



Τεχνολογία Λογισμικού
Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα ΗΜΜΥ
Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2015

Σύστημα *ThessPark*

We spot your parking place.

Σχεδίαση Συστήματος

Del.3.1

Version 1.0

Μουρούζης Χρήστος, chrimour@auth.gr
Τσακάλης Κωνσταντίνος, tsakalis@auth.gr
Μιχαήλ Μάριος, mamichai@auth.gr
Αγρότης Ιωάννης, iagrotis@auth.gr
22/5/2015



22/05/2015

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του Κ. Ε. Wiegers ^{1♣} και του Μ. Smialek's.	0.1
Α. Συμεωνίδης	29/3/2014	Προσαρμογή του εγγράφου.	0.1.3
Χ. Ζολώτας	2/4/2015	Προσθήκη διαγράμματος καταστάσεων	0.1.4

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Θ.Μαυρίδης	*	themis.mavridis@issel.ee.auth.gr
Κωνσταντίνος Τσακάλης	27	tsakalis@auth.gr
Μιχαήλ Μάριος	27	mamichai@auth.gr
Μουρούζη Χρήστος	27	chrimour@auth.gr
Αγρότης Ιωάννης	27	iagrotis@auth.gr

^{1♣} 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/>

22/05/2015

ΠΕΡΙΕΧΟΜΕΝΑ

Λίστα Σχημάτων	4
Εισαγωγή	5
1.1 Στόχος του εγγράφου	5
1.2 Αντικείμενο του λογισμικού	5
1.3 Ορισμοί, Ακρωνύμια, Συντομεύσεις	5
1.4 Στόχοι Σχεδίασης	6
1.5 Αναγνωστικό Κοινό. Τρόπος Ανάγνωσης εγγράφου	6
1.6 Επισκόπηση Εγγράφου	7
2. Τρέχουσα Αρχιτεκτονική Λογισμικού	8
2.1 Αρχιτεκτονική Client-Server (Πελάτη-Διακομιστή)	8
2.2 Αρχιτεκτονική τριών επιπέδων (3 -Tier)	9
2.3 Αρχιτεκτονική του συστήματος ThessPark	10
3. Προτεινόμενη Αρχιτεκτονική Λογισμικού	12
3.1 Αποδόμηση Συστήματος	12
3.2 Απεικόνιση Υλικού/Λογισμικού	17
3.3 Διαχείριση Μόνιμων Δεδομένων	21
3.4 Έλεγχος Πρόσβασης και Ασφάλεια	21
3.5 Γενικός έλεγχος λογισμικού	23
3.6 Οριακές Συνθήκες	24
3.6.1 Έναρξη συστήματος	24
3.6.2 Ενημέρωση Λογισμικού και Συντήρηση Συστήματος	24
3.6.3 Διακοπή σύνδεσης	24
3.6.4 Υπερφόρτωση Συστήματος	24
3.6.5 Σφάλματα Λογισμικού	25
4. Προδιαγραφές Τμηματικού Σχεδιασμού (Component Design Specifications)	26
4.1 Πρότυπα Σχεδιασμού που υιοθετήθηκαν	26
4.1.1 Δομικά Πρότυπα	26
4.1.2 Πρότυπα Συμπεριφοράς	27
Παράρτημα Ι – Ανοιχτά Θέματα	30

22/05/2015

Λίστα Σχημάτων

Σχήμα 1 - UserTypes	12
Σχήμα 2 - UsersDBProxy	13
Σχήμα 3 - ParkingsDBProxy.....	13
Σχήμα 4 - BasicUserInterface.....	14
Σχήμα 5 - BasicUserControl	14
Σχήμα 6 - SupervisorControl.....	15
Σχήμα 7 - ParkingAdminControl	15
Σχήμα 8 - MapsInteraction	16
Σχήμα 9 - Διάγραμμα Τμημάτων.....	17
Σχήμα 10 - Client Device.....	18
Σχήμα 11 - Server Device.....	19
Σχήμα 12 - Διάγραμμα Ανάπτυξης	20
Σχήμα 13 - Bridge Design Pattern.....	27
Σχήμα 14 - Facade Design Pattern.....	28
Σχήμα 15 - Observer Design Pattern	29



22/05/2015

Εισαγωγή

1.1 Στόχος του εγγράφου

Το παρόν έγγραφο <<Σχεδιασμός Συστήματος>> αποτελεί συνέχεια των δύο προηγούμενων παραδοτέων. Στόχος του εγγράφου είναι η λεπτομερής περιγραφή του σχεδιασμού του συστήματος. Συγκεκριμένα, πραγματοποιείται κατάτμηση των λειτουργιών και παρατίθενται τα σχετικά διαγράμματα όπως επίσης και διαγράμματα τα οποία επεξηγούν τη ροή του σχεδιασμού του συστήματος.

Επίσης, έχει γίνει προσπάθεια διαχωρισμού των κλάσεων και προσθήκη βοηθητικών οντοτήτων με σκοπό την ανεξαρτησία μεταξύ των κλάσεων του συστήματος και της ευκολότερης εποπτείας και λειτουργίας του.

1.2 Αντικείμενο του λογισμικού

Το λογισμικό του συστήματος είναι υπεύθυνο για την ομαλή λειτουργία του συστήματος. Είναι εκείνη η οντότητα η οποία αναλαμβάνει να συνδυάσει και να κατευθύνει όλες τις λειτουργίες των επί μέρους οντοτήτων οι οποίες εκτελούν τις διαδικασίες που απαιτούνται. Συγκεκριμένα :

- Το λογισμικό, παρουσιάζει στο χρήστη τις δυνατές λειτουργίες που μπορεί να επιλέξει μέσα από ένα μενού. Αυτό επιτυγχάνεται μέσω του πακέτου διεπαφής χρηστών.
- Το λογισμικό, αναλαμβάνει σε δεύτερο επίπεδο την εκτέλεση των εντολών του χρήστη συντονίζοντας τα πακέτα ελέγχου και λειτουργιών.
- Το λογισμικό, είναι υπεύθυνο για την καλή συνεργασία μεταξύ του κυρίως συστήματος και του βοηθητικού συστήματος Google Maps.
- Το λογισμικό, είναι υπεύθυνο για την πληρότητα όλων των λειτουργικών απαιτήσεων καθώς και των μη λειτουργικών.

1.3 Ορισμοί, Ακρωνύμια, Συντομεύσεις

User: χρήστης

Parking Administrator: Διαχειριστής Χώρου Στάθμευσης

Supervisor: Επόπτης του συστήματος

Google Maps: Εξωτερικό σύστημα διαδραστικού χάρτη και πλοήγησης

HTTP: Hypertext Transfer Protocol

DB: Βάση Δεδομένων



22/05/2015

GUI: Graphical User Interface**UML:** Unified Modeling Language**API:** Application Programming Interface

1.4 Στόχοι Σχεδίασης

Η σωστή σχεδίαση του συστήματος προϋποθέτει την ικανοποίηση των δύο διαφορετικών

ομάδων φυσικών προσώπων: του χρήστη και του προγραμματιστή. Οι τρεις αυτές ομάδες έχουν μία κοινή απαίτηση προς το σύστημα, τη αξιοπιστία του. Πελάτης του συστήματος θεωρείται οποιοσδήποτε κάτοχος αυτοκινήτου ιδιωτικής χρήσης.

Όπως είναι λογικό, ο χρήστης, επιθυμεί το σύστημα να είναι χρηστικό, φιλικό προς αυτόν και η διαδικασία εκμάθησής του να είναι απλή, εύκολη και γρήγορη. Ακόμα, θέλει η λειτουργία του συστήματος να είναι ευσταθής, να ανέχεται σφάλματα και συνεπώς να είναι αποδοτική. Επίσης, ο προγραμματιστής του συστήματος, (ή ο συντηρητής) έχει την απαίτηση, να μην παρουσιάζονται σφάλματα στο σύστημα, να προσαρμόζεται εύκολα και γρήγορα σε μεταβολές, τροποποιήσεις και αναβαθμίσεις λογισμικού και ακόμα ο διαχωρισμός των υποσυστημάτων να είναι σωστός και ορισμένος με σαφήνεια, και πρέπει να τηρεί τις αρχές της τμηματικότητας, αφαιρετικότητας και συνθετικότητας.

Όταν κάθε δραστηριότητα του συστήματος, εκτελείται από μία μονάδα του, και οι είσοδοι και έξοδοί του είναι καλώς ορισμένες, το σύστημα ορίζεται ως τμηματοποιημένο. Σύμφωνα με την αρχή της αφαιρετικότητας, τα χαμηλότερα επίπεδα του συστήματος περιέχουν λεπτομέρεια και πολυπλοκότητα, ενώ τα υψηλότερα επίπεδα είναι αφαιρετικά, έτσι ώστε να αντιμετωπίζονται ευκολότερα προβλήματα που προκύπτουν. Η συνθετικότητα αναφέρεται στην κατασκευή σύνθετων δομών, αποτελούμενων από επιμέρους υποσυστήματα.

Ακόμα, η σχεδίαση των υποσυστημάτων θα πρέπει να γίνεται με στόχο την επίτευξη υψηλής συνεκτικότητας και χαμηλής σύζευξης. Η συνεκτικότητα μετράει τις εξαρτήσεις μεταξύ των κλάσεων και η συζευξη μετράει τις εξαρτήσεις μεταξύ των υποσυστημάτων.

Παρόλα αυτά, η απόλυτη ικανοποίηση όλων των παραπάνω απαιτήσεων δε μπορεί να πραγματοποιηθεί ταυτόχρονα, και συνεπώς θα πρέπει να ληφθούν αποφάσεις και να γίνουν συμβιβασμοί για τη σωστή και αρμονική λειτουργία του συστήματος.

1.5 Αναγνωστικό Κοινό. Τρόπος Ανάγνωσης εγγράφου.

Το παρόν έγγραφο σχεδίασης συστήματος απευθύνεται στο παρακάτω αναγνωστικό κοινό:



22/05/2015

- Προϊστάμενοι, αρμόδιοι μηχανικοί λογισμικού των εταιριών ανάπτυξης αυτόματων συστημάτων σε κατοικίες
- Προγραμματιστές που θα αναλάβουν τη συγγραφή του κώδικα του προγράμματος του συστήματος
- Μηχανικοί δικτύων που θα αναλάβουν την εγκατάσταση του συστήματος.

Αρχικά, είναι σημαντικό να αναφερθεί ότι θα πρέπει να έχει προηγηθεί η ανάγνωση των δύο προηγούμενων εγγράφων, αυτού των απαιτήσεων χρηστών και αυτού των απαιτήσεων λογισμικού, έτσι ώστε αφενός, να υπάρχει συνοχή και να έχουν γίνει κατανοητές οι απαιτήσεις του εγγράφου και αφετέρου να είναι εύκολη και κατανοητή η διαδικασία ανάγνωσης του. Επίσης, η ανάγνωση των κεφαλαίων θα πρέπει να γίνει με τη σειρά που υπάρχουν στο έγγραφο για να μη δημιουργηθούν προβλήματα στην κατανόηση του.

Οι μηχανικοί λογισμικού των εταιριών ανάπτυξης αυτόματων συστημάτων, θα πρέπει να δώσουν προσοχή στο επόμενο κεφάλαιο, έτσι ώστε να γίνουν κατανοητοί οι όροι της αρχιτεκτονικής του συστήματος και των υποσυστημάτων, αλλά ταυτόχρονα για να πιστοποιήσουν αν πληρούνται οι στόχοι σχεδίασης.

Αντίστοιχα, οι προγραμματιστές και οι μηχανικοί συστημάτων θα πρέπει να δώσουν έμφαση στην κατανόηση των κεφαλαίων 3 και 4, καθώς εκεί παρουσιάζεται η προτεινόμενη αρχιτεκτονική του συστήματος και η αποδόμηση του σε υποσυστήματα.

1.6 Επισκόπηση Εγγράφου

Το πρώτο κεφάλαιο του εγγράφου είναι εισαγωγικό και περιλαμβάνει γενικές πληροφορίες

για το περιεχόμενο του εγγράφου αλλά και για τα κεφάλαια που ακολουθούν.

Το δεύτερο κεφάλαιο αναλύει την αρχιτεκτονική πάνω στην οποία βασίζεται το σύστημα, κάνει αναφορά σε ήδη υλοποιημένα συστήματα με παρόμοια λειτουργικότητα και τέλος παρουσιάζει την αρχιτεκτονική του συστήματος ThessPark.

Το τρίτο κεφάλαιο περιγράφει την αποδόμηση του συστήματος. Το συνολικό σύστημα διασπάται σε υποσυστήματα και επιμέρους τμήματα, σύμφωνα με την αρχιτεκτονική που επιλέχθηκε στο δεύτερο κεφάλαιο και ταυτόχρονα παρουσιάζονται διαγράμματα συνδέσεων μεταξύ των υποσυστημάτων. Παράλληλα, αναλύονται θέματα γενικού ελέγχου, ασφάλειας, πρόσβασης και οριακών συνθηκών στα οποία πρέπει να δοθεί ιδιαίτερη προσοχή.

Το τέταρτο κεφάλαιο περιγράφει τα πρότυπα σχεδίασης που θα πρέπει να εφαρμοσθούν στα υποσυστήματα του ThessPark και τα οποία προσδιορίζουν γενικές



22/05/2015

σχεδιαστικές λύσεις για ένα επαναλαμβανόμενο κοινό πρόβλημα, που εμφανίζεται στο σύστημα.

Το παράρτημα I παρουσιάζει τις ενδεχόμενες αλλαγές που έγιναν στα έγγραφα απαιτήσεων χρηστών και συστήματος μέσω του πίνακα ιχνηλασιμότητας.

Το παράρτημα II παρουσιάζει μια λίστα με ανοιχτά θέματα, τα οποία μπορούν να συμπεριληφθούν στο σύστημα μελλοντικά, σε σχέση με τις υπάρχουσες ή καινούργιες απαιτήσεις του πελάτη.

2. Τρέχουσα Αρχιτεκτονική Λογισμικού

2.1 Αρχιτεκτονική Client-Server (Πελάτη-Διακομιστή)

Το μοντέλο πελάτη - διακομιστή είναι μια διανεμημένη δομή αποτελούμενη από έναν ή περισσότερους διακομιστές, δηλαδή συστήματα που παρέχουν ένα σύνολο υπηρεσιών, και από τους πελάτες, οι οποίοι πρόκειται να κάνουν χρήση των υπηρεσιών αυτών.

Η επικοινωνία μεταξύ τους γίνεται δια μέσου ενός δικτύου, τοπικής ή ευρείας εμβέλειας, όπου οι πελάτες αποστέλλουν αιτήματα και οι απομακρυσμένοι διακομιστές αναμένουν και άμεσα αποκρίνονται σε αυτά.

Οι πελάτες πρέπει να γνωρίζουν τους διακομιστές που είναι διαθέσιμοι και τις διεπαφές που αυτοί παρέχουν, αλλά συνήθως δεν γνωρίζουν την ύπαρξη άλλων πελατών. Πελάτες και διακομιστές αποτελούν ξεκάθαρα διαφορετικές διεργασίες.

Η αρχιτεκτονική αυτή παρέχει πλεονεκτήματα που ικανοποιούν διαφορετικές απαιτήσεις για υπολογιστικούς πόρους στις διεργασίες πελάτη και διακομιστή, όπως ταχύτητα επεξεργαστή, μεγάλη μνήμη, χωρητικότητα δίσκου και ύπαρξη των απαιτούμενων συσκευών εισόδου/εξόδου στο σύστημα. Το περιβάλλον είναι συνήθως ετερογενές, δηλαδή η πλατφόρμα υλικού και το λειτουργικό σύστημα του πελάτη και του διακομιστή δεν είναι ίδιο.

Ένα επιπλέον σημαντικό χαρακτηριστικό της αρχιτεκτονικής αυτής είναι η επεκτασιμότητα, οριζόντια ή κάθετη. Οριζόντια επεκτασιμότητα δηλώνει τη δυνατότητα προσθήκης ή αφαίρεσης πελατών με μικρή (ως μηδαμινή) επίπτωση στην επίδοση του συνολικού μοντέλου, ενώ κάθετη δηλώνει τη δυνατότητα μετανάστευσης σε ένα μεγαλύτερο και ταχύτερο διακομιστή.

Ο σχεδιασμός συστημάτων πελάτη-διακομιστή πρέπει να αντανακλά τη λογική δομή της εφαρμογής που αναπτύσσεται. Ένας τρόπος δόμησης είναι η διαίρεση του συστήματος σε τρία επίπεδα, η οποία αναλύεται στην επόμενη ενότητα. Στα συγκεντρωτικά συστήματα, τα



22/05/2015

επίπεδα αυτά δεν χρειάζεται να είναι σαφώς διαχωρισμένα. Ωστόσο, κατά τον σχεδιασμό ενός καταναμεμημένου συστήματος θα πρέπει να γίνει σαφής διάκριση μεταξύ τους, επειδή με αυτόν τον τρόπο κάθε επίπεδο μπορεί να καταναμεηθεί σε διαφορετικούς υπολογιστικούς κόμβους.

2.2 Αρχιτεκτονική τριών επιπέδων (3 -Tier)

Στην τεχνολογία λογισμικού, η αρχιτεκτονική πολλών επιπέδων είναι κατά βάση μια αρχιτεκτονική πελάτη-διακομιστή, στην οποία η απεικόνιση γραφικού περιβάλλοντος, η εκτέλεση των λειτουργιών και οι λειτουργίες σχετικές με την διαχείριση δεδομένων και συστημάτων βρίσκονται σε ευκρινώς διαχωρισμένα επίπεδα.

Πιο συγκεκριμένα, στην αρχιτεκτονική τριών επιπέδων πελάτη-διακομιστή το γραφικό περιβάλλον της διεπαφής, το εκτελέσιμο κομμάτι του λογισμικού και ο χώρος που χρησιμοποιείται για την αποθήκευση των δεδομένων αναπτύσσονται και συντηρούνται ως ανεξάρτητες οντότητες σε ξεχωριστές πλατφόρμες. Το γραφικό περιβάλλον βρίσκεται στην πλευρά του πελάτη ενώ το εκτελέσιμο τμήμα του κώδικα και ο χώρος αποθήκευσης αναφέρονται στην πλευρά του διακομιστή.

Εκτός από τα συνήθη πλεονεκτήματα του μοντέλου πελάτη-διακομιστή, η αρχιτεκτονική πολλών επιπέδων επιτρέπει την αναβάθμιση οποιουδήποτε από τα επίπεδα, χωρίς να επηρεαστούν τα υπόλοιπα, ώστε να συμβαδίζουν με τις αλλαγές στην τεχνολογία.

Στις περισσότερες περιπτώσεις, το γραφικό περιβάλλον διεπαφής χρήστη εκτελείται σε ένα ή περισσότερα τερματικά τα οποία χρησιμοποιούν κάποια από τα συνήθη γραφικά περιβάλλοντα για την απεικόνιση. Το βασικό κομμάτι του λογισμικού που αποτελείται από μία ή περισσότερες οντότητες βρίσκεται σε κάποιο workstation ή application server, ενώ τα δεδομένα αποθηκεύονται σε ένα database server.

Αναλυτικά, τα επίπεδα της αρχιτεκτονικής τριών επιπέδων είναι:

- Παρουσίαση (Presentation Tier): Το επίπεδο της εφαρμογής που προβάλλεται στο χρήστη.

Είναι υπεύθυνο για την απεικόνιση των πληροφοριών στην εκάστοτε πλατφόρμα του χρήστη μέσω γραφικών διεπαφών. Επικοινωνεί με τα υπόλοιπα επίπεδα τόσο αποστέλλοντας τα αιτήματα στο διακομιστή εφαρμογής του δεύτερου επιπέδου, όσο και λαμβάνοντας τα επιστρεφόμενα δεδομένα.

- Επεξεργασία Εφαρμογής (Application Tier): Σε αυτό το επίπεδο βρίσκεται το κύριο κομμάτι του λογισμικού. Ελέγχει τις λειτουργίες του λογισμικού μέσω των εισόδων από το πρώτο επίπεδο και επεξεργάζεται τα δεδομένα που βρίσκονται στο τρίτο επίπεδο. Ένα λεπτό σημείο του επιπέδου αυτού είναι το γεγονός ότι μπορεί να αποτελεί από



22/05/2015

μόνο του μια πολυεπίπεδη αρχιτεκτονική, με αποτέλεσμα να έχουμε τις γενικότερες αρχιτεκτονικές πολλών επιπέδων (n-tier architecture).

- Διαχείριση Δεδομένων (Data Tier): Το τελευταίο επίπεδο της αρχιτεκτονικής περιλαμβάνει τους διακομιστές δεδομένων. Είναι το επίπεδο στο οποίο αποθηκεύονται και από το οποίο ανακτώνται τα δεδομένα, ενώ η επεξεργασία τους γίνεται αποκλειστικά στο δεύτερο επίπεδο. Με αυτό τον τρόπο, τα δεδομένα διατηρούνται ουδέτερα και ανεξάρτητα από τους χρησιμοποιούμενους διακομιστές του δεύτερου επιπέδου. Με την δημιουργία αυτού του επιπέδου επιτυγχάνεται αποδοτικότητα και επεκτασιμότητα του συστήματος.

Πρόσθετα πλεονεκτήματα της αρχιτεκτονικής τριών επιπέδων είναι:

- Η επεξεργασία των δεδομένων γίνεται σε τοπικό επίπεδο, μεταξύ 2ου και 3ου επιπέδου, μεταφέροντας στο χρήστη μόνο το επιθυμητό αποτέλεσμα, ελαχιστοποιώντας έτσι την χρήση του δικτύου (για online εφαρμογές).
- Μέσω της χρήσης διαφορετικών διακομιστών για δεδομένα και εφαρμογές επιτυγχάνεται κατανομή του φόρτου εργασίας.
- Λόγω της ανεξαρτησίας μεταξύ των επιπέδων είναι εφικτή η χρήση διαφορετικής πλατφόρμας σε κάθε επίπεδο, ώστε να μεγιστοποιηθεί η απόδοση του συστήματος.

Η αρχιτεκτονική αυτή είναι κατάλληλη για εφαρμογές μεγάλης κλίμακας με εκατοντάδες ή χιλιάδες πελάτες, εφαρμογές που τόσο τα δεδομένα όσο και η εφαρμογή είναι ευμετάβλητες, καθώς και σε εφαρμογές στις οποίες ενοποιούνται δεδομένα από πολλές πηγές.

2.3 Αρχιτεκτονική του συστήματος ThessPark

Σύμφωνα με την ανάλυση που προηγήθηκε το λογισμικό ThessPark θα ακολουθήσει ανάπτυξη αρχιτεκτονικής τριών επιπέδων.

Το πρώτο επίπεδο (επίπεδο παρουσίασης) θα αποτελείται από το σύνολο των γραφικών διεπαφών που θα προβάλλονται κάθε φορά στο χρήστη, ανάλογα με τη συσκευή από την οποία αποκτά πρόσβαση στο λογισμικό. Για να εκτελεσθεί μια λειτουργία του προγράμματος οι χρήστες θα αλληλεπιδρούν με το γραφικό περιβάλλον, είτε έχουν συνδεθεί στο σύστημα με τα δικά τους στοιχεία χρήστη, είτε το χρησιμοποιούν ως επισκέπτες.

Το δεύτερο επίπεδο (επίπεδο εφαρμογής) θα περιλαμβάνει τους διακομιστές στους οποίους είναι εγκατεστημένο το σύστημα. Οι απομακρυσμένοι αυτοί διακομιστές θα είναι



22/05/2015

υπεύθυνοι για την επεξεργασία των δεδομένων, την εκτέλεση των αιτούμενων λειτουργιών από τα συστήματα ελέγχου και την αποστολή των αποτελεσμάτων στη συσκευή από την οποία έχει πρόσβαση ο χρήστης. Στο ίδιο επίπεδο βρίσκεται και ο εφεδρικός διακομιστής, ο οποίος τίθεται εν ενεργεία όταν σταματάει η λειτουργία του κανονικού ή βρίσκεται σε φάση ενημέρωσης.

Τέλος, το τρίτο επίπεδο (επίπεδο δεδομένων) θα περιλαμβάνει έναν ή περισσότερους ήδη υπάρχοντες διακομιστές, στους οποίους είναι αποθηκευμένες οι βάσεις δεδομένων.

Η αρχιτεκτονική αυτή προτιμήθηκε καθώς μπορεί να παρέχει ανεξαρτησία μεταξύ των επιπέδων, εύκολη συντήρηση και αναβάθμιση του συστήματος, αλλά και καλύτερη απόδοση του αναπτυσσόμενου λογισμικού σε σχέση με την απλή προσέγγιση πελάτη-διακομιστή.

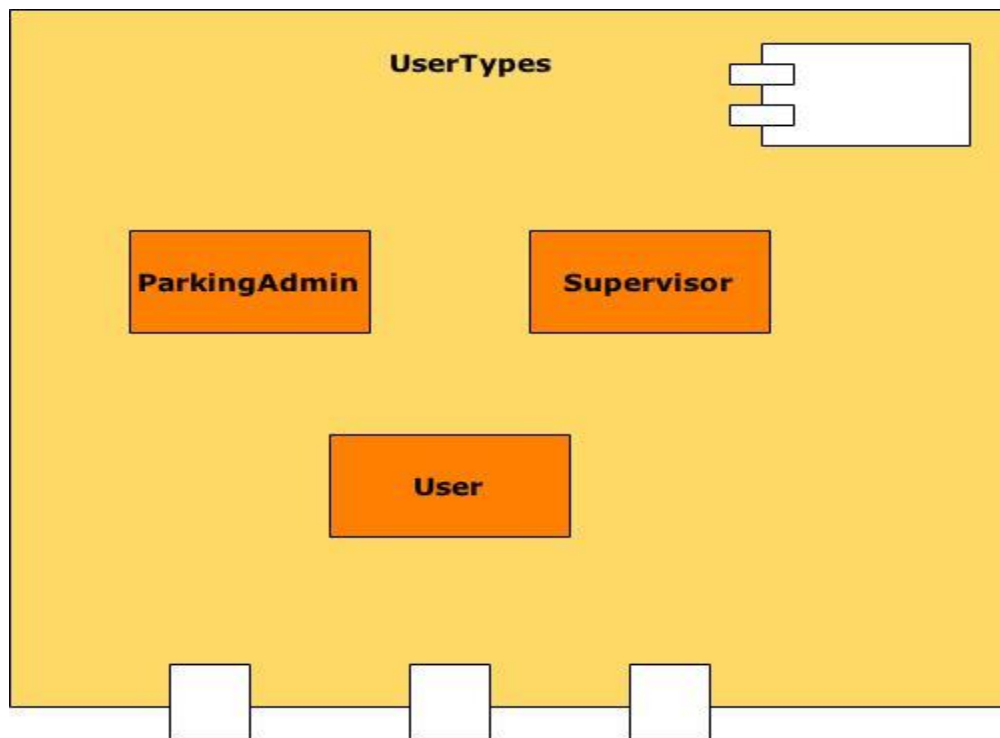
22/05/2015

3. Προτεινόμενη Αρχιτεκτονική Λογισμικού

3.1 Αποδόμηση Συστήματος

Η αποδόμηση του συστήματος περιγράφει την αποδόμηση σε τμήματα και τις ευθύνες του κάθε τμήματος/υποσυστήματος ξεχωριστά. Η ανάλυση θα ξεκινήσει ορίζοντας τα υποσυστήματα αυτά και περιγράφοντας τη λειτουργικότητα που περιλαμβάνει το καθένα, καθώς και τα επιμέρους τμήματα (components) που τα αποτελούν.

Component Package 1 – User Types : Το εν λόγω τμήμα περιλαμβάνει τους τύπους χρήστη που μπορεί να δέχεται το σύστημα. Οι κλάσεις που περιλαμβάνει είναι οι οντότητες User, Admin και ParkingAdmin.

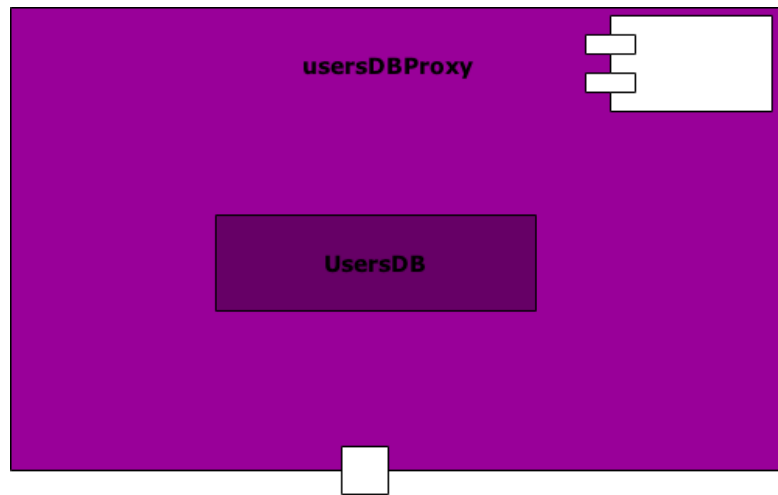


Σχήμα 1 - UserTypes



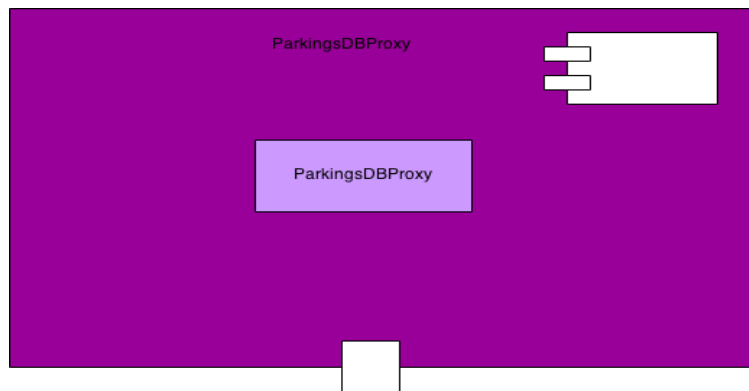
22/05/2015

Component Package 2 – UsersDBProxy: Η κλάση UsersDBProxy παρέχει πρόσβαση στη βάση δεδομένων που περιέχει τα στοιχεία των χρηστών του συστήματος. Η βάση αυτή βρίσκεται σε έναν εξωτερικό server, στον οποίο γίνεται σύνδεση για την ταυτοποίηση των χρηστών της εφαρμογής.



Σχήμα 2 - UsersDBProxy

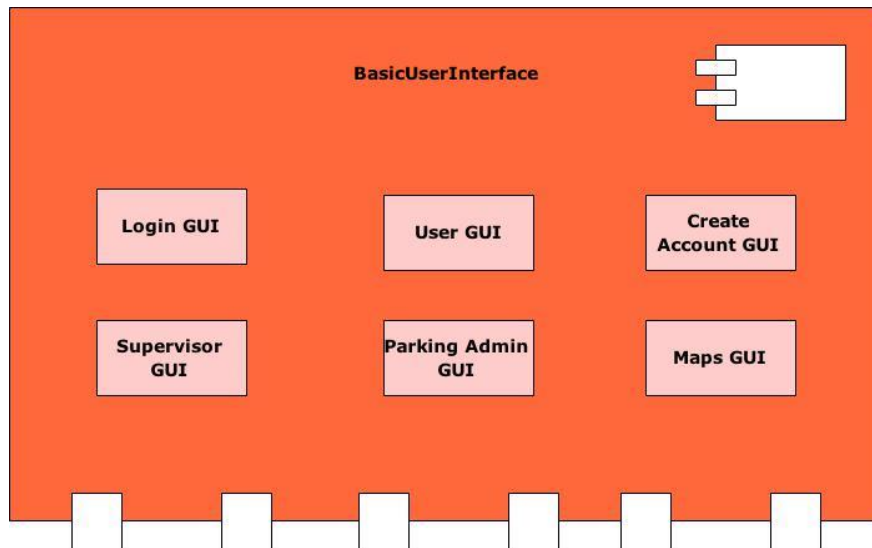
Component Package 3 – ParkingsDBProxy: Η κλάση ParkingsDBProxy παρέχει πρόσβαση στη βάση δεδομένων που περιέχει τα στοιχεία των χώρων στάθμευσης. Η βάση αυτή βρίσκεται σε έναν εξωτερικό server, στον οποίο γίνεται σύνδεση για την εμφάνιση ή ενημέρωση στοιχείων των χώρων στάθμευσης.



Σχήμα 3 - ParkingsDBProxy

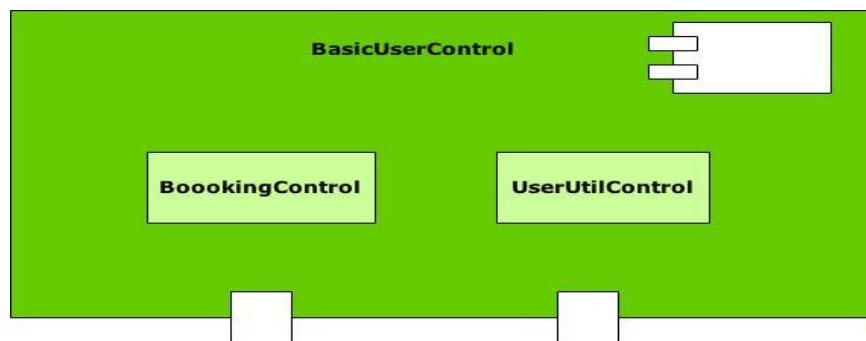
22/05/2015

Component Package 4 - BasicUserInterface : Το τμήμα αυτό περιλαμβάνει τις κύριες διεπαφές για κάθε είδους χρήστη, τη διεπαφή σύνδεσης χρηστών στο σύστημα, τη διεπαφή δημιουργίας λογαριασμού, τη διεπαφή διαχειριστή συστήματος και τη διεπαφή διαχειριστή χώρου στάθμευσης.



Σχήμα 4 - BasicUserInterface

Component Package 5 - BasicUserControl : Στο τμήμα αυτό περιέχονται οι ελεγκτές που αφορούν την επεξεργασία των δεδομένων που προκύπτουν από τις βασικές διεπαφές των χρηστών, όπως είναι η επιλογή χώρου στάθμευσης και πλοήγησης σε αυτόν, η προσωρινή κράτηση θέσης αλλά και εμφάνισης τρέχουσας χρέωσης.

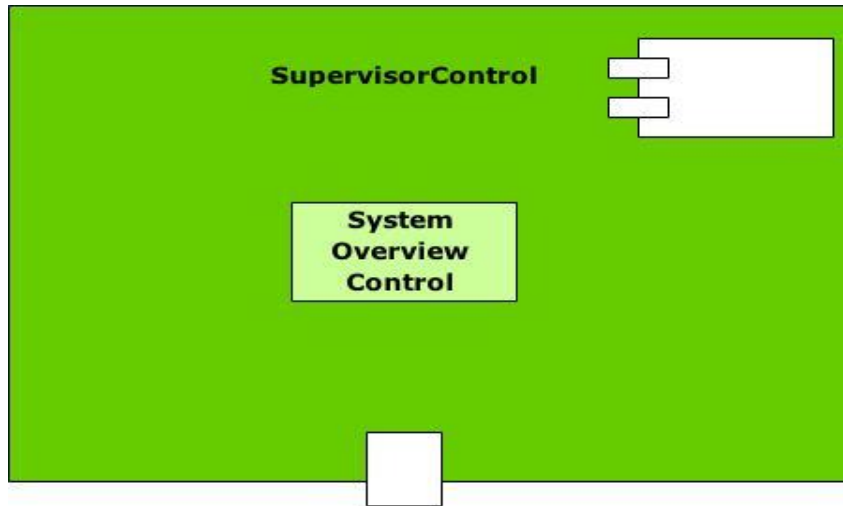


Σχήμα 5 - BasicUserControl



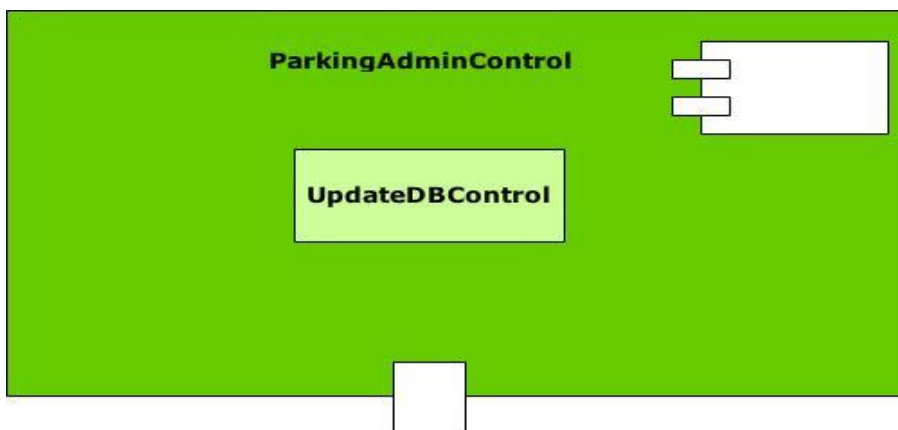
22/05/2015

Component Package 6 - SupervisorControl : Το τμήμα αυτό απαρτίζεται από τον ελεγκτή των διεπαφών του διαχειριστή. Η κλάση αυτή επεξεργάζεται τις πιθανές αλλαγές που πραγματοποιεί ο διαχειριστής στο σύστημα και του παρουσιάζει τα δεδομένα που επιθυμεί.



Σχήμα 6 - SupervisorControl

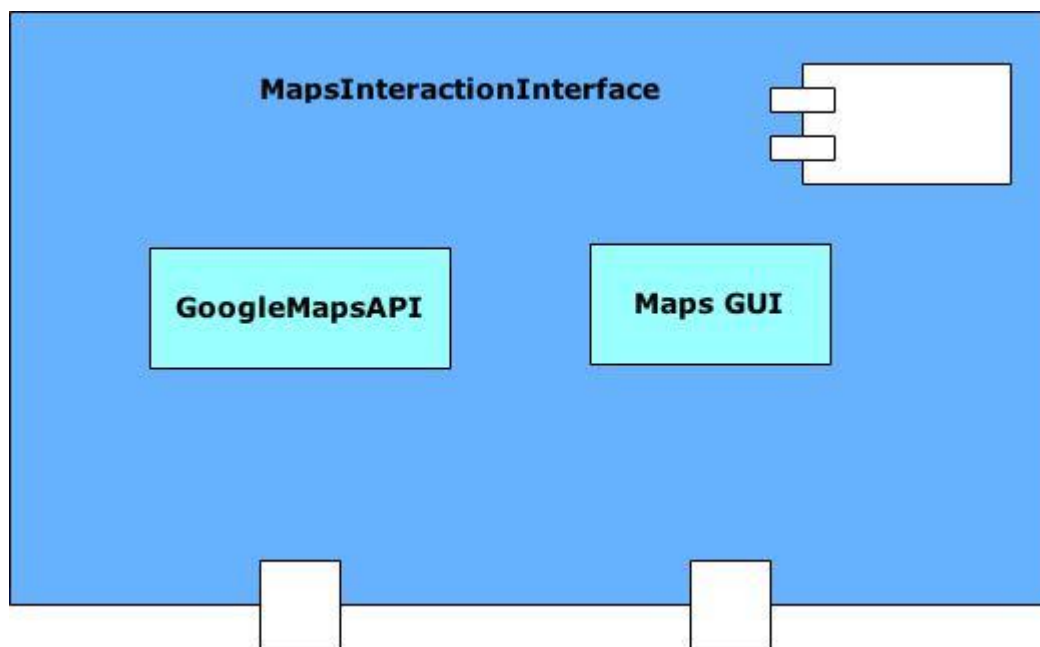
Component Package 7 - ParkingAdminControl : Το τμήμα αυτό απαρτίζεται από τον ελεγκτή των διεπαφών του διαχειριστή χώρου στάθμευσης. Η κλάση αυτή επεξεργάζεται τις πιθανές ενημερώσεις που πραγματοποιεί ο διαχειριστής όπως ενημέρωση τιμών, προσφορών και αριθμού χώρων στάθμευσης.



Σχήμα 7 - ParkingAdminControl

22/05/2015

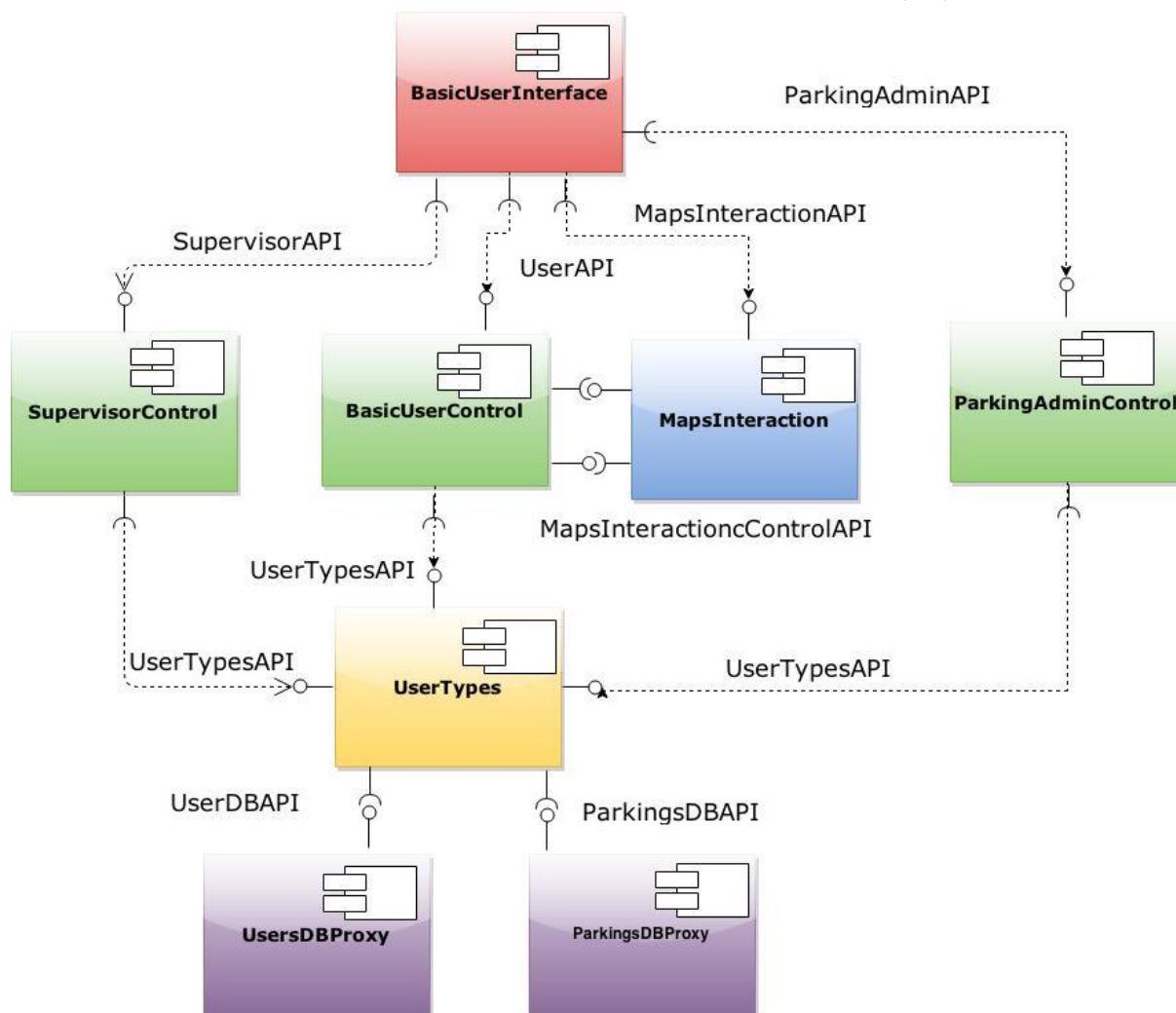
Component Package 8 - MapsInteraction : Το τμήμα αυτό απαρτίζεται από τον ελεγκτή της διεπαφής του διαδραστικού χάρτη.



Σχήμα 8 - MapsInteraction



22/05/2015



Σχήμα 9 - Διάγραμμα Τμημάτων

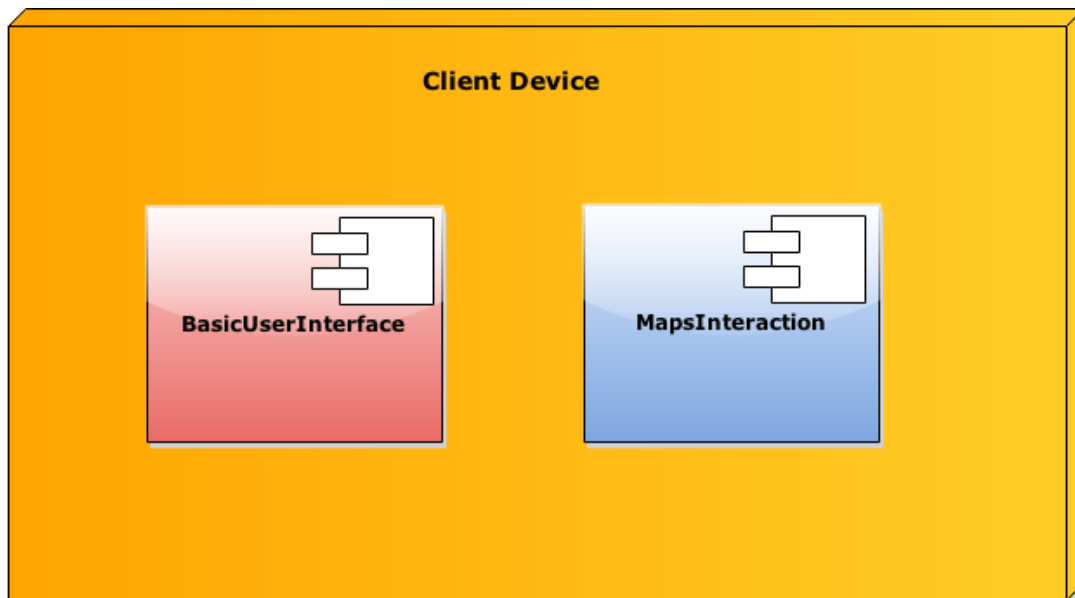
3.2 Απεικόνιση Υλικού/Λογισμικού

Το σύστημα χρησιμοποιεί το μοντέλο πελάτη - διακομιστή τριών επιπέδων, που παρέχει μια πρώτη διαμέριση ως προς το υλικό· τα τρία υποσυστήματα είναι δηλαδή διακριτές οντότητες. Με χρήση διαγραμμάτων ανάπτυξης στη UML, έχουμε τη δυνατότητα να παρουσιάσουμε την απεικόνιση του υλικού και λογισμικού, με στόχο να κατανοήσουν οι

22/05/2015

αναγνώστες του εγγράφου τις διασυνδέσεις των υποσυστημάτων αλλά και τις συσχετίσεις των τμημάτων με την υποδομή υλικού του συστήματος. Η περιγραφή χωρίζεται και πάλι ανάλογα με τα υποσυστήματα που έχουν οριστεί στην αποδόμηση συστήματος.

Στη συνέχεια παρουσιάζουμε τους κόμβους (nodes), οι οποίοι θα χρησιμοποιηθούν για την υλοποίηση του διαγράμματος ανάπτυξης του συστήματος.

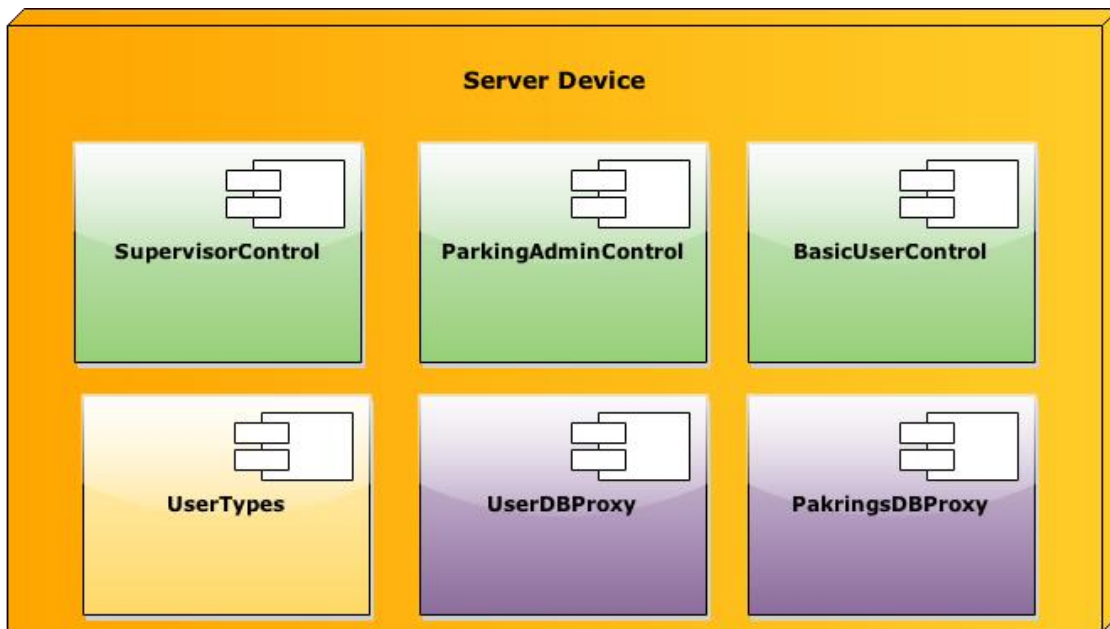


Σχήμα 10 - Client Device

Client Device : Ο κόμβος αυτός αναφέρεται γενικά στον πελάτη του συστήματος, δηλαδή στον υπολογιστή/smartphone το οποίο χρησιμοποιεί ο εκάστοτε χρήστης μια δεδομένη χρονική στιγμή. Ο χρήστης είναι αυτός που ξεκινά πάντα την επικοινωνία και ζητά υπηρεσίες από τον διακομιστή, μέσω ενός φυλλομετρητή (web browser), ή μιας smartphone εφαρμογής. Στον κόμβο αυτό προβάλλονται οι αναπαραστάσεις των γραφικών περιβάλλοντος, που δημιουργεί το σύστημα ανάλογα με τον τύπο και τις επιλογές του χρήστη. Αποτελεί το κύριο μέσο επικοινωνίας μέσα από το οποίο μπορεί ο κάθε χρήστης να χρησιμοποιεί το σύστημα.



22/05/2015



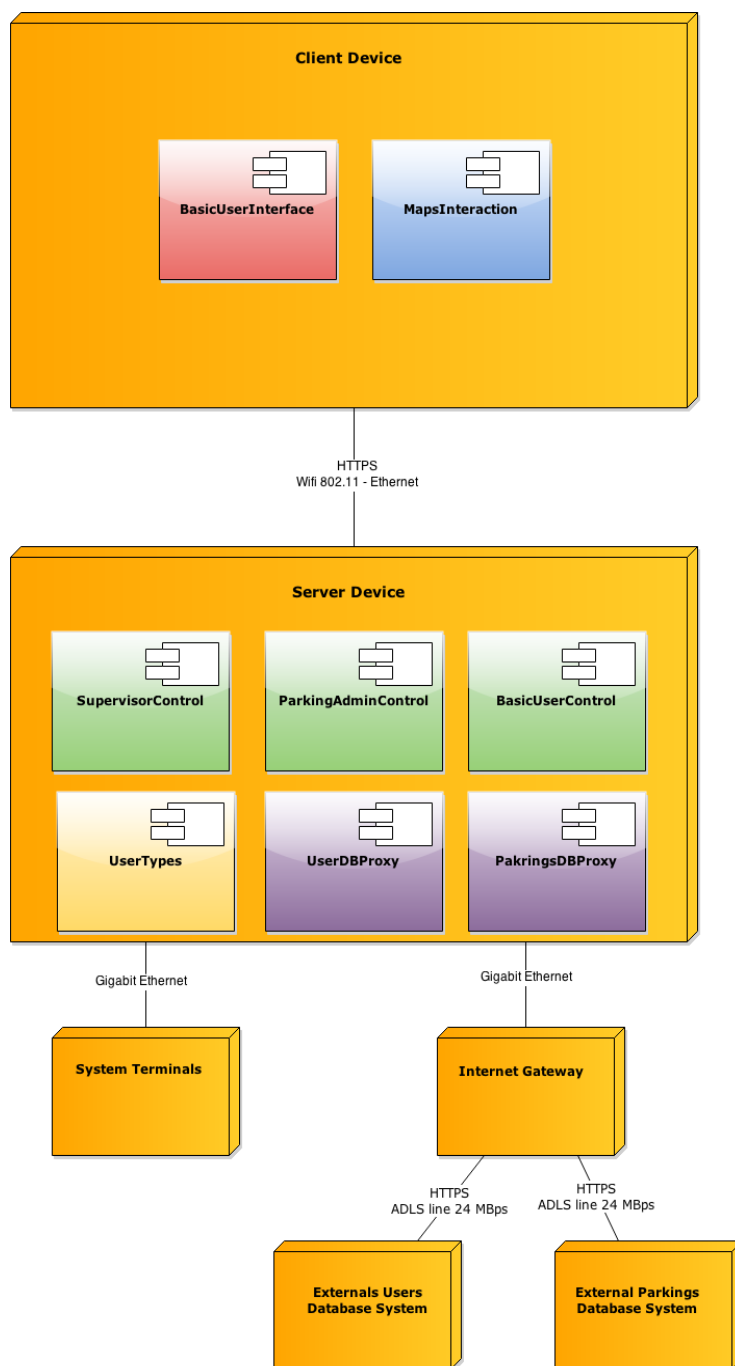
Σχήμα 11 - Server Device

System Server : Ο συγκεκριμένος κόμβος αποτελεί το κύριο τμήμα του συστήματος. Προσφέρει τις απαραίτητες υπηρεσίες στους χρήστες, μέσω την εσωτερικής επεξεργασίας των δεδομένων, αλλά και της επικοινωνίας με τις βάσεις δεδομένων. Είναι ουσιαστικά η υλοποίηση των οντοτήτων, καθώς και των κλάσεων ελέγχου οι οποίες έχουν οριστεί στο «Έγγραφο Απαιτήσεων Λογισμικού».

Πιο συγκεκριμένα, το σύστημα πάνω στο οποίο τρέχει ο Application Server, είναι πλήθος υπολογιστών, που είναι ανάλογο των απαιτήσεων σε υπολογιστική ισχύ. Το λειτουργικό σύστημα το οποίο διαχειρίζεται τον Application Server είναι Linux. Μεγάλη προσοχή πρέπει να δοθεί στην σωστή υλοποίηση του συστήματος ώστε να είναι εύκολη η κατανομή της απαιτούμενης ισχύος σε πολλούς υπολογιστές.

Επίσης, περιλαμβάνει τα τμήματα που παρέχουν πρόσβαση στις βάσεις δεδομένων που εμπλέκονται με το σύστημα, την εξωτερική βάση δεδομένων του συστήματος για την αυθεντικοποίηση χρηστών και την βάση δεδομένων των χώρων στάθμευσης.

22/05/2015



Σχήμα 12 - Διάγραμμα Ανάπτυξης

22/05/2015

3.3 Διαχείριση Μόνιμων Δεδομένων

Το σύστημα ThessPark δομείται με βάση την αρχιτεκτονική τριών επιπέδων. Ένα επίπεδο προορίζεται αποκλειστικά για την αποθήκευση των δεδομένων. Σε αυτό το επίπεδο βρίσκεται ο Main Server και ο αντίστοιχος Backup Server, ο οποίος χρησιμοποιείται σε περίπτωση ύπαρξης προβλημάτων λειτουργίας του κανονικού Main Server. Για τη συγγραφή των βάσεων δεδομένων χρησιμοποιήθηκε η γλώσσα MySQL. Η επιλογή έγινε γνωρίζοντας ότι η MySQL είναι μία από τις πιο διαδεδομένες γλώσσες ανάπτυξης βάσεων δεδομένων και διακρίνεται από υψηλή αξιοπιστία, απόλυτη συμβατότητα και πολύ καλή υποστήριξη. Τέλος, πρέπει να επισημανθεί ότι η ενημέρωση των βάσεων δεδομένων γίνεται σε πραγματικό χρόνο.

Το σύστημα χρησιμοποιεί βάσεις για την οργάνωση και διαχείριση των δεδομένων, που αφορούν τους χρήστες, τους χώρους στάθμευσης και τα δικαιώματα πρόσβασης των χρηστών. Παρακάτω ακολουθεί μια περιγραφή των βάσεων αυτών:

- Βάση δεδομένων χώρων στάθμευσης : Το σύστημα αποθηκεύει τις συντεταγμένες, τις τιμές, την διαθεσιμότητα σε θέσεις στάθμευσης, τις κρατήσεις θέσεων και τις διάφορες προσφορές του κάθε χώρου στάθμευσης,.
- Βάση δεδομένων χρηστών : Σε αυτή τη βάση είναι αποθηκευμένα τα στοιχεία και ο τύπος των χρηστών, με σκοπό να αποδίδονται σε αυτούς τα εκάστοτε δικαιώματα. Ακόμη, σε αυτή τη βάση δεδομένων αποθηκεύεται ο αριθμός κράτησης ή κατάληψης θέσης αλλά και η χρέωση του χρήστη.

3.4 Έλεγχος Πρόσβασης και Ασφάλεια

Το σύστημα ThessPark έχει τρεις διαφορετικές κατηγορίες χρηστών, τους απλούς χρήστες (ή επισκέπτες), τους διαχειριστές συστήματος και τους διαχειριστές των χώρων στάθμευσης. Κάθε μία από τις παραπάνω κατηγορίες έχει διαφορετικά δικαιώματα και μπορεί να εκτελέσει διαφορετικές λειτουργίες. Αυτή η κατηγοριοποίηση είναι απαραίτητη, ώστε να εκτελούνται συγκεκριμένες ενέργειες από κάθε τύπο χρήστη, προκειμένου να αποφευχθεί η πρόσβαση μη εξουσιοδοτημένων χρηστών σε κρίσιμες για το σύστημα διαδικασίες.

Για την επίτευξη της αυθεντικοποίησης των χρηστών, πραγματοποιείται σύνδεση με τη βάση δεδομένων χρηστών, η οποία προσδιορίζει την κατηγορία και τα δικαιώματα του κάθε χρήστη. Έπειτα, το λογισμικό εμφανίζει το γραφικό περιβάλλον που αντιστοιχεί στην κατηγορία του χρήστη. Εάν ο χρήστης επιθυμεί να πραγματοποιήσει κάποια ενέργεια στο σύστημα, την επιλέγει μέσω των συντομεύσεων του γραφικού περιβάλλοντος.



22/05/2015

Ένα σύστημα, όπως αυτό που αναπτύσσεται, είναι απαραίτητο να παρέχει ασφάλεια και προστασία στους χρήστες του. Με τον όρο ασφάλεια αναφερόμαστε σε τρία κυρίως σημεία:

- **Εχεμύθεια (confidentiality)** : Καθιστά ασφαλείς τις προσωπικές πληροφορίες των χρηστών, είτε αυτές αναφέρονται σε προσωπικά δεδομένα, είτε αναφέρονται σε αιτήσεις, εξασφαλίζοντας με βεβαιότητα ότι πρόσβαση έχουν μόνο εξουσιοδοτημένα άτομα.
- **Αξιοπιστία (integrity)** : Εξασφαλίζει ότι προγράμματα και πληροφορίες τροποποιούνται μόνο με συγκεκριμένους και εξουσιοδοτημένους τρόπους.
- **Διαθεσιμότητα (availability)** : Εξασφαλίζει ότι τα εξουσιοδοτημένα άτομα θα έχουν πρόσβαση στο σύστημα και το προφίλ τους σε κάθε περίπτωση.

Προκειμένου να επιτευχθούν αυτοί οι στόχοι, πρέπει πρώτα να καθοριστεί η πρόσβαση των διαφόρων χρηστών στις διαδικασίες του συστήματος. Για καλύτερη εποπτεία των δικαιωμάτων του κάθε χρήστη, δημιουργείται ο πίνακας πρόσβασης του συστήματος, ο οποίος παρουσιάζεται παρακάτω:

Guest	<i>Login GUI</i>	<i>Create Account GUI</i>
	btnCreateAccount()	create()
	btnLogin()	displayError()
	displayError()	
User	<i>User GUI</i>	<i>Maps GUI</i>
	btnNavigate()	getPosition()
	btnBooking()	getPark()
	selectParking()	navigate()
	btnOnlineCharge()	
Supervisor	<i>Supervisor GUI</i>	
	btnSysUpdate()	
	btnSysBackup()	



22/05/2015

	btnDBUpdate()	
Parking Admin	<i>Parking Admin GUI</i>	
	btnUpdatePrices()	
	btnUpdateOffers()	
	btnUpdateParkPlaces()	

3.5 Γενικός έλεγχος λογισμικού

Το ThessPark είναι ένα σύστημα στο οποίο μπορούν να έχουν πρόσβαση πολλοί χρήστες ταυτόχρονα. Μέσω των γραφικών διεπαφών, μπορούν να πραγματοποιούν λειτουργίες του συστήματος δίνοντας εντολές σε αυτό. Αυτές οι εντολές, μεταφέρονται σε αντικείμενα ελέγχου, τα οποία είτε τις εκτελούν, είτε τις προωθούν σε άλλα αντικείμενα ελέγχου. Έχοντας αυτό υπόψιν, και παρατηρώντας τα διαγράμματα ακολουθιών του εγγράφου απαιτήσεων λογισμικού, τα οποία έχουν μια δομή σκάλας και όχι πιρουνιού οδηγούμαστε στο συμπέρασμα να χρησιμοποιήσουμε το αποκεντρωμένο μοντέλο για τον έλεγχο του λογισμικού.

Η χρήση του αποκεντρωμένου μοντέλου έχει πλεονεκτήματα, όπως η ενίσχυση της αντικειμενοστρέφειας, αλλά ταυτόχρονα μοιράζει την ευθύνη του ελέγχου του συστήματος σε πολλά αντικείμενα.

Όσο αφορά το συγχρονισμό του συστήματος, εφόσον αυτό προσφέρει την ταυτόχρονη ύπαρξη πολλών χρηστών, και αυτοί μπορούν να παρεμβαίνουν επίσης ταυτόχρονα στο σύστημα, κάνοντας διαφορετικές ενέργειες πάνω σε αυτό, ο πολυνηματισμός είναι απαραίτητος. Έτσι, κατά την εισαγωγή ενός χρήστη στο σύστημα θα δημιουργείται ένα νήμα με ένα ID που το χαρακτηρίζει. Όταν ένας άλλος χρήστης εισέλθει στο σύστημα, τότε δημιουργείται ένα νήμα με διαφορετικό ID. Αυτό επιτρέπει την ταυτόχρονη επέμβαση των χρηστών στο σύστημα. Το νήμα που αντιστοιχεί σε κάθε χρήστη καταστρέφεται όταν αυτός αποσυνδεθεί από το σύστημα (Logout).

Επίσης διαφορετικά νήματα δημιουργούνται για κάθε προσωρινή κράτηση θέσης. Σε αυτή την περίπτωση το νήμα καταστρέφεται μετά από 10 λεπτά σε περίπτωση που δεν καταληφθεί η θέση σε αυτό το χρονικό διάστημα.

Διαφορετικά νήματα δημιουργούνται και στη περίπτωση κατάληψης θέσης στάθμευσης, ένα για κάθε θέση. Το ίδιο νήμα, είναι υπεύθυνο και για τη τρέχουσα χρέωση και καταστρέφεται μετά την πληρωμή από το χρήστη.

Επιπλέον το σύστημα διαχειρίζεται και ένα συγκεκριμένο αριθμό νημάτων, τα οποία αφορούν τον αυτόνομο χρονοπρογραμματισμό των συσκευών, δηλαδή για κάθε συνδεδεμένη εξωτερική συσκευή, το σύστημα διαθέτει ένα νήμα, που το διαχειρίζεται ένα αντικείμενο ελέγχου.



22/05/2015

3.6 Οριακές Συνθήκες

Το σύστημα ThessPark έχει ως κύριο στόχο την εξασφάλιση της αξιοπιστίας των υπηρεσιών του σε όλες του τις φάσεις και κατ' επέκταση στις οριακές συνθήκες του. Παρακάτω παρουσιάζονται οι βασικότερες περιπτώσεις των οριακών συνθηκών του συστήματος.

3.6.1 Έναρξη συστήματος

Το λογισμικό ThessPark θα εγκατασταθεί στον application server του συστήματος σε κάποια μεγάλης χωρητικότητας αποθηκευτική μονάδα. Ταυτόχρονα θα πρέπει να οριστούν και να συγχρονιστούν οι συνδέσεις με τα τερματικά και με εξωτερικά συστήματα, όπως είναι ο main server, ο backup server και τέλος, η σύνδεση με το διαδίκτυο.

3.6.2 Ενημέρωση Λογισμικού και Συντήρηση Συστήματος

Κατά καιρούς η ομάδα ανάπτυξης και συντήρησης του συστήματος θα προβαίνει σε βελτιώσεις, με αποτέλεσμα να έχουμε καινούργιες εκδόσεις, με ανανεωμένα ή ακόμα και πρόσθετα χαρακτηριστικά. Παρουσιάζεται, λοιπόν, η ανάγκη να ενημερώνεται το λογισμικό του συστήματος για αυτές τις αλλαγές.

3.6.3 Διακοπή σύνδεσης

Πολλές φορές υπάρχει περίπτωση η σύνδεση ενός χρήστη να διακοπεί προσωρινά, για τεχνικούς ή άλλους λόγους, πριν ο χρήστης να προλάβει να αποθηκεύσει τις όποιες αλλαγές έκανε μέχρι τη συγκεκριμένη χρονική στιγμή της διακοπής. Σε αυτή την περίπτωση, το σύστημα πρέπει να αναλαμβάνει την αποθήκευση της μέχρι τότε προόδου του χρήστη, ώστε να μπορέσει να ανακτήσει τα δεδομένα του, όταν ξανασυνδεθεί στο σύστημα. Αν κατά τη στιγμή της διακοπής ο χρήστης ήταν έτοιμος να εκτελέσει κάποια ενέργεια, λόγω των αλλοιωμένων πακέτων που θα καταφθάνουν στους servers του συστήματος, η ενέργεια αυτή θα απορρίπτεται.

3.6.4 Υπερφόρτωση Συστήματος

Σε περιπτώσεις όπου συνδέονται πάρα πολλοί χρήστες στο σύστημα, ή όταν πολλοί χρήστες προσπαθούν ταυτόχρονα να εκτελέσουν κάποιες ενέργειες, υπάρχει η πιθανότητα να υπερφορτωθεί το σύστημα. Σε μια τέτοια περίπτωση, για να αποφευχθεί η κατάρρευση του συστήματος, αλλά και για να επιτευχθεί μια λύση με τις λιγότερες δυνατές επιπτώσεις, το σύστημα αυτόματα θα δημιουργεί μια σειρά προτεραιότητας για τους χρήστες που είναι συνδεδεμένοι και για όσους θα συνδεθούν μετέπειτα.

22/05/2015

3.6.5 Σφάλματα Λογισμικού

Σφάλματα λογισμικού μπορούν να συμβούν κατά τη διάρκεια προσπέλασης μιας βάσης δεδομένων για την ενημέρωσή της, για ανάκτηση δεδομένων από αυτή ή κατά τη διάρκεια της εκτέλεσης κάποιας ενέργειας η οποία απαιτεί επικοινωνία του συστήματος με ένα εξωτερικό σύστημα. Σε περίπτωση σφάλματος, ακολουθούνται οι εξής διαδικασίες:

- Αναφορά : Γίνεται επαναφορά του συστήματος σε ευσταθή κατάσταση λειτουργίας, γίνεται μια απλή αναφορά στο στοιχείο που προκάλεσε το σφάλμα και δεν εκτελείται άλλη ενέργεια.
- Διόρθωση : Το σύστημα επιστρέφει στην προηγούμενη ευσταθή του κατάσταση για να εκτελεστεί αποσφαλμάτωση και εμφανίζεται στην οθόνη του χρήστη ένα μήνυμα το οποίο τον παρακινεί να εκτελέσει ξανά την ενέργεια η οποία διακόπηκε.
- Επανάληψη : Το σύστημα προσπαθεί να βρει έναν εναλλακτικό τρόπο για την εκτέλεση της ενέργειας που διακόπηκε λόγω σφάλματος.



22/05/2015

4. Προδιαγραφές Τμηματικού Σχεδιασμού (Component Design Specifications)

Το κεφάλαιο του Εγγράφου Σχεδιασμού Λογισμικού, περιγράφει την τελική αποδόμηση των υποσυστημάτων σε πακέτα καθώς και την οργάνωση του κώδικα στο σύστημα αρχείων του υπολογιστή. Η περιγραφή περιλαμβάνει μία επισκόπηση του κάθε πακέτου, τις εξαρτήσεις του με άλλα πακέτα και την προσδοκώμενη χρήση του.

4.1 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

Η λειτουργία του λογισμικού σε αντικειμενοστραφή περιβάλλοντα βασίζεται στη συνεργασία των αντικειμένων βάση της οποίας θα προκύψει η ορθή λειτουργία του συστήματος. Καθώς αναπτύσσονται συστήματα, είναι πιθανό να παρουσιάζονται στο εκάστοτε περιβάλλον, προβλήματα τα οποία εμφανίζονται πάνω από μία φορά. Το πρότυπο σχεδίασης (dressing pattern) είναι το μέσο το οποίο περιγράφει το πρόβλημα αυτό και περιγράφει τον πυρήνα της επίλυσης του με τέτοιο τρόπο ώστε να μπορεί η λύση να επαναχρησιμοποιηθεί χωρίς όμως να ξαναυλοποιηθεί.

Μια τέτοια πιθανή τροποποιήσιμη σχεδίαση μας δίνει την δυνατότητα να έχουμε ένα επαναληπτικό και προοδευτικό κύκλο ανάπτυξης με ευελιξία στις αλλαγές, καθώς και μείωση δημιουργίας νέων προβλημάτων βάση της επιλύσεως των παλιών, καθώς και παραγωγή περισσότερης λειτουργικότητας μετά το πρώτο πρωτότυπο.

Τα βασικά συστατικά μιας τέτοιας τροποποιήσιμης σχεδίασης είναι:

- ☐ Ύπαρξη χαμηλής σύζευξης και υψηλής συνεκτικότητας
- ☐ Ύπαρξη ξεκάθαρων εξαρτήσεων : Τα πρότυπα πρέπει να περιγράφουν βαθύτερους μηχανισμούς και σχέσεις- εξαρτήσεις , και όχι μόνο μια απλή αναφορά σ αυτά.
- ☐ Ύπαρξη υποθέσεων οι οποίες είναι σαφώς ορισμένες : Ένα πρότυπο οφείλει να παρουσιάζει τη λύση του προβλήματος και όχι μια γενική προσέγγιση ή μια στρατηγική.

Στα σχήματα που ακολουθούν και προβάλουν την εφαρμογή του κάθε προτύπου φαίνονται μόνο οι συναρτήσεις που προστίθενται κάθε φορά στις αντίστοιχες κλάσεις. Η αναφορά όλων των συναρτήσεων δεν προσφέρει κάτι πρόσθετο στον αναγνώστη.

4.1.1 Δομικά Πρότυπα

Τα δομικά πρότυπα εισάγουν μια abstract κλάση με σκοπό την μείωση της σύζευξης ανάμεσα σε δύο ή και περισσότερες κλάσεις, την ευκολότερη τυχόν επέκταση του συστήματος στο μέλλον, καθώς και να κάνουν εφικτή την ύπαρξη πιο σύνθετων δομών.

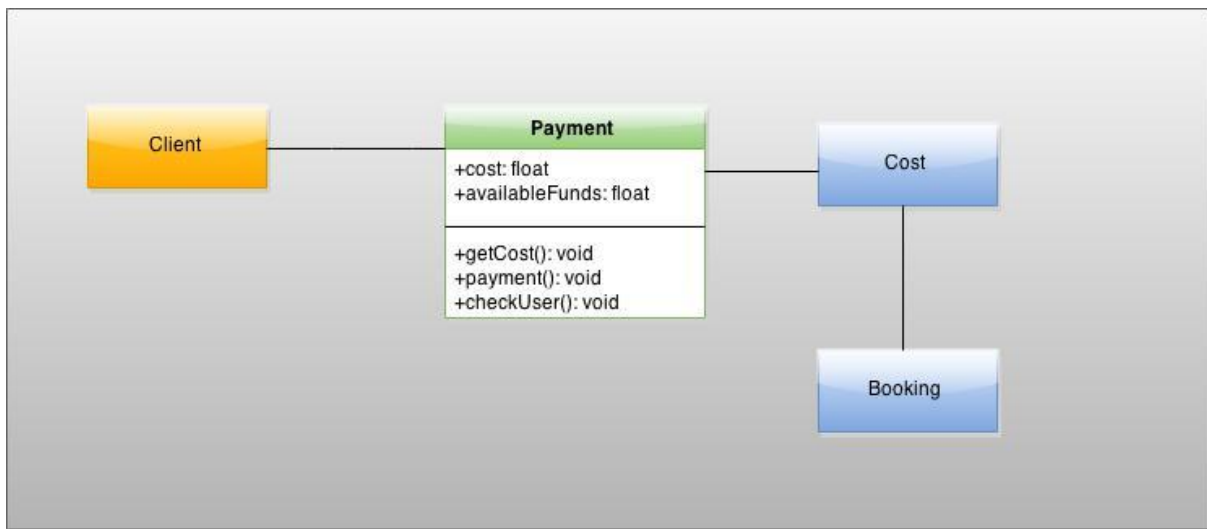
22/05/2015

Όνομα προτύπου : Bridge Design Pattern

Ταξινόμια : Δομικά Πρότυπα

Πρόβλημα που αντιμετωπίστηκε: Ένα σημαντικό κενό στην υλοποίηση μας λόγω περιορισμένων λειτουργικών απαιτήσεων, το οποίο είναι αναγκαίο να προσδιορισθεί είναι η μη ύπαρξη τρόπου πληρωμής του κόστους του εκάστοτε πελάτη, πράξη που κάνει την εφαρμογή πολύ πιο χρήσιμη για τον χρήστη.

Υποσυστήματα που αναδιοργανώθηκαν: Για την επίτευξη του σκοπού αυτού διαμορφώθηκε μια καινούργια κλάση Payment, η οποία μέσω από την κλάση Booking παίρνει το κόστος του πελάτη και αφού ελέγξει το διαθέσιμο υπόλοιπο του χρήστη εκτελεί την πράξη της πληρωμής εάν αυτή είναι δυνατή.



Σχήμα 13 - Bridge Design Pattern

4.1.2 Πρότυπα Συμπεριφοράς

Τα πρότυπα συμπεριφοράς μας δίνουν την δυνατότητα να επιλέξουμε ανάμεσα σε αλγόριθμους καθώς και να ορίσουμε ευθύνες ανάμεσα σε αντικείμενα. Επίσης έχουν την δυνατότητα να χαρακτηρίσουν σύνθετες ροές ελέγχου, οι οποίες κατά την εκτέλεση τους είναι πολύ δύσκολο να παρακολουθηθούν.

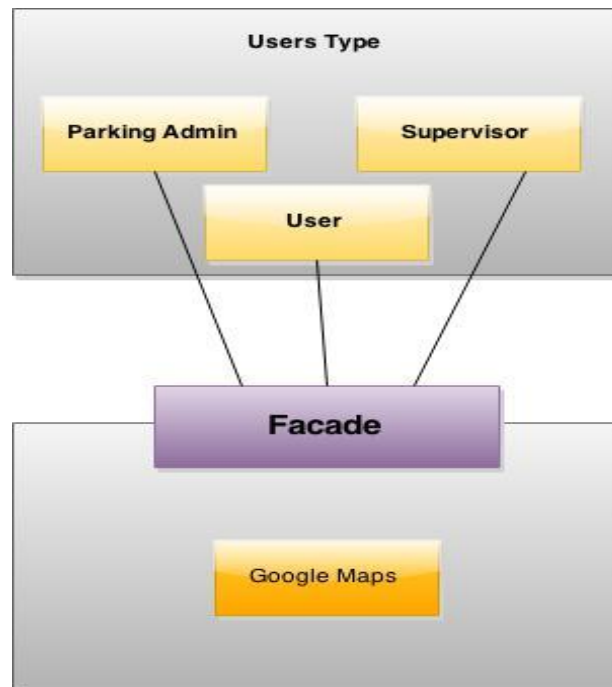


22/05/2015

Όνομα προτύπου : Facade Design Pattern**Ταξινόμια :** Πρότυπα Συμπεριφοράς

Πρόβλημα που αντιμετωπίστηκε: Η ταυτόχρονη προσπάθεια επικοινωνίας των χρηστών με το σύστημα googlemaps μπορεί να οδηγήσει στην κακή χρήση του συστήματος. Μ αυτό το τρόπο δίνουμε την δυνατότητα στο υποσύστημα να αποφασίσει ακριβώς τον τρόπο με τον οποίο θα πραγματοποιηθεί η προσπέλαση του.

Υποσυστήματα που αναδιοργανώθηκαν: Το υποσύστημα που αναδιοργανώθηκε είναι το google maps το οποίο όπως προαναφέρθηκε και στην προηγούμενη παράγραφο έχει πλέον την δυνατότητα να αποφασίζει μόνο του τον ακριβές τρόπο με τον οποίο θα εκτελείται η προσπέλαση του και με αυτό το τρόπο να ανταποκρίνεται στις διαφορετικές απαιτήσεις του χρήστη.



Σχήμα 14 - Facade Design Pattern

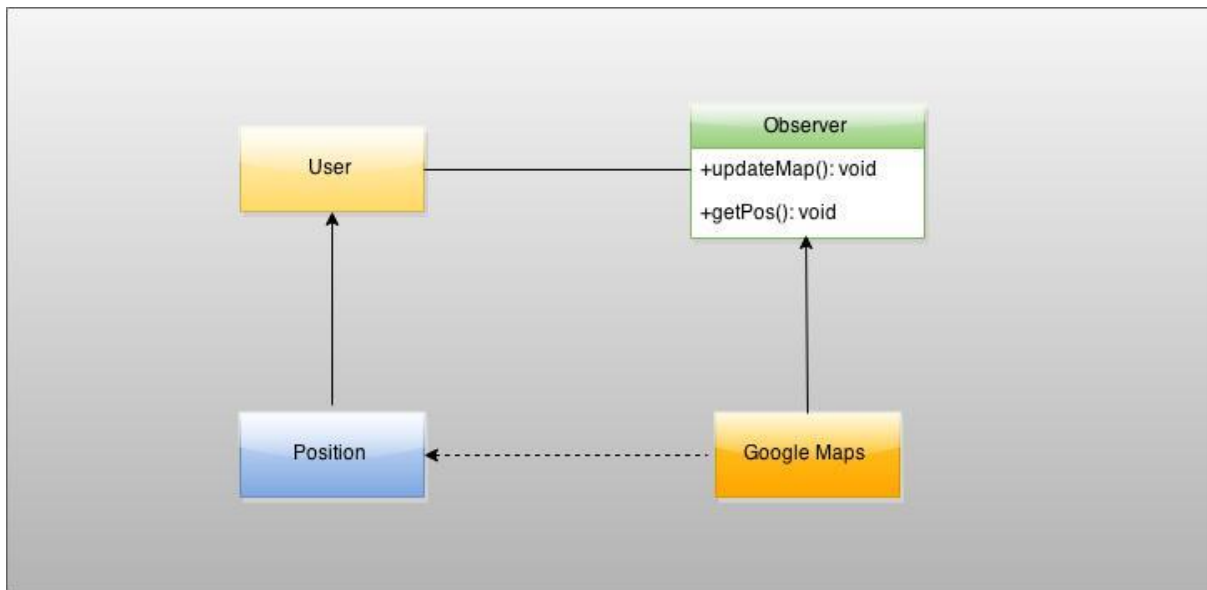


22/05/2015

Όνομα προτύπου : Observer Design Pattern**Ταξινόμια :** Πρότυπα Συμπεριφοράς

Πρόβλημα που αντιμετωπίστηκε: Ένα βασικό ζήτημα της εφαρμογής του οποίου οι εναλλαγές στην κατάσταση είναι αμέτρητες κατά την λειτουργία του συστήματος είναι η ανανέωση του διαδραστικού χάρτη με βάση τις αλλαγές της τοποθεσίας του χρήστη. Επειδή αυτό είναι ένα δυναμικό πρόβλημα που αλλάζει επανειλημμένα κατάσταση κρίθηκε αναγκαία η αναδιαμόρφωση του, με σκοπό την ανεξαρτησία μεταξύ των εμπλεκόμενων κλάσεων και την ευκολότερη εποπτείας της λειτουργίας του.

Υποσυστήματα που αναδιοργανώθηκαν: Στο σύστημα μας προστέθηκε ένα καινούργιο πρότυπο σχεδίασης τύπου Observer η οποία παρέχει την δυνατότητα στο σύστημα μας να κάνει συνεχείς ενημέρωση θέσεως του χρήστη ανάλογα με την τρέχουσα θέση του μέσω του συστήματος του googlemaps.



Σχήμα 15 - Observer Design Pattern



22/05/2015

Παράρτημα Ι – Ανοιχτά Θέματα

Το θέμα που πρέπει να συζητηθεί με τον πελάτη πριν την οριστικοποίηση Εγγράφου Σχεδίασης Συστήματος είναι η τιμολογιακή πολιτική του συστήματος.

Η τιμολόγηση του λογισμικού αναμφισβήτητα θα είναι συνδεδεμένη με το μέγεθος του έργου. Η τελική απόφαση μπορεί να βασιστεί είτε σε μοντέλο τιμολόγησης άλλων αντίστοιχων λογισμικών, είτε μέσα από γνώμη οικονομολόγων σε ιστοσελίδες διαδικτύου αντίστοιχου εύρους ανάπτυξης.

Το έργο θα μπορούσε να επιχορηγηθεί από το Δήμο Θεσσαλονίκης αφού επηρεάζει άμεσα τις συνθήκες ζωής των πολιτών και της ευρύτερης κοινωνίας. Με αυτό το έργο ολόκληρη η πόλη θα άλλαζε προς το καλύτερο και θα συνιστούσε στην ανάπτυξη του βιοτικού επιπέδου των δημοτών.

Στο μέλλον προσδοκούμε στην πλήρη χαρτογράφηση των δημόσιων χώρων στάθμευσης επί των οδών. Για να επιτευχθεί αυτός ο στόχος όμως, ο Δήμος Θεσσαλονίκης θα πρέπει να αναβαθμίσει το συγκοινωνιακό δίκτυο της πόλης αλλά και να καταρτιστεί με τον κατάλληλο εξοπλισμό ελέγχου των χώρων.