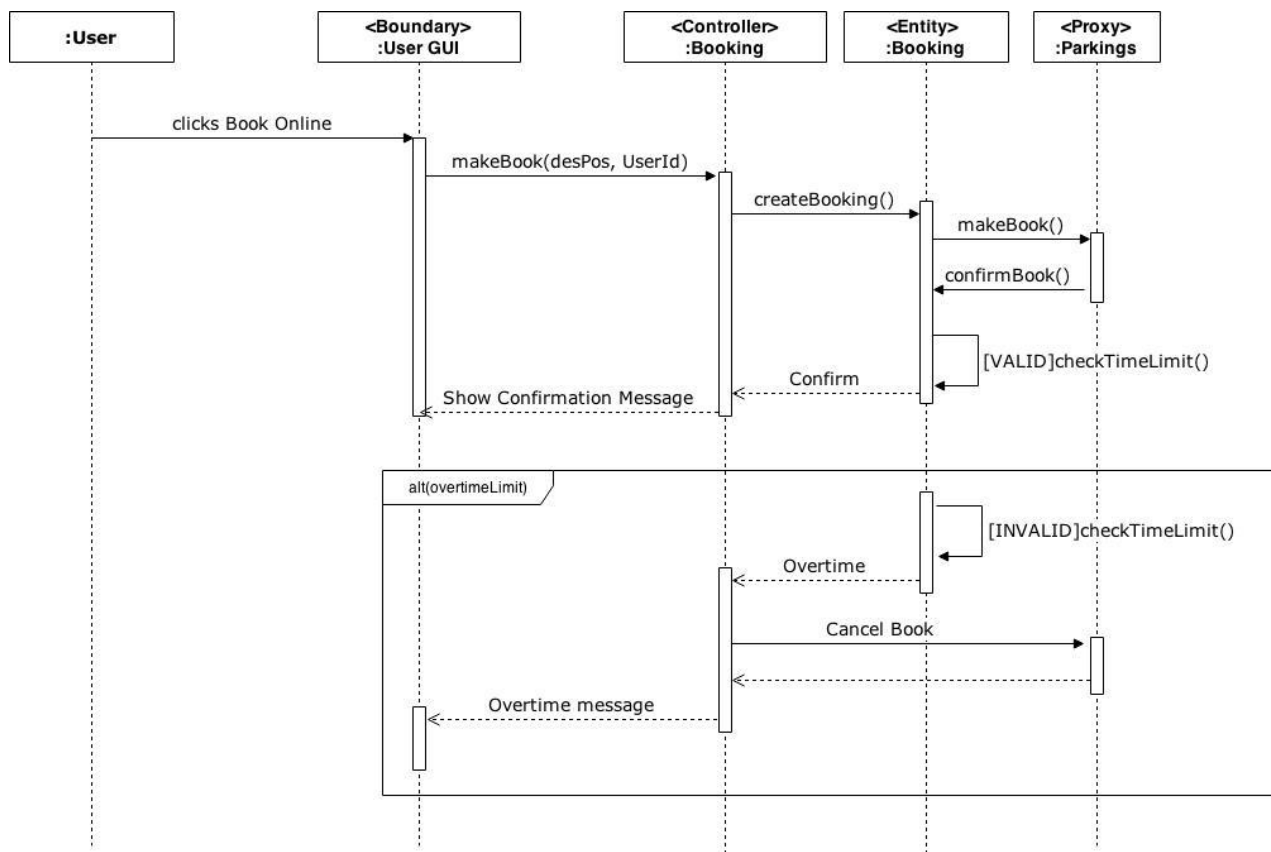
Το πακέτο αυτό αφορά την περίπτωση που ο χρήστης μπορεί να κάνει κράτηση θέσης πριν να φτάσει στον χώρο στάθμευσης που επιθυμεί. Η χρονική διάρκεια της κράτησης θέσης ορίζεται στα 10 λεπτά έτσι ώστε να αποφεύγεται η μόνιμη κράτηση θέσης.

Οι λειτουργικές απαιτήσεις που αντιστοιχίζονται σε αυτό το πακέτο αφήγησης είναι:

- <ΛΑ-9> Ο απλός χρήστης πρέπει να μπορεί να κάνει κράτηση θέσης.

#### 3.1.4.1 Αφήγηση Σεναρίου:

Ο User επιλέγει το κουμπί Book Online στο User GUI αφού έχει ήδη επιλέξει τον επιθυμητό χώρο στάθμευσης. Εκτελείται η makeBook(desPos, UserId) και ο Booking Controller δημιουργά την οντότητα Booking μέσω της συνάρτησης createBooking(). Η οντότητα Booking κάνει την κράτηση με την makeBook() και ενημερώνει τη βάση δεδομένων Parkings. Μετά η οντότητα Booking ελέγχει αν το χρονικό όριο της κράτησης ξεπεράστηκε. Μόλις ξεπεραστεί το όριο των 10 λεπτών ενημερώνει τον χρήστη ότι η κράτηση του ακυρώθηκε.



Σχήμα 42: Διάγραμμα ακολουθιών Κράτησης Θέσης



### 3.1.5 Πακέτο Τρέχουσας Χρέωσης.

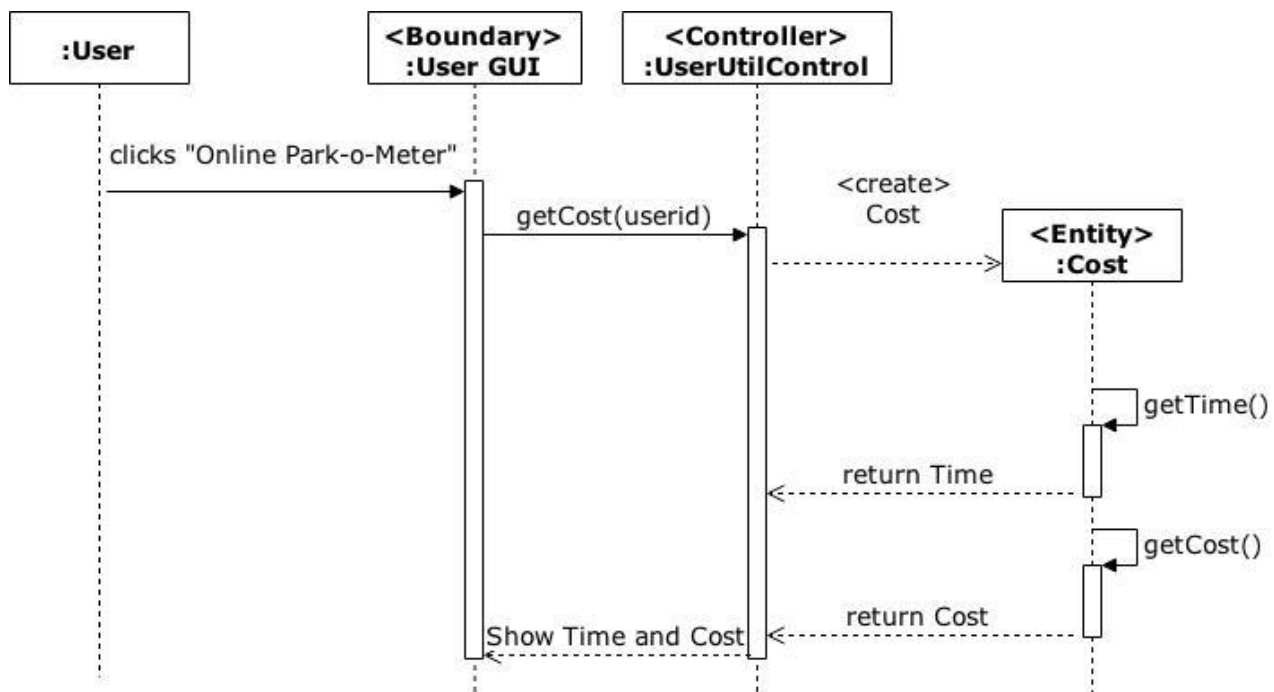
Το πακέτο αυτό αφορά την περίπτωση που ο χρήστης μπορεί να βλέπει τη τρέχουσα χρέωση του αφού έχει ήδη σταθμεύσει.

Οι λειτουργικές απαιτήσεις που αντιστοιχίζονται σε αυτό το πακέτο αφήγησης είναι:

- <ΛΑ-11> Ο απλός χρήστης πρέπει να μπορεί να βλέπει τη τρέχουσα χρέωση του.

#### 3.1.5.1 Αφήγηση Σεναρίου:

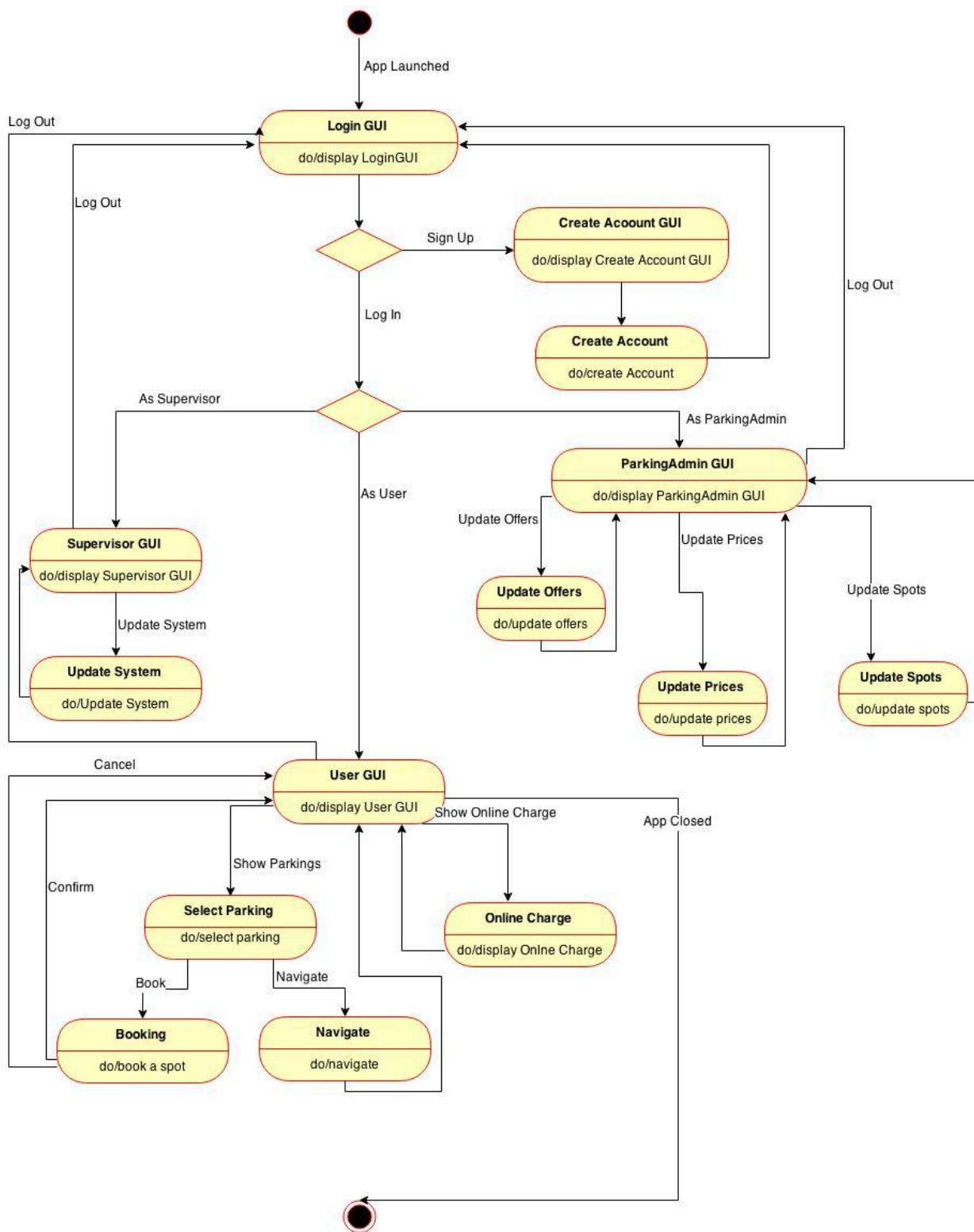
Ο User επιλέγει το κουμπί "Online Park-o-Meter" πακέτο User GUI. Εκτελείται η συνάρτηση `getCost(userid)` και ο `UserUtilControl` με τη συνάρτηση `createCost()` δημιουργεί την οντότητα `Cost`. Η οντότητα `Cost` με τις συναρτήσεις `getTime()` και `getCost()` επιστρέφει το χρόνο και τη τρέχουσα χρέωση του χρήστη.



Σχήμα 43: Διάγραμμα ακολουθιών τρέχουσας χρέωσης



### 3.2 Διάγραμμα Καταστάσεων



Σχήμα 45: Διάγραμμα καταστάσεων



### **3.3 Μη λειτουργικές απαιτήσεις**

#### **3.3.1 Απαιτήσεις επίδοσης**

##### **<ΜΛΑ-1>**

Το σύστημα πρέπει να επιτρέπει την ταυτόχρονη είσοδο όλων των χρηστών που επιθυμούν να εισέλθουν στο σύστημα.

User Priority: (5/5)

Technical Priority: (5/5)

Stability: (5/5)

#### **3.3.2 Απαιτήσεις ασφάλειας (Security)**

##### **<ΜΛΑ-4>**

Το σύστημα πρέπει να μπορεί να δώσει νέο κωδικό στο χρήστη.

User Priority: (5/5)

Technical Priority: (5/5)

Stability: (5/5)

#### **3.3.3 Απαιτήσεις Χρηστικότητας (Usability)**

##### **<ΜΛΑ-3>**

Το σύστημα πρέπει να εμφανίζει μήνυμα σφάλματος σε περίπτωση ταυτόχρονης κράτησης θέσης.

User Priority: (5/5)

Technical Priority: (3/5)

Stability: (5/5)

##### **<ΜΛΑ-5>**

Το σύστημα πρέπει να επιτρέπει μέγιστη διάρκεια κράτησης του χρήστη να είναι 10 λεπτά.

User Priority: (5/5)

Technical Priority: (4/5)

Stability: (2/5)

#### **3.3.4 Απαιτήσεις Αξιοπιστίας (Reliability)**

##### **<ΜΛΑ-2>**

Το σύστημα πρέπει να επιτρέπει ως μέγιστη κράτηση το πολύ δύο θέσεων μέσα σε διάστημα δύο ωρών.

User Priority: (5/5)

Technical Priority: (2/5)

Stability: (5/5)





## Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας

### Πακέτο Βασικής Διεπαφής Χρήστη (Κεφάλαιο 2.1)

- **Πακέτο Users**  
<ΛΑ-1>,<ΛΑ-2>,<ΛΑ-3>
- **Πακέτο GUI**  
<ΛΑ-1>,<ΛΑ-2>,<ΛΑ-3>,<ΛΑ-4>,<ΛΑ-5>,<ΛΑ-6>,<ΛΑ-7>,<ΛΑ-8>,<ΛΑ-9>,<ΛΑ-10>,<ΛΑ-11>

### Πακέτα Utilities (Κεφάλαιο 2.2)

- **Πακέτο User Utilities**  
<ΛΑ-5>,<ΛΑ-6>,<ΛΑ-7>,<ΛΑ-8>,<ΛΑ-9>,<ΛΑ-10>,<ΛΑ-11>
- **Πακέτο ParkingAdmin Utilities**  
<ΛΑ-3>

### Πακέτο SystemControl (Κεφάλαιο 2.3)

<ΛΑ-1>,<ΛΑ-2>,<ΛΑ-3>,<ΛΑ-4>,<ΛΑ-5>,<ΛΑ-6>,<ΛΑ-7>,<ΛΑ-8>,<ΛΑ-9>,<ΛΑ-10>,<ΛΑ-11>

### Πακέτο DataBase (Κεφάλαιο 2.4)

<ΛΑ-3>

### Πακέτο API (Κεφάλαιο 2.5)

<ΛΑ-6>,<ΛΑ-7>,<ΛΑ-10>



## Παράρτημα II – Ανοιχτά Θέματα

Λόγω της περιορισμένης έκτασης της εργασίας κάποιες συναρτήσεις ή κλάσεις δεν χρησιμοποιήθηκαν στα διαγράμματα ακολουθιών και κλάσεων. Θα μπορούσε το έγγραφο να είναι πιο αναλυτικό και πλήρως λειτουργικό για όλες τις διαδικασίες της εφαρμογής, αλλά για λόγους απλότητας και περιορισμού του μεγέθους του εγγράφου παραλήφθηκαν οι λεπτομέρειες.