**ΤΕΙ ΠΕΙΡΑΙΑ-ΤΜΗΜΑ ΗΥΣ**

**Εργαστήριο Μηχανικής Λογισμικού**

**Εξαμηνιαία εργασία**

****

**Επιμέλεια Εργασίας:**

|  |  |
| --- | --- |
| **Ονοματεπώνυμο** | **Α.Μ** |
| Άρης Κύρκος | 39438 |
| Δημήτρης Μαυρογιώργος | 39285 |
| Μανώλης Χρυσόστομος | 32637 |

Περιεχόμενα

[1. Έντυπο Προδιαγραφών Απαιτήσεων Λογισμικού 4](#_Toc377344174)

[Εισαγωγή 4](#_Toc377344175)

[1.1 Σκοπός του κειμένου 4](#_Toc377344176)

[1.2 Προορισμένο ακροατήριο και προτάσεις ανάγνωσης 4](#_Toc377344177)

[1.3 Στόχος και εύρος του προϊόντος 4](#_Toc377344178)

[1.4 Αναφορές 5](#_Toc377344179)

[2. Γενική Περιγραφή 5](#_Toc377344180)

[2.1 Προέλευση και προοπτική του προϊόντος 5](#_Toc377344181)

[2.2 Γενική περιγραφή των λειτουργιών του προϊόντος 5](#_Toc377344182)

[2.3 Κατηγορίες χρηστών 5](#_Toc377344183)

[2.4 Περιβάλλον λειτουργίας 5](#_Toc377344184)

[2.5 Περιορισμοί στη σχεδίαση και την υλοποίηση 5](#_Toc377344185)

[2.6 Τεκμηρίωση για το χρήστη 6](#_Toc377344186)

[3. Απαιτήσεις για τις εξωτερικές διεπαφές 6](#_Toc377344187)

[3.1 Διεπαφή χρήστη 6](#_Toc377344188)

[3.2 Διεπαφές υλικού 6](#_Toc377344189)

[3.3 Διεπαφές λογισμικού 6](#_Toc377344190)

[3.4 Διεπαφές επικοινωνιών 6](#_Toc377344191)

[4. Λειτουργίες συστήματος 7](#_Toc377344192)

[Λειτουργία 1 7](#_Toc377344193)

[Λειτουργία 2 7](#_Toc377344194)

[Λειτουργία 3 8](#_Toc377344195)

[5. Μη λειτουργικές απαιτήσεις 8](#_Toc377344196)

[5.1 Επιδόσεις 8](#_Toc377344197)

[5.2 Ασφάλεια πληροφορίας 8](#_Toc377344198)

[2. Περιπτώσεις Χρήσης 9](#_Toc377344199)

[Διάγραμμα Περιπτώσεων Χρήσης 9](#_Toc377344200)

[Κατάλογος Περιπτώσεων Χρήσης 10](#_Toc377344201)

[Πρότυπο Περιπτώσεων Χρήσης 11](#_Toc377344202)

[3. Διαγράμματα UML 20](#_Toc377344203)

[Διάγραμμα κλάσεων 20](#_Toc377344204)

[Διαγράμματα ακολουθίας 21](#_Toc377344205)

[Διαγράμματα Δραστηριότητας 24](#_Toc377344206)

[4. Αποτελέσματα PMD Static analysis 26](#_Toc377344207)

[5. Unit testing – Code coverage 29](#_Toc377344208)

[6. GUI – Screenshots 30](#_Toc377344209)

[7. Σενάρια για User Acceptance 34](#_Toc377344210)

[8. Παράρτημα Source Code 35](#_Toc377344211)

[Class Anafores 35](#_Toc377344212)

[Class AnefodiasmoiAdd 36](#_Toc377344213)

[Class DiaxiristisCard 38](#_Toc377344214)

[Class IpalilosCard 40](#_Toc377344215)

[Class Login 42](#_Toc377344216)

[Class Mysql 45](#_Toc377344217)

[Class OdigoiAdd 51](#_Toc377344218)

[Class OdigoiList 56](#_Toc377344219)

[Class OdigosCard 58](#_Toc377344220)

[Class OximataAdd 60](#_Toc377344221)

[Class OximataList 63](#_Toc377344222)

[Class AllTests 66](#_Toc377344223)

[Class AnaforesTest 67](#_Toc377344224)

[Class AnefodiasmoiAddTest 67](#_Toc377344225)

[Class IpalilosCardTest 68](#_Toc377344226)

[Class LoginTest 69](#_Toc377344227)

[Class OdigoiAddTest 70](#_Toc377344228)

[Class OdigosCardTest 71](#_Toc377344229)

[Class OximataAddTest 71](#_Toc377344230)

# 1. Έντυπο Προδιαγραφών Απαιτήσεων Λογισμικού

για το Έργο

Συστήματος Επίβλεψης Στόλου Λεωφορείων

Από

Άρης Κύρκος

Δημήτρης Μαυρογιώργος

Μανώλης Χρυσόστομος

12/01/2014

**Αναθεωρήσεις**

|  |  |  |  |
| --- | --- | --- | --- |
| **Συγγραφέας** | **Ημερομηνία** | **Αίτια αναθεωρήσεων** | **Έκδοση κειμένου** |
| Άρης Κύρκος | 12/01/2014 |  | 1.0 |
| Δημήτρης Μαυρογιώργος | 12/01/2014 |  | 1.0 |
| Μανώλης Χρυσόστομος | 12/01/2014 |  | 1.0 |

## Εισαγωγή

### 1.1 Σκοπός του κειμένου

*Αυτό το SRS περιγράφει το λογισμικό, τις λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος επίβλεψης στόλου λεωφορείων. Το έγγραφο αυτό προορίζεται να χρησιμοποιηθεί από τα μέλη της ομάδας έργου που θα υλοποιήσει και να εξακριβώσει την ορθή λειτουργία του συστήματος.*

### 1.2 Προορισμένο ακροατήριο και προτάσεις ανάγνωσης

*Το παρών έγγραφο απευθύνεται στον πρόεδρο και τα υψηλόβαθμα στελέχη της εταιρίας καθώς και στο τμήμα IT. Επίσης από τη μεριά μας απευθύνεται στον πρόεδρο, τον Project Manager όπως και στο τμήμα Development.*

### 1.3 Στόχος και εύρος του προϊόντος

Το σύστημα επίβλεψης στόλου λεωφορείων θα προσφέρει στην εταιρία μια εύχρηστη και αξιόπιστη διαχείριση και επίβλεψη του στόλου των λεωφορείων, μειώνοντας τα λειτουργικά έξοδα και τη γραφειοκρατία, κάνοντας την εργασία πιο εύκολη. Περισσότερες πληροφορίες βρίσκονται στα επόμενα κεφάλαια.

### 1.4 Αναφορές

* Manual συσκευής αποστολής δεδομένων GPS
* Manual H/Y
* Manual Server

## 2. Γενική Περιγραφή

### 2.1 Προέλευση και προοπτική του προϊόντος

*Το λογισμικό για τη επίβλεψη στόλου λεωφορείων αποτελεί μια εφαρμογή που ως βασικό στόχο έχει την διαχείριση και την επίβλεψη του στόλου των λεωφορείων. Το λογισμικό μπορεί να αναβαθμιστεί εάν αυτό κριθεί απαραίτητο στο μέλλον ως αποτέλεσμα της καλύτερης απόδοσης λογισμικού και την πάροδο της τεχνολογίας ή την αύξηση των απαιτήσεων της εταιρείας.*

### 2.2 Γενική περιγραφή των λειτουργιών του προϊόντος

*Το λογισμικό θα έχει τις εξής λειτουργίες:*

* *Καταγραφή και επεξεργασία στοιχείων των οδηγών και των οχημάτων*
* *Λήψη πληροφοριών σχετικά με τα* λεωφορεία
* *Επίβλεψη* λεωφορείων *και οδηγών*
* *Ζωντανή παρακολούθηση τοποθεσίας* λεωφορείων
* *Καταγραφή αναφορών για τα* λεωφορεία
* *Καταγραφή ανεφοδιασμών λεωφορείων*
* *Καταγραφή στατιστικών για τα λεωφορεία*

### 2.3 Κατηγορίες χρηστών

*Οι κατηγορίες των χρηστών που θα μπορούν να χρησιμοποιούν το λογισμικό είναι:*

* *Οι οδηγοί*
* *Οι υπάλληλοι γραφείου*
* *Ο προϊστάμενος*

### 2.4 Περιβάλλον λειτουργίας

Το λογισμικό θα τρέχει σε υπολογιστές με λειτουργικό σύστημα Windows 7, επεξεργαστή i3(και παραπάνω i5,i7) και μνήμη RAM 4GB.

Η βάση δεδομένων θα τρέχει σε Sever με λειτουργικό σύστημα Windows 7 και ταχύτητα επεξεργαστή 3.4GHz.

Τα εγχειρίδια χρήσης (Manuals) των H/Υ και του Server έχουν δοθεί και υπάρχουν αναφορές τους στην παράγραφο 1.4.

### 2.5 Περιορισμοί στη σχεδίαση και την υλοποίηση

Περιορισμοί αποτελούν το λειτουργικό σύστημα (Windows 7) καθώς επίσης και το υλικό που είναι εγκατεστημένο στους Η/Υ της εταιρίας. Επίσης περιορισμός στη σχεδίαση και υλοποίηση του λογισμικού αποτελεί η χρήση της MySQL ως βάση δεδομένων και η ύπαρξη Server, στον οποίο θα εγκατασταθεί η βάση δεδομένων και θα εξυπηρετεί πολλούς χρήστες ταυτόχρονα. Τέλος η χρήση κρυπτογραφημένης γραμμής SSL θα διασφαλίζει την κρυπτογραφημένη τοποθέτηση των δεδομένων τόσο στην διεπαφή του χρήστη όσο και στην εισαγωγή δεδομένων στη βάση δεδομένων.

Το λογισμικό θα δουλεύει με τουλάχιστον 100 ασύγχρονους χαρακτήρες καθώς επίσης και όλες οι ενέργειες θα γίνονται σε κάτω από 10 δευτερόλεπτα.

### 2.6 Τεκμηρίωση για το χρήστη

Μαζί με το λογισμικό θα παραδοθεί:

1. Εγχειρίδιο χρήστη

## 3. Απαιτήσεις για τις εξωτερικές διεπαφές

### 3.1 Διεπαφή χρήστη

*Η διεπαφή με το χρήστη θα γίνεται μέσω παραθύρων. Σε καρτέλες λιστών θα εμφανίζονται οι πληροφορίες για τους οδηγούς, για τα οχήματα και η καταγραφή των ανεφοδιασμών των οχημάτων. Σε παράθυρο θα μπορούμε να εντοπίζουμε την κίνηση των οχημάτων και εύκολα θα μπορούμε να βλέπουμε τα δεδομένα που αφορούν το όχημα και τον οδηγό.*

### 3.2 Διεπαφές υλικού

*Μια συσκευή τοποθετημένη στο όχημα στέλνει γενικές πληροφορίες για τη θέση του και για το ίδιο μέσω δικτύου GPS/GPRS στο λογισμικό. Το Manual της συσκευής αυτής και υπάρχει αναφορά του στην παράγραφο 1.4.*

### 3.3 Διεπαφές λογισμικού

Η εφαρμογή θα ανταλλάζει δεδομένα με μία MySQL Database η οποία θα είναι εγκατεστημένη στον Server της εταιρίας.

### 3.4 Διεπαφές επικοινωνιών

Στην εταιρία υπάρχει εγκατεστημένη γραμμή του 1Gb/s downstream. Τέλος πρέπει να γίνεται χρήση κρυπτογράφησης SSL.

## 4. Λειτουργίες συστήματος

### Λειτουργία 1

4.1.1 Περιγραφή και προτεραιότητα

Η πρώτη λειτουργία είναι η αποστολή δεδομένων για τους οδηγούς και τα οχήματα στην Βάση δεδομένων.

4.1.2 Ακολουθίες ερεθίσματος / απόκρισης

Οι χρήστες θα μπορούν να κάνουν εισαγωγή, διαγραφή, επεξεργασία καινούργιων οδηγών και οχημάτων στο σύστημα βάσης δεδομένων μέσω της εφαρμογής μέσω παραθύρων από όπου θα βλέπουν όλο το εργατικό δυναμικό σε οδηγούς και θα διαχειρίζονται σε ποιο λεωφορείο θα δουλέψει ο καθένας από τους οδηγούς. Η εισαγωγή θα γίνεται μέσα σε ένα παράθυρο όπου θα υπάρχουν κατάλληλα textboxes θα συμπληρώνονται και μετά με το πάτημα ενός button θα εισάγονται στη βάση δεδομένων. Ακόμη το όχημα θα διαθέτει και έναν αισθητήρα GPS/GPRS από όπου θα μπορεί κάποιος να βρίσκει την τοποθεσία του εκείνη την στιγμή.

4.1.3 Ανάλυση σε λειτουργικές απαιτήσεις

*Οι χρήστες θα μπορούν στο ένα παράθυρο να επεξεργάζονται να διαγράφουν και να εισάγουν στοιχεία που αφορούν το όχημα καθώς επίσης και να επιβλέπουν που είναι το όχημα ζωντανά.*

*Τα στοιχεία που επεξεργάζονται οι χρήστες για το όχημα είναι :*

* *Κωδικός Οχήματος*
* *Διαδρομή*

Ακόμη οι χρήστες μπορούν να τροποποιούν, διαγράφουν, εισάγουν δεδομένα που αφορούν τους οδηγούς όπως:

* *Όνομα*
* *Επώνυμο*
* *Username*
* *Password*
* *Όχημα που οδηγούν*

### Λειτουργία 2

4.2.1 Περιγραφή και προτεραιότητα

Η δεύτερη λειτουργία είναι η προσθήκη ανεφοδιασμού και η προβολή αναφορών του οχήματος.

4.2.2 Ακολουθίες ερεθίσματος / απόκρισης

*Οι χρήστες θα μπορούν να κάνουν τις προσθήκες των ανεφοδιασμών για το κατάλληλο λεωφορείο με βενζίνη καθώς και να βλέπουν τις αναφορές σχετικά με το όχημα .*

4.2.3 Ανάλυση σε λειτουργικές απαιτήσεις

*Μέσω επιλογής ο χρήστης καταλήγει σε ένα παράθυρο που μπορεί να κάνει προσθήκη ανεφοδιασμού σε λίτρα για το όχημα που επιλέγει στην βάση δεδομένων καθώς επίσης σε νέο παράθυρο θα μπορεί να βλέπει τις αναφορές για το κάθε όχημα τις οποίες θα γίνονται μέσω ερώτησης στην βάση δεδομένων .*

### Λειτουργία 3

4.3.1 Περιγραφή και προτεραιότητα

Η τρίτη λειτουργία είναι η αναφορά προβλήματος για το λεωφορείο.

4.3.2 Ακολουθίες ερεθίσματος / απόκρισης

*Ο χρήστης επιλεγέντας το λεωφορείο θα μπορεί να αποστείλει αναφορές σχετικά με το όχημα π.χ. εάν παρουσιάστηκε κάποιο σφάλμα η αν χρειάζεται service.*

4.3.3 Ανάλυση σε λειτουργικές απαιτήσεις

Ο χρήστης μέσω ενός παραθύρου θα μπορεί να αναφέρει το πρόβλημα του σχετικά με το λεωφορείο στο οποίο εργάζεται και να κάνει και τα σχετικά σχόλια για αυτό.

## 5. Μη λειτουργικές απαιτήσεις

### 5.1 Επιδόσεις

Η απαίτηση για επίδοση υπάρχει στην ζωντανή λήψη πληροφοριών του οχήματος καθώς και τη θέση του, ώστε να υπάρχει η δυνατότητα έκδοσης ακριβών στατιστικών στοιχείων και ενημέρωση των χρηστών.

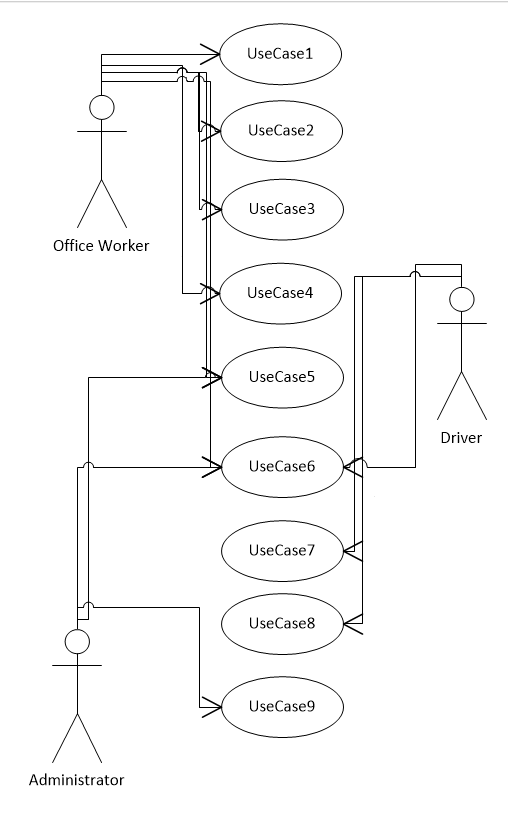
### 5.2 Ασφάλεια πληροφορίας

*Θα πρέπει να υπάρχει πιστοποίηση των χρηστών που θα χρησιμοποιούν το λογισμικό μέσω κωδικών πρόσβασης κατά την είσοδο τους σε αυτό.*

*Τέλος όπως αναφέρθηκε και στην παράγραφο 3.4 θα χρησιμοποιείται κρυπτογραφημένη γραμμή SSL.*

# 2. Περιπτώσεις Χρήσης

## Διάγραμμα Περιπτώσεων Χρήσης

****

## Κατάλογος Περιπτώσεων Χρήσης

|  |  |
| --- | --- |
| ***Εμπλεκόμενοι Ρόλοι*** | ***Περιπτώσεις Χρήσης*** |
| Υπάλληλος | UC-1. Είσοδος στη εφαρμογή με δικαιώματα Υπαλλήλου γραφείου  UC-2. Προσθήκη, επεξεργασία και προβολή οδηγών  UC-3. Προσθήκη, επεξεργασία και προβολή λεωφορείων, προβολή θέσης λεωφορείου.  UC-4. Προσθήκη Ανεφοδιασμού  UC-5. Προβολή Αναφορών  UC-6. Έξοδος από την εφαρμογή |
| Διαχειριστής | UC-7. Είσοδος στην εφαρμογή με δικαιώματα Διαχειριστή  UC-8. Προβολή αναφορών.  UC-6. Έξοδος από την εφαρμογή |
| Οδηγός | UC-9. Είσοδος στην εφαρμογή με δικαιώματα Οδηγού  UC-5. Προσθήκη αναφορών.  UC-6. Έξοδος από την εφαρμογή |

## Πρότυπο Περιπτώσεων Χρήσης

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-1 | | |
| Ονομασία: | Είσοδος στη εφαρμογή με δικαιώματα Υπαλλήλου γραφείου | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο Υπάλληλος γραφείου θέλει να συνδεθεί με την εφαρμογή. |
| Γεγονός Εκκίνησης: | Βγαίνει το παράθυρο για να πληκτρολογήσει το username & password |
| Προϋποθέσεις: | 1. Το username & password πρέπει να έχουν μπει στην βάση δεδομένων . |
| Τελική Κατάσταση: | 1. Είσοδος στο πρόγραμμα |
| Φυσιολογική Ροή: | 1. Εισαγωγή του username & password στο παράθυρο login 2. Είσοδος στην εφαρμογή . |
| Εναλλακτική Ροή: | 1. Εισαγωγή του username & password στο παράθυρο login 2. Σφάλμα μη έγκυρο username ή password στο παράθυρο login 3. Προσπαθήστε ξανά |
| Εξαιρέσεις: | 1.0 Πρόβλημα με το username & password επειδή δεν το έχουν βάλει στη βάση δεδομένων.  2.0 Πρόβλημα σύνδεσης με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-2 | | |
| Ονομασία: | Προσθήκη, επεξεργασία και προβολή οδηγών | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος γραφείου |
| Περιγραφή: | Ο Υπάλληλος γραφείου επιλέγοντας από το μενού την ιδιότητα ¨ Οδηγοί ¨ μπορεί να έχει πλήρη εικόνα της κατάστασης του οδηγού το οποίο επίλεξε . |
| Γεγονός Εκκίνησης: | Βγαίνει το μενού επιλογών . |
| Προϋποθέσεις: | 1. Ο υπάλληλος πρέπει να έχει κάνει login . |
| Τελική Κατάσταση: | 1. Πίνακας δεδομένων σχετικά με τους οδηγούς που έχουμε επιλέξει. |
| Φυσιολογική Ροή: | 1.Επιλογή του οδηγού που θέλουμε μέσο μενού από την βάση δεδομένων .   1. Επιστροφή αποτελεσμάτων στην οθόνη μέσο πίνακα . 2. Επεξεργασία του επιλεγμένου οδηγού. 3. Εγγραφή στην βάση δεδομένων του συγκεκριμένου οδηγού. 4. Προσθήκη νέου οδηγού 5. Εγγραφή στην βάση δεδομένων |
| Εναλλακτική Ροή: | 1. Μη συμπληρώσει των απαραίτητων στοιχείων για τον νέο οδηγό. 2. Μήνυμα λάθους. 3. Διόρθωση κατάλληλων πεδίων. |
| Εξαιρέσεις: | * 1. Πρόβλημα με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-3 | | |
| Ονομασία: | Προσθήκη, επεξεργασία και προβολή λεωφορείων και προβολή θέσης λεωφορείου. | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος γραφείου |
| Περιγραφή: | Ο Υπάλληλος γραφείου επιλέγοντας από το μενού την ιδιότητα ¨ Οχήματα ¨ μπορεί να έχει πλήρη εικόνα της κατάστασης του οχήματος το οποίο επίλεξε . |
| Γεγονός Εκκίνησης: | Βγαίνει το μενού επιλογών . |
| Προϋποθέσεις: | 1. Ο υπάλληλος πρέπει να έχει κάνει login . |
| Τελική Κατάσταση: | 1. Πίνακας δεδομένων σχετικά με τα λεωφορεία που έχουμε επιλέξει. |
| Φυσιολογική Ροή: | 1. Επιλογή του λεωφορείου που θέλουμε μέσο μενού από την βάση δεδομένων . 2. Επιστροφή αποτελεσμάτων στην οθόνη μέσο πίνακα . 3. Επεξεργασία του επιλεγμένου λεφορείου. 4. Εγγραφή στην βάση δεδομένων του συγκεκριμένου λεωφορείου. 5. Προσθήκη νέου λεωφορείου. 6. Εγγραφή στην βάση δεδομένων. 7. Προβολή θέσης λεωφορείου μέσω GPS/GPRS που δείχνει την θέση του λεωφορείου στον χάρτη. |
| Εναλλακτική Ροή: | 1. Μη συμπληρώσει των απαραίτητων στοιχείων για τον νέο λεωφορείο. 2. Μήνυμα λάθους. 3. Διόρθωση κατάλληλων πεδίων. |
| Εξαιρέσεις: | * 1. Πρόβλημα με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-4 | | |
| Ονομασία: | Προσθήκη ανεφοδιασμού | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος γραφείου |
| Περιγραφή: | Ο Υπάλληλος γραφείου επιλέγοντας από το μενού την ιδιότητα ¨ Προσθήκη Ανεφοδιασμου ¨ μπορεί να κάνει προσθήκη ανεφοδιασμού σε λίτρα στο κατάλληλο όχημα . |
| Γεγονός Εκκίνησης: | Βγαίνει το μενού επιλογών . |
| Προϋποθέσεις: | 1. Ο υπάλληλος πρέπει να έχει κάνει login . |
| Τελική Κατάσταση: | 1. Πίνακας δεδομένων σχετικά με τους ανεφοδιασμούς στο εκάστοτε όχημα. |
| Φυσιολογική Ροή: | 1. Επιλογή του οχήματος που θέλουμε μέσο dropdown list από την βάση δεδομένων. 2. Προσθήκη αριθμού λίτρων . 3. Σώσιμο δεδομένων στην βάση. |
| Εναλλακτική Ροή: | 1. Μη συμπληρώσει του αριθμού λίτρων . 2. Μήνυμα λάθους. 3. Μη συμπληρώσει με ακέραιο στο πεδίο Λίτρα. 4. Μήνυμα λάθους. 5. Διόρθωση. 6. Εισαγωγή στην βάση δεδομένων. |
| Εξαιρέσεις: | * 1. Πρόβλημα με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-5 | | |
| Ονομασία: | Προβολή Αναφορών | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος γραφείου |
| Περιγραφή: | Ο Υπάλληλος γραφείου επιλέγοντας από το μενού την ιδιότητα ¨ Αναφορές¨ μπορεί να κάνει προβολή αναφορών για τα οχήματα. |
| Γεγονός Εκκίνησης: | Βγαίνει το μενού επιλογών . |
| Προϋποθέσεις: | 1. Ο υπάλληλος πρέπει να έχει κάνει login . |
| Τελική Κατάσταση: | 1. Πίνακας δεδομένων σχετικά με τις αναφορές . |
| Φυσιολογική Ροή: | 1. Προβολή αναφορών σχετικά με τα οχήματα. |
| Εναλλακτική Ροή: | 1. Πρόβλημα με την βάση δεδομένων. 2. Μήνυμα λάθους. |
| Εξαιρέσεις: | 2.1 Πρόβλημα με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-6 | | |
| Ονομασία: | Έξοδος από την εφαρμογή. | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος γραφείου, Οδηγοί, Διαχειριστής. |
| Περιγραφή: | Ο Υπάλληλος γραφείου, οδηγός, διαχειριστής κλείνοντας όλα τα παράθυρα, ουσιαστικά τερματίζει την εφαρμογή. |
| Γεγονός Εκκίνησης: | Βγαίνει το μενού επιλογών. |
| Προϋποθέσεις: | 1. Ο χρήστης πρέπει να έχει κάνει login . |
| Τελική Κατάσταση: | 1. Έξοδος από την εφαρμογή . |
| Φυσιολογική Ροή: | 1. Έξοδος από την εφαρμογή. |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: | - |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-7 | | |
| Ονομασία: | Είσοδος στην εφαρμογή με δικαιώματα διαχειριστή. | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Διαχειριστής |
| Περιγραφή: | Ο Διαχειριστής θέλει να συνδεθεί στην εφαρμογή. |
| Γεγονός Εκκίνησης: | Βγαίνει το παράθυρο για να πληκτρολογήσετε το username και password . |
| Προϋποθέσεις: | 1. Το username & password πρέπει να έχουν μπει στην βάση δεδομένων. |
| Τελική Κατάσταση: | 1. Είσοδος στο πρόγραμμα με δικαιώματα διαχειριστή. |
| Φυσιολογική Ροή: | 1. Εισαγωγή του username & password στο παράθυρο login 2. Είσοδος στην εφαρμογή . |
| Εναλλακτική Ροή: | 1. Εισαγωγή του username & password στο παράθυρο login 2. Σφάλμα μη έγκυρο username ή password στο παράθυρο login. 3. Προσπαθήστε ξανά. |
| Εξαιρέσεις: | 3.1 Πρόβλημα με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-8 | | |
| Ονομασία: | Είσοδος στην εφαρμογή με δικαιώματα οδηγού. | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

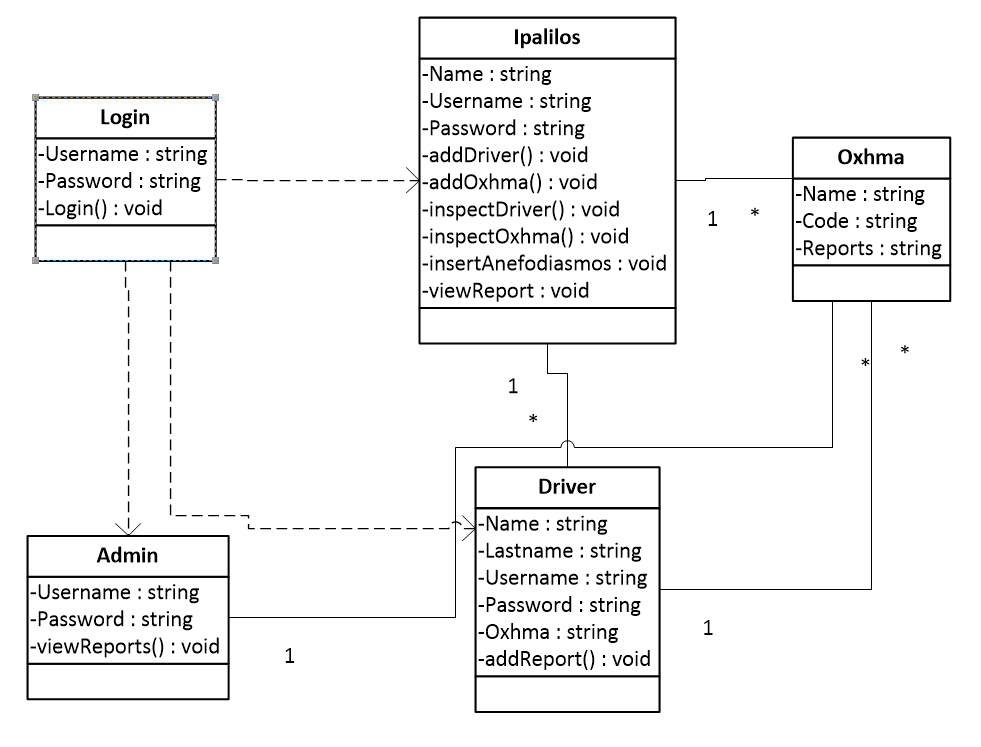
|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Οδηγός |
| Περιγραφή: | Ο Οδηγός θέλει να συνδεθεί στην εφαρμογή. |
| Γεγονός Εκκίνησης: | Βγαίνει το παράθυρο για να πληκτρολογήσετε το username και password . |
| Προϋποθέσεις: | 1. Το username & password πρέπει να έχουν μπει στην βάση δεδομένων. |
| Τελική Κατάσταση: | 1. Είσοδος στο πρόγραμμα με δικαιώματα οδηγού. |
| Φυσιολογική Ροή: | 1. Εισαγωγή του username & password στο παράθυρο login 2. Είσοδος στην εφαρμογή . |
| Εναλλακτική Ροή: | 1. Εισαγωγή του username & password στο παράθυρο login 2. Σφάλμα μη έγκυρο username ή password στο παράθυρο login. 3. Προσπαθήστε ξανά. |
| Εξαιρέσεις: | 3.1 Πρόβλημα με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-9 | | |
| Ονομασία: | Προσθήκη Αναφοράς | | |
| Δημιουργήθηκε από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* | Τελευταία ενημέρωση από: | *Άρης Κύρκος*  *Δημήτρης Μαυρογιώργος*  *Μανώλης Χρυσόστομος* |
| Ημερομηνία Συγγραφής: | *12/01/2014* | Ημερομηνία τελευταίας ενημέρωσης: | *12/01/2014* |

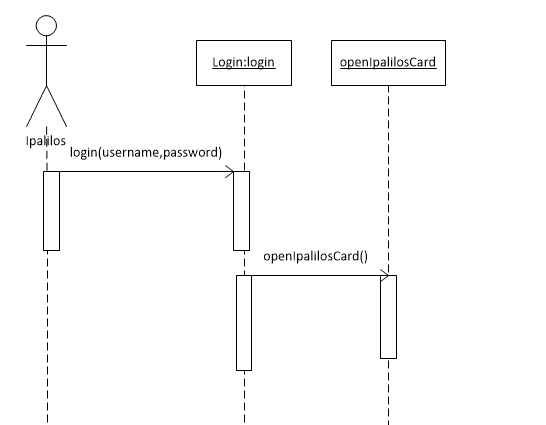
|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Οδηγός |
| Περιγραφή: | Ο Οδηγός θέλει αναφέρει ένα πρόβλημα που αντιμετώπισε με το όχημα του. |
| Γεγονός Εκκίνησης: | Βγαίνει το παράθυρο για την αναφορά προβλήματος. |
| Προϋποθέσεις: | 1. Ο Οδηγός να έχει συνδεθεί στην εφαρμογή. |
| Τελική Κατάσταση: | 1. Αναφορά προβλήματος για το εκάστοτε όχημα. |
| Φυσιολογική Ροή: | 1. Επιλογή του προβλήματος από μια dropdown list . 2. Εισαγωγή σχολίων. 3. Καταχώρηση αναφοράς στη βάση δεδομένων |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: | 3.0 Πρόβλημα σύνδεσης με την βάση δεδομένων. |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | Περίπου 250 χρήσεις ανά ημέρα |
| Business Rules: | BR-1, BR-2,BR-4 |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

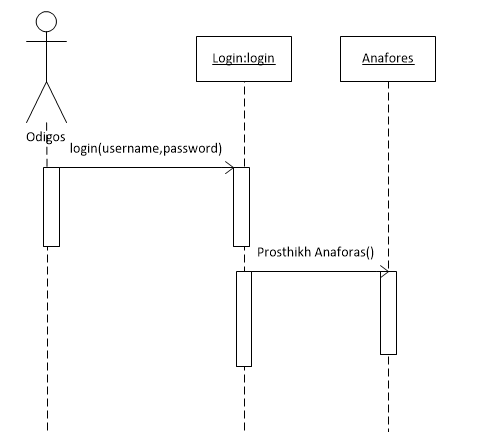
# 3. Διαγράμματα UML

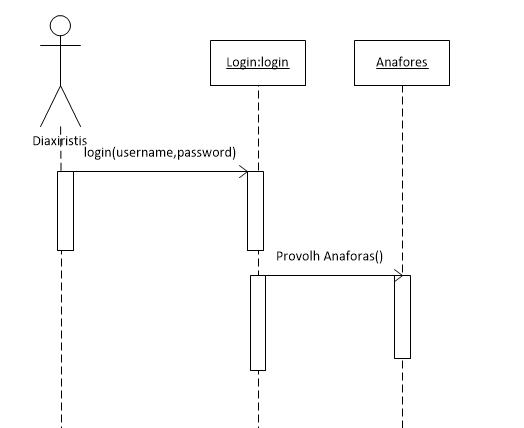
## Διάγραμμα κλάσεων



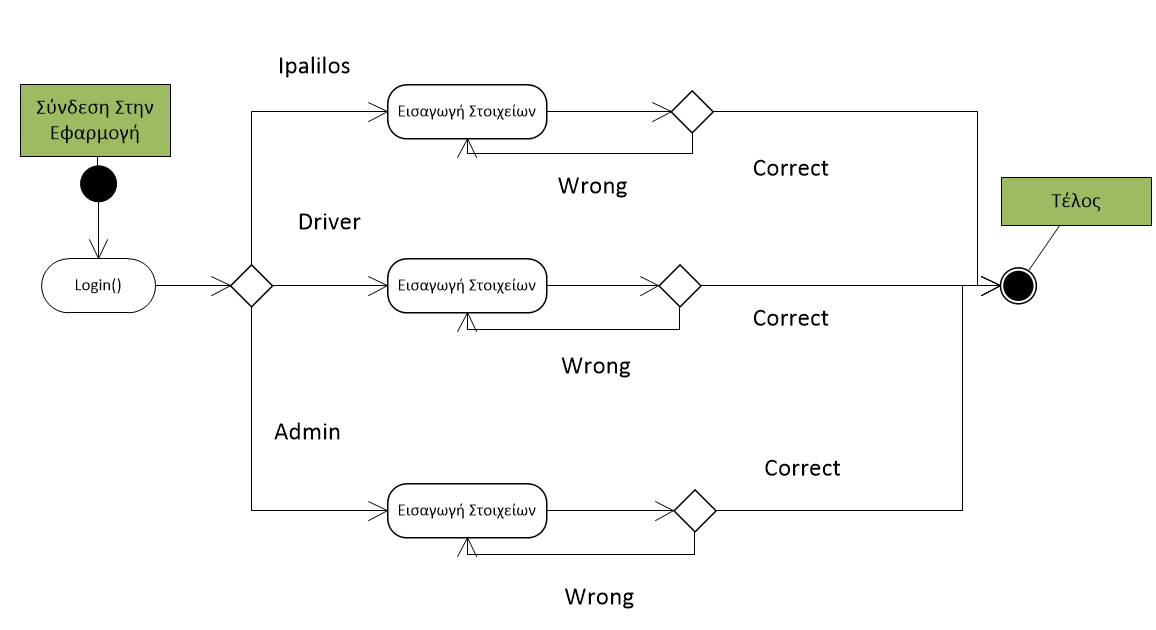
## Διαγράμματα ακολουθίας

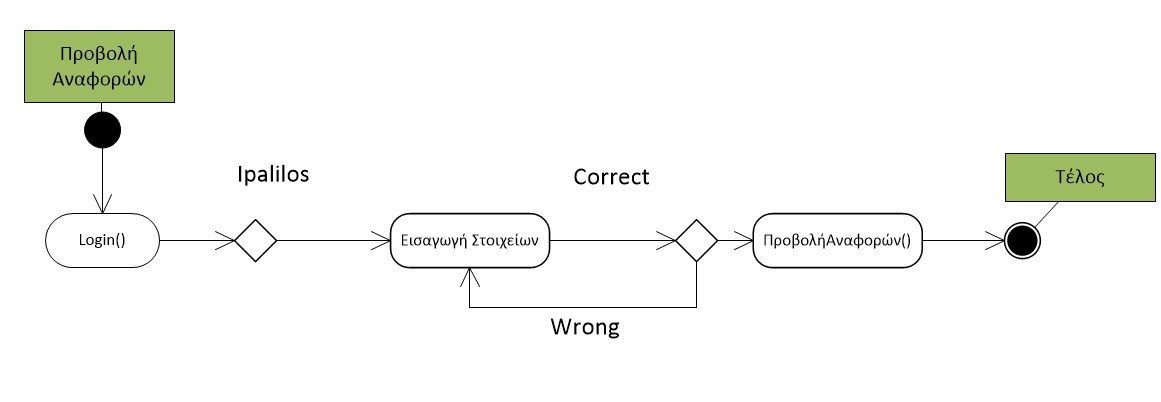


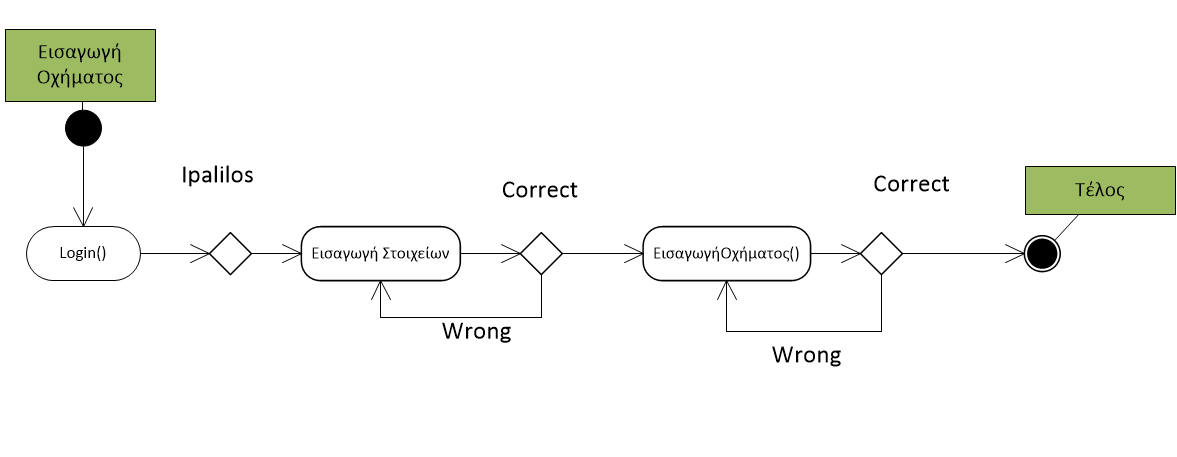




## Διαγράμματα Δραστηριότητας

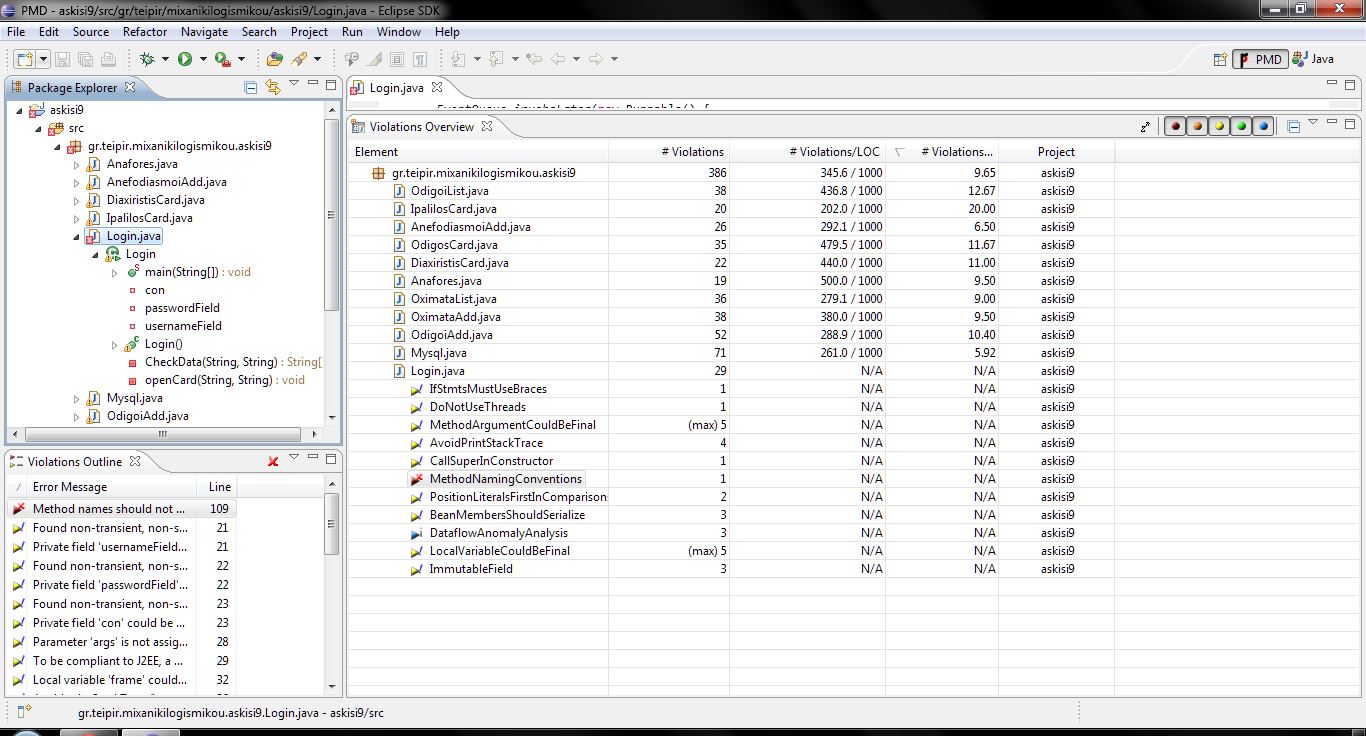


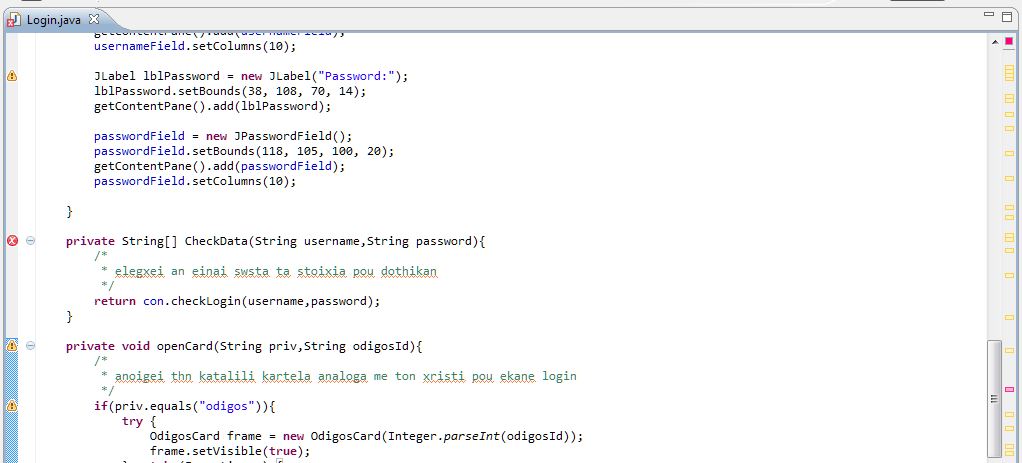




# 4. Αποτελέσματα PMD Static analysis

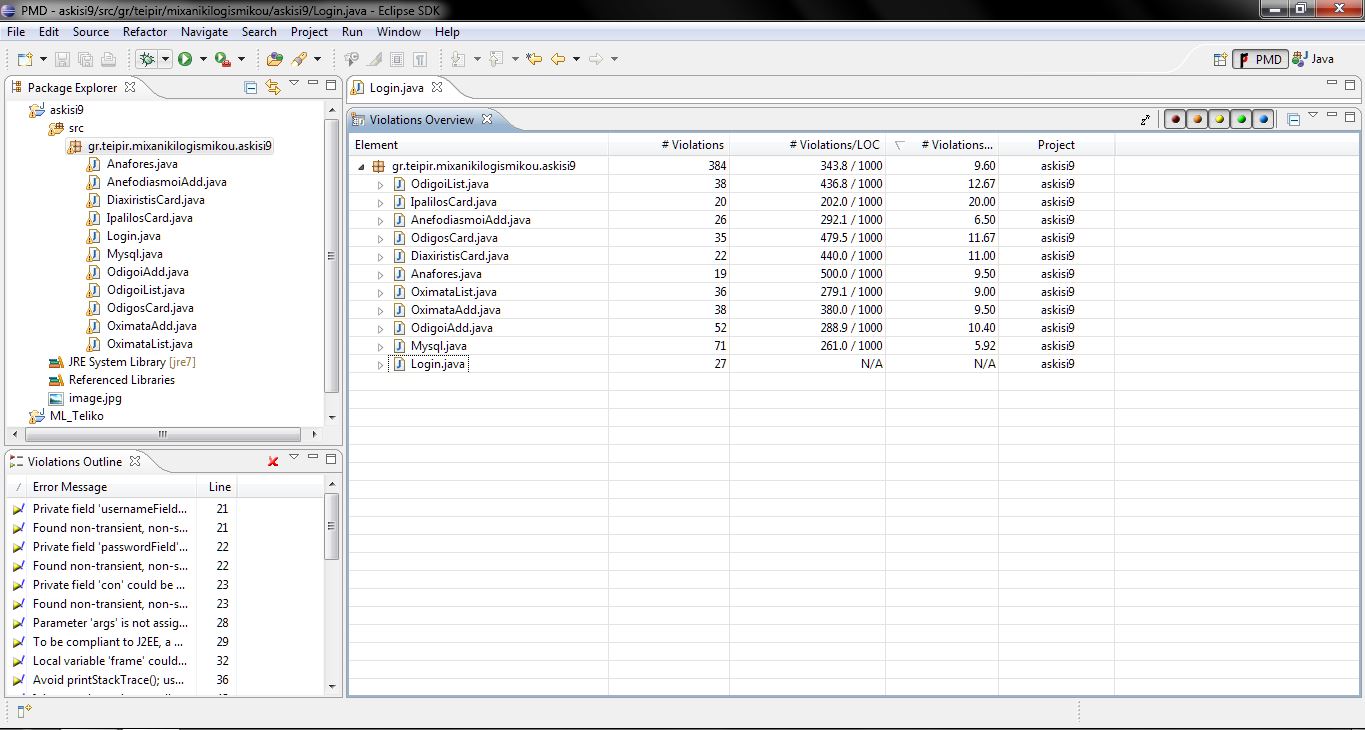
Κάνοντας την στατική ανάλυση του κώδικα μας μέσω του εργαλείου PMD μπορούμε να βρούμε εύκολα τα συντακτικά λάθη που προέκυψαν στο κώδικα μας. Τρέχοντας πρώτη φορά των κώδικά μας βλέπουμε ότι :





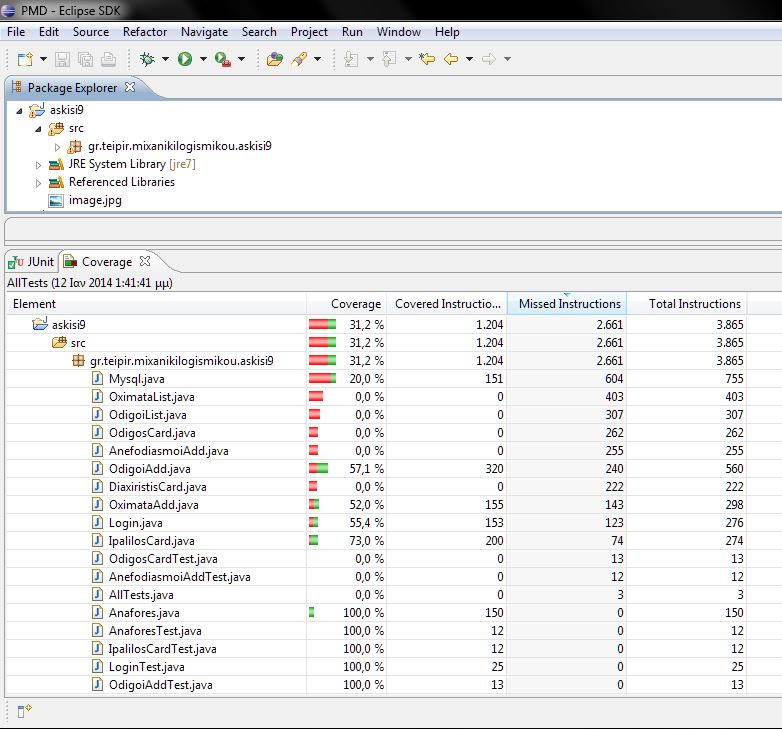
Πιο αναλυτικά βλέπουμε ότι έχουμε ένα πρόβλημα διότι έχουμε ορίσει την μέθοδο CheckData() με κεφαλαίο όμως θα έπρεπε να την είχαμε ορίσει με μικρό διότι είναι μέθοδος.

Μόλις διορθώσαμε το σφάλμα μπορούμε να δούμε ότι όλα τα test του PMD περνάνε χωρίς να ανιχνεύσουν κανένα σφάλμα.



# 5. Unit testing – Code coverage

Μόλις φτιάξαμε τα σενάρια για τα test τα οποία θέλουμε πάνω στον κώδικα μπορούμε να δούμε αναλυτικά τι επικάλυψη έχουμε στο πακέτο του κώδικα που ετοιμάσαμε.



Βλέπουμε πως υπάρχει **31.2 %** επικάλυψη στο συνολικό project.

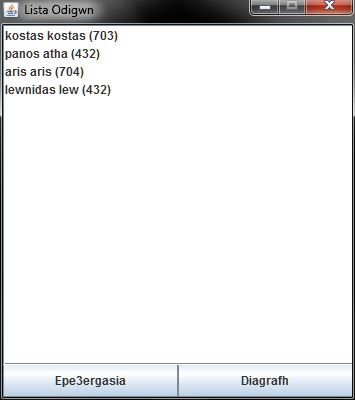
# 6. GUI – Screenshots



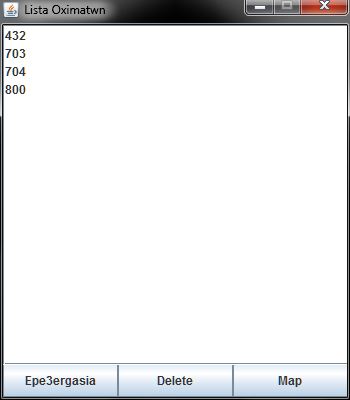
**Παράθυρο Login.**



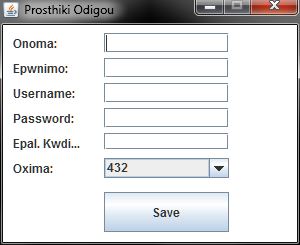
**Παράθυρο Menu Υπαλλήλου.**



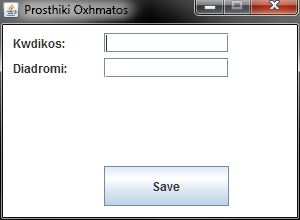
**Παράθυρο Λίστας Οδηγών.**

****

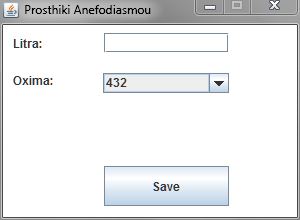
**Παράθυρο Λίστας Οχημάτων.**

****

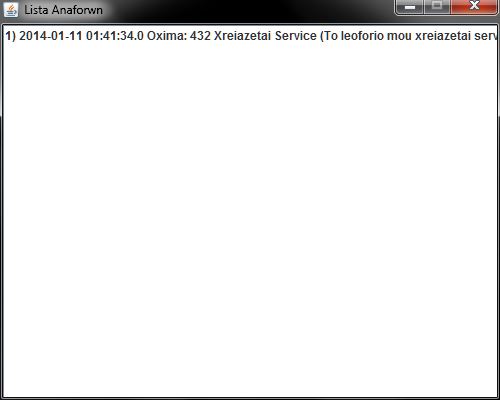
**Παράθυρο Προσθήκης Οδηγού.**

****

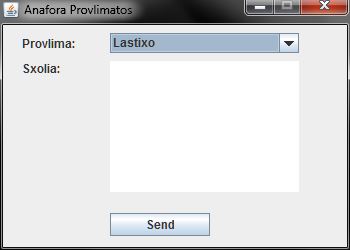
**Παράθυρο Προσθήκης Οχήματος.**

****

**Παράθυρο Προσθήκης Ανεφοδιασμού.**

****

**Παράθυρο Λίστας Αναφορών.**

****

**Παράθυρο Αναφοράς Σφάλματος.**

# 7. Σενάρια για User Acceptance

Δώσαμε την δυνατότητα στον πελάτη να ελέγξει την λειτουργικότητα, την χρηστικότητα και την δυνατότητα να μπορεί να χρησιμοποιήσει το λογισμικό που αναπτύξαμε και έτσι φτιάξαμε σε συνεργασία κάποια σενάρια χρήσης. Κάναμε εγκατάσταση του λογισμικού και της βάσης δεδομένων στους υπολογιστές της εταιρίας και είδαμε μαζί το λογισμικό να τρέχει.

Τα σενάρια που τρέξαμε είναι τα παρακάτω :

Έλεγχος αποδοχής χρήσης υπαλλήλων.

Οι υπάλληλοι της εταιρίας βάλανε τα στοιχεία username και password που λάβανε από εμάς και τα εισάγανε στην εφαρμογή.

Ελέγξανε αν τα δεδομένα είναι έγκυρα.

Εμφανίστηκε παράθυρο σφάλματος .

Στη συνέχεια εισάγανε τα σωστά δεδομένα και κάνανε είσοδο στην εφαρμογή.

Πήγανε στην επιλογή ¨Προσθήκη Οχήματος¨ και κάνανε εισαγωγή λεωφορείων στη βάση δεδομένων.

Μετά επιλέξανε την επιλογή ¨Προσθήκη Οδηγού¨ και κάνανε προσθήκη νέων οδηγών στη βάση δεδομένων καθώς και αποδώσανε και σε ποια οχήματα εργάζεται ο κάθε οδηγός.

Στη συνέχεια επιλέξανε την επιλογή ¨Προσθήκη Ανεφοδιασμού¨ , όπου και εισάγανε κάποιους ανεφοδιασμούς που έχουν γίνει εκείνη την ημέρα από τις αναφορές που πήρανε από τους οδηγούς.

Τέλος κάνανε αποσύνδεση από την εφαρμογή.

Έλεγχος αποδοχής χρήσης οδηγών.

Οι οδηγοί βάλανε τα στοιχεία τους username και password στην εφαρμογή και πραγματοποιήσανε είσοδο.

Στη συνέχεια προσθέσανε μια αναφορά από τα προβλήματα που είχανε μέσα στην ημέρα σχετικά με το λεωφορείο που εργάζονται.

Τέλος κλείσανε την εφαρμογή.

Στο σενάριο αυτό είδαμε τι γίνεται αν δεν έχουν οι οδηγοί κωδικούς (username,password) για να εισέλθουν στην εφαρμογή που οδήγησε την λειτουργία τις εφαρμογής σε μήνυμα σφάλματος εισαγωγής δεδομένων, και τέλος όσοι οδηγοί δεν εισήχθησαν στο σύστημα κάνανε εγγραφή μέσω ενός υπαλλήλου της εταιρείας.

Έλεγχος αποδοχής χρήσης Διαχειριστή.

Ο διαχειριστής έβαλε τους δικούς του κωδικούς (username, password) για την εισαγωγή του στο σύστημα και μπήκε στην εφαρμογή με δικαιώματα διαχειριστή.

Στη συνέχεια έκανε προβολή των αναφορών από το σύστημα.

Τέλος έκανε αποσύνδεση από την εφαρμογή.

# 8. Παράρτημα Source Code

### Class Anafores

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** javax.swing.DefaultListModel;

**import** javax.swing.JFrame;

**import** javax.swing.JList;

**import** javax.swing.JScrollPane;

**public** **class** Anafores **extends** JFrame {

**private** DefaultListModel<String> listModel; //lista pou tha mpoun oi anafores gia na emfanistou

**private** JList<String> anaforesList; //lista pou tha emfanizontai oi anafores

**private** ArrayList<HashMap<String, String>> anafores; //lista pou tha mpoun oi anafores me ta onomata twn sthlwn pou exoun sthn bash

**private** HashMap<String, String> anaforesInfo;

**private** Mysql con; //antikeimeno ths Mysql

**public** Anafores() {

/\*kartela anaforwn\*/

con = **new** Mysql();

setTitle("Lista Anaforwn"); //titlos para8yrou

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 500x400 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-250, dim.height/2-200, 500, 400);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

// lista me anafores

listModel = **new** DefaultListModel<String>();

anaforesList = **new** JList<String>(listModel);

getContentPane().add(anaforesList);

//vazei thn idiotita scroll sthn lista

JScrollPane scrollPane = **new** JScrollPane(anaforesList);

scrollPane.setBounds(0, 0, 500, 400);

getContentPane().add(scrollPane);

addAnafores(); //pairnei tis anafores apo thn vash kai ths vazei sthn lista

}

**private** **void** addAnafores(){

anafores=con.getAnafores(); //pairnei tis anafores apo thn vash

anaforesInfo = **new** HashMap<String, String>(); //dimiourgia HashMap

**for**(**int** i=0;i<anafores.size();i++){

anaforesInfo=anafores.get(i); //pairnei ta stoixia mias seiras

listModel.addElement((i+1)+

") "+anaforesInfo.get("time")+" Oxima: "+anaforesInfo.get("oxima")+" "+anaforesInfo.get("provlima")+" ("+anaforesInfo.get("sxolia")+")");

//ta vazei sthn lista pros emfanish

}

}

}

### Class AnefodiasmoiAdd

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseAdapter;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;

**import** javax.swing.JComboBox;

**public** **class** AnefodiasmoiAdd **extends** JFrame {

**private** JTextField litra; //pedio litrwn

**private** Mysql con; //antikeimeno ths Mysql

**private** JComboBox<String> oximataBox; //dropdown list me ta oximata

**public** AnefodiasmoiAdd() {

/\* kartela prosthikis anefodiasmwn\*/

con = **new** Mysql();

setTitle("Prosthiki Anefodiasmou"); //titlos para8yrou

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 300x220 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-150, dim.height/2-110, 300, 220);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

JLabel litraLabel = **new** JLabel("Litra:");

litraLabel.setBounds(10, 11, 70, 14);

getContentPane().add(litraLabel);

litra = **new** JTextField();

litra.setBounds(101, 8, 125, 20);

getContentPane().add(litra);

litra.setColumns(10);

JLabel oximaL = **new** JLabel("Oxima:");

oximaL.setBounds(10, 48, 46, 14);

getContentPane().add(oximaL);

oximataBox = **new** JComboBox<String>();

oximataBox.setBounds(100, 48, 126, 20);

getContentPane().add(oximataBox);

addOximata(); //pairnei ta oximata apo thn vash kai ta vazei sthn dropdown lista oximataBox

JButton saveButton = **new** JButton("Save");

saveButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\* otan path8ei to koumpi pairnei tin timi pou exei mpei sta litra kanei elegxo an einai swsth

\* kai epeita kalei thn function save(String litraV,String oximaV) kai apothikeuei ta stoixia sthn vash

\*/

String litraValue=litra.getText(); //pairnei thn timi tou pedioi

String oximaValue=oximataBox.getSelectedItem().toString(); //pairnei tin timi tis dropdown list

**if**(checkData(litraValue)) //sunarthsh pou elegxei an einai egkurh h timi

save(litraValue,oximaValue); //apothikeuei ta stoixia sthn vash

}

});

saveButton.setBounds(101, 141, 125, 40);

getContentPane().add(saveButton);

}

**private** **void** addOximata(){

/\*

\* vazei ta oximata apo thn vash sthn dropdown lista

\*/

ArrayList<String> oximata=con.getOximata(); //pairnei ta oximata apo tin vasi kai ta apothikeui sthn ArrayList<String>

**for**(**int** i=0;i<oximata.size();i++){

oximataBox.addItem(oximata.get(i)); //prostheti to oxima sthn dropdown lista

}

}

**private** Boolean checkData(String litra){

/\*

\* elegxei an einai swsti i timi kai an oxi tote emfanizei minima lathous

\*/

**if**(litra.length()<1){

JOptionPane.*showMessageDialog*(**null**, "Den Dwsate arithmo litrwn");

**return** **false**;

}

**try**{

**double** d = Double.*parseDouble*(litra);

}**catch**(NumberFormatException nfe){

JOptionPane.*showMessageDialog*(**null**, "Dwste Egkiro arithmo!");

**return** **false**;

}

**return** **true**;

}

**private** **void** save(String litraV,String oximaV){

/\*

\* apothikeuei ta stoixia sthn vash elegxei an apothikeutikan kai emfanizei minima epituxia i apotuxias antistoixa

\*/

String query="insert into anefodiasmoi(litra,oxima\_id) values(?,?)"; //dhmiourgia query

**try** {

con.pS=con.conn.prepareStatement(query);

con.pS.setString(1, litraV);

con.pS.setInt(2, Integer.*parseInt*(oximaV));

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(con.makeQuery(query)){ //ektelei to query me thn vohthia tiw function makeQuery(String query)

JOptionPane.*showMessageDialog*(**null**, "Saved!"); //an den upardxei lathos emfanizei to minima epituxias

litra.setText("");

}**else**{

JOptionPane.*showMessageDialog*(**null**, "Something went wrong Please Try later!"); //an uparxei lathos tote emfanizei minima lathous

}

}

}

### Class DiaxiristisCard

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** javax.swing.DefaultListModel;

**import** javax.swing.JFrame;

**import** javax.swing.JList;

**import** javax.swing.JScrollPane;

**import** javax.swing.JLabel;

**public** **class** DiaxiristisCard **extends** JFrame {

**private** DefaultListModel<String> listModel;

**private** JList<String> oximataList;

**private** ArrayList<HashMap<String, String>> oximata;

**private** HashMap<String, String> oximataInfo;

**private** Mysql con;

**private** JLabel odigoiLabel,oximataLabel;

**public** DiaxiristisCard() {

/\*

\* kartela diaxiristi

\*/

con = **new** Mysql();

setTitle("Lista Anaforwn");

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 700x500 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-350, dim.height/2-250, 700, 500);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

odigoiLabel = **new** JLabel("");

odigoiLabel.setBounds(15, 11, 100, 14);

getContentPane().add(odigoiLabel);

oximataLabel = **new** JLabel("");

oximataLabel.setBounds(120, 11, 100, 14);

getContentPane().add(oximataLabel);

listModel = **new** DefaultListModel<String>();

oximataList = **new** JList<String>(listModel);

getContentPane().add(oximataList);

JScrollPane scrollPane = **new** JScrollPane(oximataList);

scrollPane.setBounds(0, 30, 700, 500);

getContentPane().add(scrollPane);

addStatistika();// function i opoia pairnei ta statistika apo tin vasi kai ta emfanizei

}

**private** **void** addStatistika(){

/\*

\* pairnei ola ta aparaithta stoixia apo thn vash kai ta emfanizei ston diaxiristi

\*/

String oximataNum=con.getNumOfOximata(); //pairnei ton arithmo twn oximatwn ths vashs

String odigoiNum=con.getNumOfOdigoi(); //pairnei ton arithmo twn odigwn thw vashs

oximataLabel.setText("Oximata: "+oximataNum); //emfanizei ton arithmo ton oximatwn

odigoiLabel.setText("Odigoi: "+odigoiNum); //emfanizei ton arithmo ton odigwn

oximata=con.getOximataStatistika(); //pairnei ta aparaitita stoixia apo thn vash

oximataInfo = **new** HashMap<String, String>();

**for**(**int** i=0;i<oximata.size();i++){

oximataInfo=oximata.get(i);

listModel.addElement((i+1)+

") Oxima: "+oximataInfo.get("id")+" Odigos: "+oximataInfo.get("odigosOnoma")+" ("+oximataInfo.get("odigosId")+") Anafores: "+oximataInfo.get("anaforesNum")+" Anefodiasmoi: "+oximataInfo.get("anefodiasmoiNum"));

//vazei ta dedomena sthn lista

}

}

}

### Class IpalilosCard

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JButton;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**public** **class** IpalilosCard **extends** JFrame {

**public** IpalilosCard() {

/\*

\* kartela ipalilou apo thn opoioa epilegei ti energeia thelei na kanei

\*/

**this**.setTitle("Kartela Ipallilou"); //titlos parathirou

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 250x238 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-125, dim.height/2-119, 250, 238);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setLayout(**null**);

JButton odigoi = **new** JButton("Odigoi");

odigoi.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei thn kartela me thn lista twn odhgwn

\*/

OdigoiList frame = **new** OdigoiList();

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

odigoi.setBounds(0, 0, 244, 35);

getContentPane().add(odigoi);

JButton oximata = **new** JButton("Oximata");

oximata.setBounds(0, 35, 244, 35);

getContentPane().add(oximata);

oximata.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei thn kartela me thn lista twn oximatwn

\*/

OximataList frame = **new** OximataList();

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

JButton odigoiAdd = **new** JButton("Prosthiki Odigou");

odigoiAdd.setBounds(0, 70, 244, 35);

getContentPane().add(odigoiAdd);

odigoiAdd.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei to parathiro prosthikis twn odhgwn

\*/

OdigoiAdd frame = **new** OdigoiAdd(0);

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

JButton oximataAdd = **new** JButton("Prosthiki Oximatos");

oximataAdd.setBounds(0, 105, 244, 35);

getContentPane().add(oximataAdd);

oximataAdd.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei to parathiro prosthikis twn oximatwn

\*/

OximataAdd frame = **new** OximataAdd(0);

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

JButton anefodiasmoiAdd = **new** JButton("Prosthiki Anefodiasmou");

anefodiasmoiAdd.setBounds(0, 140, 244, 35);

getContentPane().add(anefodiasmoiAdd);

anefodiasmoiAdd.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei to parathiro me thn lista twn anefodiasmwn

\*/

AnefodiasmoiAdd frame = **new** AnefodiasmoiAdd();

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

JButton anafores = **new** JButton("Anafores");

anafores.setBounds(0, 175, 244, 35);

getContentPane().add(anafores);

anafores.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei to parathiro me thn lista twn anaforwn

\*/

Anafores frame = **new** Anafores();

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

}

}

### Class Login

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.Dimension;

**import** java.awt.EventQueue;

**import** java.awt.Toolkit;

**import** javax.swing.JFrame;

**import** javax.swing.JButton;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPasswordField;

**import** java.awt.Font;

**import** javax.swing.JLabel;

**import** java.awt.Color;

**import** javax.swing.JTextField;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**public** **class** Login **extends** JFrame {

**private** JTextField usernameField;

**private** JPasswordField passwordField;

**private** Mysql con;

/\*\*

\* Dimiourgei to frame gia to login

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Login frame = **new** Login();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

**public** Login() {

/\*

\* eisagontai username kai password ginetai elegxos an einai swsta kai anoigei thn katallhlh kartela analoga me ton xrhsth pou ekane login

\*/

con = **new** Mysql();

setTitle("Login");

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 300x250 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-150, dim.height/2-125, 300, 250);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

JButton btnLogin = **new** JButton("Login");

btnLogin.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\*

\* pairnei ta stoixia kanei elegxo kai emfanizei munhma lathous h anoigei thn katalhlh kartela analoga thn orthothta autwn

\*/

String username,password;

String[] isDataCorrect=**new** String[2];

String odigosId=**null**;

username=usernameField.getText(); //pairnei thn timi

password=passwordField.~~getText~~();

isDataCorrect=checkData(username,password); //elegxei an einai apodexta ta stoixia

**if**(isDataCorrect[0].equals("0")){ //an einai lathos ta stoixia tote emfanizei minima lathous

JOptionPane.*showMessageDialog*(**null**, "Ta stoixeia pou dwsate einai lathos!");

}**else**{

**if**(isDataCorrect[0].equals("odigos")) //an o xristis einai odigos pernaei to id tou sthn metavlhth

odigosId=isDataCorrect[1];

openCard(isDataCorrect[0],odigosId); //anoigei thn katallhlh kartela analoga me ton xristi pou ekane login

}

}

});

btnLogin.setFont(**new** Font("Aharoni", Font.*PLAIN*, 17));

btnLogin.setBounds(100, 171, 100, 40);

getContentPane().add(btnLogin);

JLabel lblUsername = **new** JLabel("Username:");

lblUsername.setBounds(38, 64, 70, 14);

getContentPane().add(lblUsername);

usernameField = **new** JTextField();

usernameField.setBounds(118, 61, 100, 20);

getContentPane().add(usernameField);

usernameField.setColumns(10);

JLabel lblPassword = **new** JLabel("Password:");

lblPassword.setBounds(38, 108, 70, 14);

getContentPane().add(lblPassword);

passwordField = **new** JPasswordField();

passwordField.setBounds(118, 105, 100, 20);

getContentPane().add(passwordField);

passwordField.setColumns(10);

}

**private** String[] checkData(String username,String password){ // NOPMD by Kalom on 11/1/2014 10:34 μμ

/\*

\* elegxei an einai swsta ta stoixia pou dothikan

\*/

**return** con.checkLogin(username,password);

}

**private** **void** openCard(String priv,String odigosId){

/\*

\* anoigei thn katalili kartela analoga me ton xristi pou ekane login

\*/

**if**(priv.equals("odigos")){

**try** {

OdigosCard frame = **new** OdigosCard(Integer.*parseInt*(odigosId));

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}**else** **if**(priv.equals("diaxiristis")){

**try** {

DiaxiristisCard frame = **new** DiaxiristisCard();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}**else**{

**try** {

IpalilosCard frame = **new** IpalilosCard();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

dispose(); //kleinei to parathiro tou login meta to anoigma tis neas kartelas

}

}

### Class Mysql

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** javax.swing.JOptionPane;

**public** **class** Mysql {

**public** Connection conn=**null**;

**public** PreparedStatement pS=**null**;

**public** Statement stmt;

**private** ResultSet rs;

**private** String user="root";

**private** String pass="123456";

**public** Mysql(){

/\*

\* kanei thn sundesh me thn vash sthn dimiourgia tou antikeimenou

\*/

**try**{

Class.*forName*("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost:3306/askisi9?useUnicode=true&characterEncoding=UTF-8&";

conn = DriverManager

.*getConnection*(url

+ "user="+user+"&password="+pass);

stmt=conn.createStatement();

stmt.execute("set names utf8");

}**catch**(Exception e){

JOptionPane.*showMessageDialog*(**null**, "Apotixia Epikoinwnias me tin vash!");

e.printStackTrace();

}

}

**public** ArrayList<String> getOximata(){

/\*

\* epistefei ta oxhmata ths vashs

\*/

ArrayList<String> oximata=**new** ArrayList<String>();

**try** {

pS = conn.prepareStatement("SELECT \* FROM oximata");

rs=pS.executeQuery();

**while**(rs.next())

oximata.add(rs.getString("id"));

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** oximata;

}

**public** ArrayList<HashMap<String, String>> getOxima(**int** id){

/\*

\* epistefei ta stoixia enos sugkekrimenou oximatos

\*/

ArrayList<HashMap<String, String>> oxima=**new** ArrayList<HashMap<String, String>>();

HashMap<String, String> map = **new** HashMap<String, String>();

**try** {

pS = conn.prepareStatement("SELECT \* FROM oximata WHERE id=?");

pS.setInt(1, id);

rs=pS.executeQuery();

**if**(rs.next()){

map.put("id", id+"");

map.put("diadromi", rs.getString("diadromi"));

oxima.add(map);

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** oxima;

}

**public** String getNumOfOximata(){

/\*

\* epistefei ton arithmo twn oximatwn ths vashs

\*/

String num=**null**;

**try** {

pS = conn.prepareStatement("SELECT COUNT(\*) AS num FROM oximata");

rs=pS.executeQuery();

**while**(rs.next()){

num=rs.getString("num");

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** num;

}

**public** String getNumOfOdigoi(){

/\*

\* epistefei ton arithmo twn odigwn ths vashs

\*/

String num=**null**;

**try** {

pS = conn.prepareStatement("SELECT COUNT(\*) AS num FROM odigoi");

rs=pS.executeQuery();

**while**(rs.next()){

num=rs.getString("num");

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** num;

}

**public** ArrayList<HashMap<String, String>> getAnafores(){

/\*

\* epistrefei tis anafores ths vashs se mia lista opou pairnei to onoma tou column kai thn timi tou

\*/

ArrayList<HashMap<String, String>> anafores=**new** ArrayList<HashMap<String, String>>();

**try** {

pS = conn.prepareStatement("SELECT \* FROM anafores INNER JOIN odigoi ON odigos\_id=odigoi.id");

rs=pS.executeQuery();

**while**(rs.next()){

HashMap<String, String> map = **new** HashMap<String, String>();

map.put("id", rs.getString("id"));

map.put("provlima", rs.getString("provlima"));

map.put("sxolia", rs.getString("sxolia"));

map.put("oxima", rs.getString("oxima"));

map.put("time", rs.getString("time"));

anafores.add(map);

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** anafores;

}

**public** ArrayList<HashMap<String, String>> getOximataStatistika(){

/\*

\* epistrefei ta statistika gia ton diaxiristi

\*/

ArrayList<HashMap<String, String>> anafores=**new** ArrayList<HashMap<String, String>>();

PreparedStatement newPs;

ResultSet newRs;

**try** {

pS = conn.prepareStatement("SELECT \* FROM oximata INNER JOIN odigoi ON oximata.id=oxima");

rs=pS.executeQuery();

**while**(rs.next()){

HashMap<String, String> map = **new** HashMap<String, String>();

map.put("id", rs.getString("oximata.id"));

map.put("odigosId", rs.getString("odigoi.id"));

map.put("odigosOnoma", rs.getString("onoma"));

newPs= conn.prepareStatement("SELECT COUNT(\*) AS num FROM anafores INNER JOIN odigoi ON odigos\_id=odigoi.id WHERE" +

" odigoi.id =?");

newPs.setString(1, rs.getString("odigoi.id"));

newRs=newPs.executeQuery();

**if**(newRs.next())

map.put("anaforesNum", newRs.getString("num"));

newPs= conn.prepareStatement("SELECT COUNT(\*) AS num FROM anefodiasmoi INNER JOIN oximata ON oxima\_id=oximata.id WHERE" +

" oximata.id =?");

newPs.setString(1, rs.getString("oximata.id"));

newRs=newPs.executeQuery();

**if**(newRs.next())

map.put("anefodiasmoiNum", newRs.getString("num"));

anafores.add(map);

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** anafores;

}

**public** ArrayList<HashMap<String, String>> getOdigous(){

/\*

\* epistrefei tous odigous ths vashs

\*/

ArrayList<HashMap<String, String>> odigoi=**new** ArrayList<HashMap<String, String>>();

**try** {

pS = conn.prepareStatement("SELECT \* FROM odigoi");

rs=pS.executeQuery();

**while**(rs.next()){

HashMap<String, String> map = **new** HashMap<String, String>();

map.put("id", rs.getString("id"));

map.put("onoma", rs.getString("onoma"));

map.put("eponimo", rs.getString("eponimo"));

map.put("username", rs.getString("username"));

map.put("password", rs.getString("password"));

map.put("oxima", rs.getString("oxima"));

odigoi.add(map);

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** odigoi;

}

**public** ArrayList<HashMap<String, String>> getOdigo(**int** id){

/\*

\* epistrefei ta stoixia enos sugkekrimenou odigou

\*/

ArrayList<HashMap<String, String>> odigoi=**new** ArrayList<HashMap<String, String>>();

HashMap<String, String> map = **new** HashMap<String, String>();

**try** {

pS = conn.prepareStatement("SELECT \* FROM odigoi WHERE id=?");

pS.setInt(1, id);

rs=pS.executeQuery();

**if**(rs.next()){

map.put("id", rs.getString("id"));

map.put("onoma", rs.getString("onoma"));

map.put("eponimo", rs.getString("eponimo"));

map.put("username", rs.getString("username"));

map.put("password", rs.getString("password"));

map.put("oxima", rs.getString("oxima"));

odigoi.add(map);

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** odigoi;

}

**public** String[] getLatLon(**int** id){

/\*

\* epistrefei tis suntetagmenes enos oximatos

\*/

String[] oxima=**new** String[2];

**try** {

pS = conn.prepareStatement("SELECT \* FROM oximata WHERE id=?");

pS.setInt(1, id);

rs=pS.executeQuery();

**if**(rs.next()){

oxima[0]=rs.getString("lat");

oxima[1]=rs.getString("lon");

}

} **catch** (SQLException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** oxima;

}

**public** Boolean makeQuery(String query){

/\*

\* ektelei to query pou tou dinetai kai epistrefei an ektelesthke kai an oxi

\*/

**try** {

**if**(pS.executeUpdate()==0)

**return** **false**;

**else**

**return** **true**;

} **catch** (SQLException e) {

e.printStackTrace();

**return** **false**;

} **catch** (Exception e) {

e.printStackTrace();

**return** **false**;

}

}

**public** String[] checkLogin(String us,String pass){

/\*

\* elegxei an einai swsta ta stoixia tou login kai epistrefei to epipedo tou user px(ipalilos)

\*/

String[] priv= **new** String[2];

priv[0]="0";

**try** {

pS = conn.prepareStatement("SELECT \* FROM users WHERE username=? AND password=?");

pS.setString(1, us);

pS.setString(2, pass);

rs=pS.executeQuery();

**while**(rs.next())

priv[0]=rs.getString("privilages");

**if**(priv[0].equals("0")){

pS = conn.prepareStatement("SELECT \* FROM odigoi WHERE username=? AND password=?");

pS.setString(1, us);

pS.setString(2, pass);

rs=pS.executeQuery();

**while**(rs.next()){

priv[0]="odigos";

priv[1]=rs.getString("id");

}

}

} **catch** (SQLException e) {

e.printStackTrace();

priv[0]="0";

} **catch** (Exception e) {

e.printStackTrace();

priv[0]="0";

}

**return** priv;

}

}

### Class OdigoiAdd

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.EventQueue;

**import** java.awt.Toolkit;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** javax.swing.JList;

**import** javax.swing.JSpinner;

**import** javax.swing.JPasswordField;

**import** javax.swing.JButton;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** javax.swing.JComboBox;

**public** **class** OdigoiAdd **extends** JFrame {

/\*\*

\*

\*/

**private** **static** **final** **long** *serialVersionUID* = 1L;

**private** JTextField name,surname,username;

**private** JPasswordField pass;

**private** JPasswordField pass2;

**private** JComboBox<String> oximataBox;

**private** Mysql con;

**private** **int** id;

**private** JLabel oximaLabel;

**public** OdigoiAdd(**int** editId) {

/\*

\* parathiro gia thn prosthiki i tin epe3ergasia odigou, an to editId einai 0 tote kanei prosthiki alliws epexergasia

\*/

con = **new** Mysql();

id=editId;

**if**(id==0) // vazei ton katallhlo titlo analoga thn leitourgia pou exei epilex8ei

setTitle("Prosthiki Odigou");

**else**

setTitle("Epe3ergasia Odigou");

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 300x220 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-150, dim.height/2-125, 300, 245);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

JLabel onomaLabel = **new** JLabel("Onoma:");

onomaLabel.setBounds(10, 11, 70, 14);

getContentPane().add(onomaLabel);

JLabel eponimoLabel = **new** JLabel("Epwnimo:");

eponimoLabel.setBounds(10, 36, 70, 14);

getContentPane().add(eponimoLabel);

JLabel usernameLabel = **new** JLabel("Username:");

usernameLabel.setBounds(10, 61, 70, 14);

getContentPane().add(usernameLabel);

JLabel kodikosLabel = **new** JLabel("Password:");

kodikosLabel.setBounds(10, 86, 70, 14);

getContentPane().add(kodikosLabel);

JLabel kodikosELabel = **new** JLabel("Epal. Kwdikou:");

kodikosELabel.setBounds(10, 111, 70, 14);

getContentPane().add(kodikosELabel);

oximaLabel = **new** JLabel("Oxima:");

oximaLabel.setBounds(10, 136, 90, 14);

getContentPane().add(oximaLabel);

name = **new** JTextField();

name.setBounds(101, 8, 125, 20);

getContentPane().add(name);

name.setColumns(10);

surname = **new** JTextField();

surname.setBounds(101, 33, 125, 20);

getContentPane().add(surname);

surname.setColumns(10);

username = **new** JTextField();

username.setBounds(101, 58, 125, 20);

getContentPane().add(username);

username.setColumns(10);

pass = **new** JPasswordField();

pass.setBounds(101, 83, 125, 20);

getContentPane().add(pass);

pass2 = **new** JPasswordField();

pass2.setBounds(101, 108, 125, 17);

getContentPane().add(pass2);

oximataBox = **new** JComboBox<String>();

oximataBox.setBounds(101, 133, 125, 20);

getContentPane().add(oximataBox);

addOximata();

JButton saveButton = **new** JButton("Save");

saveButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\*

\* apothikeuei tis times sthn vash afou tis elen3ei prwta

\*/

String nameValue=name.getText();

String surnameValue=surname.getText();

String usernameValue=username.getText();

String passValue=pass.~~getText~~();

String pass2Value=pass2.~~getText~~();

String oximaValue=oximataBox.getSelectedItem().toString();

String action;

**if**(checkData(nameValue,surnameValue,passValue,pass2Value)){

**if**(id==0)

action="insert";

**else**

action="update";

save(nameValue,surnameValue,usernameValue,passValue,oximaValue,action);

}

}

});

saveButton.setBounds(101, 167, 125, 40);

getContentPane().add(saveButton);

**if**(id!=0){ // an exei epilexthei i leitourgia tiw epe3ergasias tote pairnei ta stoixia tou odigou pros epe3ergasia

getOdigoForEdit(id);

}

}

**private** **void** addOximata(){

/\*

\* vazei ta oximata tis vashs sthn lista

\*/

ArrayList<String> oximata=con.getOximata();

**for**(**int** i=0;i<oximata.size();i++){

oximataBox.addItem(oximata.get(i));

}

}

**private** **void** getOdigoForEdit(**int** id){

/\*

\* pairnei ta stoixia tou odigou pou einai pros epe3ergasia

\*/

ArrayList<HashMap<String, String>> odigos=con.getOdigo(id);

HashMap<String, String> odigosInfo = **new** HashMap<String, String>();

odigosInfo=odigos.get(0);

name.setText(odigosInfo.get("onoma"));

surname.setText(odigosInfo.get("eponimo"));

}

**private** Boolean checkData(String name,String surname,String pass,String pass2){

/\*

\* elegxei an einai swsta ta dedomena pou dothikan kai epistrefei to katalilo mnm an den einai

\*/

**if**(name.length()<3){

JOptionPane.*showMessageDialog*(**null**, "Dwste egkiro onoma!");

**return** **false**;

}**else** **if**(surname.length()<3){

JOptionPane.*showMessageDialog*(**null**, "Dwste egkiro epwnimo!");

**return** **false**;

}**else** **if**(pass.length()<5){

JOptionPane.*showMessageDialog*(**null**, "O kwdikos prepei na einai megaliteros apo 4 xaraktires!");

**return** **false**;

}**else** **if**(!pass.equals(pass2)){

JOptionPane.*showMessageDialog*(**null**, "Oi kwdikoi den teriazoun!");

**return** **false**;

}**else**{

**return** **true**;

}

}

**private** **void** save(String nameV,String surnameV,String usernameV,String passV,String oximaV,String action){

/\*

\* apothikeuei ta stoixia i tis allages sthn vash kai emfanizei ta katallhla minimata

\*/

String query;

**if**(action.equals("insert"))

query="insert into odigoi(onoma,eponimo,username,password,oxima) values(?,?,?,?,?)";

**else**

query="update odigoi set onoma=?, eponimo=? , username=?,password=?,oxima=? WHERE id=?";

**try** {

con.pS=con.conn.prepareStatement(query);

con.pS.setString(1, nameV);

con.pS.setString(2, surnameV);

con.pS.setString(3, usernameV);

con.pS.setString(4, passV);

con.pS.setString(5, oximaV);

**if**(action.equals("update"))

con.pS.setInt(6, id);

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(con.makeQuery(query)){

JOptionPane.*showMessageDialog*(**null**, "Saved!");

**if**(id!=0)

dispose();

**else**{

name.setText("");

surname.setText("");

username.setText("");

pass.setText("");

pass2.setText("");

}

}**else**{

JOptionPane.*showMessageDialog*(**null**, "Something went wrong please try later!");

}

}

}

### Class OdigoiList

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** java.net.URL;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** javax.swing.DefaultListModel;

**import** javax.swing.ImageIcon;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JScrollPane;

**import** javax.swing.JList;

**import** javax.swing.JButton;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**public** **class** OdigoiList **extends** JFrame {

**private** Mysql con;

**private** DefaultListModel<String> listModel;

**private** JList<String> odigoiList;

**private** ArrayList<String> odigoiId=**new** ArrayList<String>();

**private** ArrayList<HashMap<String, String>> odigoi;

**private** HashMap<String, String> odigoiInfo;

**public** OdigoiList() {

con = **new** Mysql();

setTitle("Lista Odigwn");

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 355x40 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-150, dim.height/2-125, 355, 400);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

listModel = **new** DefaultListModel<String>();

odigoiList = **new** JList<String>(listModel);

getContentPane().add(odigoiList);

JScrollPane scrollPane = **new** JScrollPane(odigoiList);

scrollPane.setBounds(0, 0, 355, 340);

getContentPane().add(scrollPane);

JButton editButton = **new** JButton("Epe3ergasia");

editButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei to parathiro epexergasias gia ton epilegmeno odigo

\*/

odigoiInfo=odigoi.get(odigoiList.getSelectedIndex());

OdigoiAdd frame = **new** OdigoiAdd(Integer.*parseInt*(odigoiInfo.get("id")));

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

editButton.setBounds(0, 338, 175, 34);

getContentPane().add(editButton);

JButton deleteButton = **new** JButton("Diagrafh");

deleteButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\*

\* diagrafi ton epilegmeno odigo epeita apo erwthsh

\*/

**int** dialogResult = JOptionPane.*showConfirmDialog* (**null**, "Eiste sigouri oti thelete na diagrapsete ton sigkekrimeno odigo?","Diagrafh",JOptionPane.*YES\_NO\_OPTION*);

**if**(dialogResult == JOptionPane.*YES\_OPTION*){

delete(Integer.*parseInt*(odigoiId.get(odigoiList.getSelectedIndex())),odigoiList.getSelectedIndex());

}

}

});

deleteButton.setBounds(175, 338, 175, 34);

getContentPane().add(deleteButton);

addOdigous();

}

**private** **void** addOdigous(){

/\*

\* vazei tous odigous ths vashs sthn lista

\*/

odigoi=con.getOdigous();

odigoiInfo = **new** HashMap<String, String>();

**for**(**int** i=0;i<odigoi.size();i++){

odigoiInfo=odigoi.get(i);

odigoiId.add(odigoiInfo.get("id"));

listModel.addElement(odigoiInfo.get("onoma")+

" "+odigoiInfo.get("eponimo")+" ("+odigoiInfo.get("oxima")+")");

}

}

**private** **void** delete(**int** odigosId,**int** index){

/\*

\* diagrafei ton epilegmeno odigo apo thn vash kai epistrefei to katallilo minima

\*/

String query="delete from odigoi where id=?";

**try** {

con.pS=con.conn.prepareStatement(query);

con.pS.setInt(1, odigosId);

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(con.makeQuery(query)){

JOptionPane.*showMessageDialog*(**null**, "Diagraftike!");

listModel.remove(index);

odigoiId.remove(index);

}**else**{

JOptionPane.*showMessageDialog*(**null**, "Something went wrong please try later!");

}

}

}

### Class OdigosCard

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.BorderLayout;

**import** java.awt.Dimension;

**import** java.awt.EventQueue;

**import** java.awt.Toolkit;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextPane;

**import** javax.swing.JComboBox;

**import** javax.swing.JButton;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.sql.SQLException;

**public** **class** OdigosCard **extends** JFrame {

**private** JTextPane sxolia;

**private** JComboBox<String> provlima;

**private** String[] provlimata={"Lastixo","Xreiazetai Service","Den pairnei Mprosta","Trakarisma","Allo"};

**private** Mysql con;

**private** **int** odigosId;

/\*\*

\* Create the frame.

\*/

**public** OdigosCard(**int** id) {

/\*

\* kartela odigou gia anafora provlimatos

\*/

con = **new** Mysql();

odigosId=id;

**this**.setTitle("Anafora Provlimatos");

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 350x250 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-150, dim.height/2-125, 350, 250);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setLayout(**null**);

JLabel provlimaLabel = **new** JLabel("Provlima:");

provlimaLabel.setBounds(19, 11, 67, 14);

getContentPane().add(provlimaLabel);

JLabel sxoliaLabel = **new** JLabel("Sxolia:");

sxoliaLabel.setBounds(19, 36, 67, 14);

getContentPane().add(sxoliaLabel);

sxolia = **new** JTextPane();

sxolia.setBounds(107, 36, 189, 131);

getContentPane().add(sxolia);

provlima = **new** JComboBox<String>();

provlima.setBounds(107, 8, 189, 20);

getContentPane().add(provlima);

**for**(**int** i=0;i<provlimata.length;i++)

provlima.addItem(provlimata[i]);

JButton sendButton = **new** JButton("Send");

sendButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\*

\* elegxei ta stoixia kai epeita stelnei tin anafora apothikeuontas thn sthn vash

\*/

**if**(checkData(sxolia.getText()))

send(provlima.getSelectedItem().toString(),sxolia.getText());

}

});

sendButton.setBounds(107, 188, 100, 23);

getContentPane().add(sendButton);

}

**private** **void** send(String pro,String sxo){

/\*

\* apothikeuei sthn vash thn anafora

\*/

String query="insert into anafores(provlima,sxolia,odigos\_id) values(?,?,?)";

**try** {

con.pS=con.conn.prepareStatement(query);

con.pS.setString(1, pro);

con.pS.setString(2, sxo);

con.pS.setInt(3, odigosId);

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(con.makeQuery(query)){

JOptionPane.*showMessageDialog*(**null**, "Stalthike!");

sxolia.setText("");

}**else**{

JOptionPane.*showMessageDialog*(**null**, "Something went wrong please try later!");

}

}

**private** Boolean checkData(String sx){

/\*

\* elegxei an exei dothei sxolio

\*/

**if**(sx.length()<1){

JOptionPane.*showMessageDialog*(**null**, "Grapse kapoio sxolio!");

**return** **false**;

}**else**{

**return** **true**;

}

}

}

### Class OximataAdd

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.EventQueue;

**import** java.awt.Toolkit;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;

**import** javax.swing.border.EmptyBorder;

**public** **class** OximataAdd **extends** JFrame {

**private** JTextField kodikos,diadromi;

**private** Mysql con;

**private** **int** id;

/\*\*

\* Create the frame.

\*/

**public** OximataAdd(**int** editId) {

/\*

\* parathiro gia prosthiki oximatos an to editId einai 0 i epe3ergasia an einai megaluterao tou 0

\*/

con = **new** Mysql();

id=editId;

**if**(id==0) //emfanizei ton katalilo titlo analoga thn leitourgia pou exei epilex8ei

setTitle("Prosthiki Oxhmatos");

**else**

setTitle("Epe3ergasia Oxhmatos");

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 300x220 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-150, dim.height/2-110, 300, 220);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

JLabel kodikosLabel = **new** JLabel("Kwdikos:");

kodikosLabel.setBounds(10, 11, 70, 14);

getContentPane().add(kodikosLabel);

JLabel diadromiLabel = **new** JLabel("Diadromi:");

diadromiLabel.setBounds(10, 36, 70, 14);

getContentPane().add(diadromiLabel);

kodikos = **new** JTextField();

kodikos.setBounds(101, 8, 125, 20);

getContentPane().add(kodikos);

kodikos.setColumns(10);

diadromi = **new** JTextField();

diadromi.setBounds(101, 33, 125, 20);

getContentPane().add(diadromi);

diadromi.setColumns(10);

JButton saveButton = **new** JButton("Save");

saveButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\*

\* apothikeuei sthn vash ta stoixia tou oximatos afou kanei elegxo se auta prwta

\*/

String kodikosValue=kodikos.getText();

String diadromiValue=diadromi.getText();

**if**(checkData(kodikosValue,diadromiValue))

save(kodikosValue,diadromiValue);

}

});

saveButton.setBounds(101, 141, 125, 40);

getContentPane().add(saveButton);

**if**(id!=0){

getOximaForEdit(id);

}

}

**private** **void** getOximaForEdit(**int** id){

/\*

\* pairnei ta stoixia tou oximatos pou einai pros epe3ergasia apo thn vash

\*/

ArrayList<HashMap<String, String>> oxima=con.getOxima(id);

HashMap<String, String> oximaInfo = **new** HashMap<String, String>();

oximaInfo=oxima.get(0);

kodikos.setText(oximaInfo.get("id"));

diadromi.setText(oximaInfo.get("diadromi"));

}

**private** Boolean checkData(String kod,String diad){

/\*

\* elegxei an einai swsta ta dedomena pou exoun do8ei

\*/

**if**(kod.length()<2){

JOptionPane.*showMessageDialog*(**null**, "Dwste egkuro kwdiko!");

**return** **false**;

}**else** **if**(diad.length()<3){

JOptionPane.*showMessageDialog*(**null**, "Dwste egkiri diadromh!");

**return** **false**;

}**else**{

**return** **true**;

}

}

**private** **void** save(String kodikosV,String daidromiV){

/\*

\* apothikeuei ta stoixia sthn vash kai emfanizei to katalilo minima ston xristi

\*/

String query="insert into oximata(id,diadromi,lat,lon) values(?,?,'37.973263','23.740768') on duplicate key update diadromi=?";

**try** {

con.pS=con.conn.prepareStatement(query);

con.pS.setInt(1, Integer.*parseInt*(kodikosV));

con.pS.setString(2, daidromiV);

con.pS.setString(3, daidromiV);

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(con.makeQuery(query)){

JOptionPane.*showMessageDialog*(**null**, "Saved!");

**if**(id!=0)

dispose();

**else**{

kodikos.setText("");

diadromi.setText("");

}

}**else**{

JOptionPane.*showMessageDialog*(**null**, "Something went wrong please try later!");

}

}

}

### Class OximataList

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** java.net.URL;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** javax.swing.DefaultListModel;

**import** javax.swing.ImageIcon;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JScrollPane;

**import** javax.swing.JList;

**import** javax.swing.JButton;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**public** **class** OximataList **extends** JFrame {

**private** Mysql con;

**private** DefaultListModel<String> listModel;

**private** ArrayList<String> oximataId=**new** ArrayList<String>();

**private** JList<String> oximataList;

**public** OximataList() {

/\*

\* parathiro me thn lista twn oximatwn

\*/

con = **new** Mysql();

setTitle("Lista Oximatwn");

/\* vgazei to para8uro sto kedro tis othonis me diastaseis 350x400 \*/

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

**this**.setBounds(dim.width/2-175, dim.height/2-200, 350, 400);

/\*--------------\*/

**this**.setResizable(**false**);

getContentPane().setBackground(Color.*WHITE*);

getContentPane().setLayout(**null**);

listModel = **new** DefaultListModel<String>();

oximataList = **new** JList<String>(listModel);

getContentPane().add(oximataList);

JScrollPane scrollPane = **new** JScrollPane(oximataList);

scrollPane.setBounds(0, 0, 350, 340);

getContentPane().add(scrollPane);

JButton editButton = **new** JButton("Epe3ergasia");

editButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

**try**{

/\*

\* anoigei to parathiro epe3ergasias gia to epilegmeno oxima

\*/

OximataAdd frame = **new** OximataAdd(Integer.*parseInt*(oximataList.getSelectedValue().toString()));

frame.setVisible(**true**);

}**catch**(Exception e){

e.printStackTrace();

}

}

});

editButton.setBounds(0, 338, 115, 34);

getContentPane().add(editButton);

JButton deleteButton = **new** JButton("Delete");

deleteButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\*

\* diagrafei to epilegmeno oxima epeita apo erwthsh

\*/

**int** dialogResult = JOptionPane.*showConfirmDialog* (**null**, "Thelete sigoura na diagrapsete to sigkekrimeno oxima?","Delete",JOptionPane.*YES\_NO\_OPTION*);

**if**(dialogResult == JOptionPane.*YES\_OPTION*){

delete(Integer.*parseInt*(listModel.get(oximataList.getSelectedIndex())),oximataList.getSelectedIndex());

}

}

});

deleteButton.setBounds(115, 338, 115, 34);

getContentPane().add(deleteButton);

JButton mapButton = **new** JButton("Map");

mapButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

/\*

\* anoigei ton xarti me thn topothesia tou oximatos

\*/

seeOnMap(Integer.*parseInt*(listModel.get(oximataList.getSelectedIndex())));

}

});

mapButton.setBounds(230, 338, 115, 34);

getContentPane().add(mapButton);

addOximata();

}

**private** **void** addOximata(){

/\*

\* vazei ta oximata tis vashs sthn lista

\*/

ArrayList<String> oximata=con.getOximata();

**for**(**int** i=0;i<oximata.size();i++){

listModel.addElement(oximata.get(i));

}

}

**private** **void** delete(**int** oximaId,**int** index){

/\*

\* diagrafei to epilegmeno oxima

\*/

String query="delete from oximata where id=?";

**try** {

con.pS=con.conn.prepareStatement(query);

con.pS.setInt(1, oximaId);

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(con.makeQuery(query)){

JOptionPane.*showMessageDialog*(**null**, "Deleted!");

listModel.remove(index);

oximataId.remove(index);

}**else**{

JOptionPane.*showMessageDialog*(**null**, "Something went wrong please try later!");

}

}

**private** **void** seeOnMap(**int** id){

/\*

\* emfanizei ton xarth me kedro thn topothesia tou oximatos

\*/

JFrame mapFrame = **new** JFrame("Topothesia Oxhmatos");

String[] latLon=con.getLatLon(id);

String lat,lon;

**if**(latLon!=**null**){

lat=latLon[0];

lon=latLon[1];

**try** {

String imageUrl = "http://maps.google.com/staticmap?center="+lat+","+lon+"&zoom=16&size=630x600&maptype=roadmap&markers=color:blue%7Clabel:S%7A"+lat+","+lon+"&key=ABQIAAAAgb5KEVTm54vkPcAkU9xOvBR30EG5jFWfUzfYJTWEkWk2p04CHxTGDNV791-cU95kOnweeZ0SsURYSA&format=jpg&sensor=false";

String destinationFile = "image.jpg";

URL url = **new** URL(imageUrl);

InputStream is = url.openStream();

OutputStream os = **new** FileOutputStream(destinationFile);

**byte**[] b = **new** **byte**[2048];

**int** length;

**while** ((length = is.read(b)) != -1) {

os.write(b, 0, length);

}

is.close();

os.close();

mapFrame.add(**new** JLabel(**new** ImageIcon((**new** ImageIcon("image.jpg")).getImage().getScaledInstance(630, 600,

java.awt.Image.*SCALE\_SMOOTH*))));

mapFrame.setVisible(**true**);

mapFrame.pack();

} **catch** (IOException e) {

e.printStackTrace();

System.*exit*(1);

}**catch** (Exception e) {

e.printStackTrace();

System.*exit*(1);

}

}

}

}

### Class AllTests

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** org.junit.runner.RunWith;

**import** org.junit.runners.Suite;

**import** org.junit.runners.Suite.SuiteClasses;

@RunWith(Suite.**class**)

@SuiteClasses({ AnaforesTest.**class**, IpalilosCardTest.**class**, LoginTest.**class**,

OdigoiAddTest.**class**, OximataAddTest.**class** })

**public** **class** AllTests {

}

### Class AnaforesTest

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** **static** org.junit.Assert.\*;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**public** **class** AnaforesTest {

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Before

**public** **void** setUp() **throws** Exception {

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** testAnafores() {

Anafores anafora = **new** Anafores();

}

}

### Class AnefodiasmoiAddTest

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** **static** org.junit.Assert.\*;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**public** **class** AnefodiasmoiAddTest {

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Before

**public** **void** setUp() **throws** Exception {

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** testAnefodiasmoiAdd() {

AnefodiasmoiAdd anefodiasmos = **new** AnefodiasmoiAdd();

}

}

### Class IpalilosCardTest

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** **static** org.junit.Assert.\*;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JButton;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**public** **class** IpalilosCardTest {

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Before

**public** **void** setUp() **throws** Exception {

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** testIpalilosCard() {

IpalilosCard ipalilos = **new** IpalilosCard();

}

}

### Class LoginTest

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** **static** org.junit.Assert.\*;

**import** java.awt.EventQueue;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**public** **class** LoginTest {

**private** Mysql con;

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Before

**public** **void** setUp() **throws** Exception {

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** testMain() {

con = **new** Mysql();

Login frame = **new** Login();

frame.setVisible(**true**);

}

@Test

**public** **void** testLogin(){

Login login = **new** Login();

}

}

### Class OdigoiAddTest

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** **static** org.junit.Assert.\*;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**public** **class** OdigoiAddTest {

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Before

**public** **void** setUp() **throws** Exception {

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** testOdigoiAdd() {

OdigoiAdd odigos = **new** OdigoiAdd(0);

}

}

### Class OdigosCardTest

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** **static** org.junit.Assert.\*;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**public** **class** OdigosCardTest {

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Before

**public** **void** setUp() **throws** Exception {

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** testOdigosCard() {

OdigosCard driver = **new** OdigosCard(51);

}

}

### Class OximataAddTest

**package** gr.teipir.mixanikilogismikou.askisi9;

**import** **static** org.junit.Assert.\*;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**public** **class** OximataAddTest {

@BeforeClass

**public** **static** **void** setUpBeforeClass() **throws** Exception {

}

@AfterClass

**public** **static** **void** tearDownAfterClass() **throws** Exception {

}

@Before

**public** **void** setUp() **throws** Exception {

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** testOximataAdd() {

OximataAdd prosthese = **new** OximataAdd(0);

}

}