**Σύστημα έκδοσης Θεατρικών Εισιτηρίων**

**1/14/2014**

Χρήστος Πετρόπουλος - 40202

Παναγιώτης Δημητρακόπουλος - 39275

Γιώργος Χαλικιόπουλος - 37082

Repository: ***https://github.com/PanosDimitrakopoulos/TheaterTickets***

Περιεχόμενα

[**1.**  **Εισαγωγή** 5](#_Toc378370272)

[1.1 Σκοπός του κειμένου 5](#_Toc378370273)

[1.2 Συμβάσεις του κειμένου 5](#_Toc378370274)

[1.3 Προορισμένο ακροατήριο και προτάσεις ανάγνωσης 5](#_Toc378370275)

[1.4 Στόχος και εύρος του προϊόντος 5](#_Toc378370276)

[2. **Γενική Περιγραφή** 6](#_Toc378370277)

[2.1 Προέλευση και προοπτική του προϊόντος 6](#_Toc378370278)

[2.2.1     Class Diagram 6](#_Toc378370279)

[2.2.2     User Classes and Characteristics 7](#_Toc378370280)

[2.2 Γενική περιγραφή των λειτουργιών του προϊόντος 7](#_Toc378370281)

[2.3 Κατηγορίες χρηστών 7](#_Toc378370282)

[2.4 Περιβάλλον λειτουργίας 8](#_Toc378370283)

[2.5 Περιορισμοί στη σχεδίαση και την υλοποίηση 8](#_Toc378370284)

[2.6 Τεκμηρίωση για το χρήστη 8](#_Toc378370285)

[**3.**  **Απαιτήσεις για τις εξωτερικές διεπαφές** 8](#_Toc378370286)

[3.1 Διεπαφή χρήστη 8](#_Toc378370287)

[3.2 Διεπαφές υλικού 8](#_Toc378370288)

[**4.**  **Λειτουργίες συστήματος** 8](#_Toc378370289)

[4.1 Επιλογή Παράστασης και θέσεων 8](#_Toc378370290)

[4.1.1 Περιγραφή και προτεραιότητα 8](#_Toc378370291)

[4.1.2 Ακολουθίες ερεθίσματος / απόκρισης 9](#_Toc378370292)

[4.1.3 Ανάλυση σε λειτουργικές απαιτήσεις 9](#_Toc378370293)

[4.2 Εκτύπωση εισιτηρίων 9](#_Toc378370294)

[4.2.1 Περιγραφή και προτεραιότητα 9](#_Toc378370295)

[4.2.2 Ακολουθίες ερεθίσματος / απόκρισης 9](#_Toc378370296)

[4.2.3 Ανάλυση σε λειτουργικές απαιτήσεις 9](#_Toc378370297)

[4.3 Καταχώρηση πληροφοριών Παράστασης 9](#_Toc378370298)

[4.3.1 Περιγραφή και προτεραιότητα 9](#_Toc378370299)

[4.3.2 Ακολουθίες ερεθίσματος / απόκρισης 10](#_Toc378370300)

[4.3.3 Ανάλυση σε λειτουργικές απαιτήσεις 10](#_Toc378370301)

[**5.**  **Μη λειτουργικές απαιτήσεις** 10](#_Toc378370302)

[5.1 Επιδόσεις 10](#_Toc378370303)

[5.2 Φυσική ασφάλεια 10](#_Toc378370304)

[5.3 Ασφάλεια πληροφορίας 10](#_Toc378370305)

[5.4 Ποιότητα λογισμικού 10](#_Toc378370306)

[5.5 Επιχειρησιακοί κανόνες 11](#_Toc378370307)

[**Περιπτώσεις Χρήσης** 12](#_Toc378370308)

[**Διάγραμμα Περιπτώσεων Χρήσης** 13](#_Toc378370309)

[**Κατάλογος Περιπτώσεων Χρήσης** 14](#_Toc378370310)

[1. **Document Management** 28](#_Toc378370311)

[1.1 Contributors 28](#_Toc378370312)

[1.2 Version Control 28](#_Toc378370313)

[**2.** **USER ACCEPTANCE TESTING** 29](#_Toc378370314)

[2.1 User Acceptance Definition 29](#_Toc378370315)

[2.2 UAT Responsibilities 29](#_Toc378370316)

[**3.** **UAT STRATEGY** 29](#_Toc378370317)

[**4. USER TESTING** 30](#_Toc378370318)

[**ΚΩΔΙΚΑΣ:** 31](#_Toc378370319)

[Play.cs Play Class 31](#_Toc378370320)

[seats.cs Seats Class 31](#_Toc378370321)

[**Form1.cs Main Window** 32](#_Toc378370322)

[Manage.cs Manager Window 37](#_Toc378370323)

[**seat\_map.cs Seat Map Window** 39](#_Toc378370324)

[ticket.cs Ticket Window 41](#_Toc378370325)

[**6.**  **ΜΟΝΤΕΛΟ ΑΝΑΛΥΣΗΣ** 45](#_Toc378370326)

[6.1.1  Αρχικό μενού (figure 1) 45](#_Toc378370327)

[6.1.2  Αρχικό μενού-Επιλογή Παράστασης (figure 2) 46](#_Toc378370328)

[6.2.1  Χάρτης θέσεων προς κράτηση (figure 3) 46](#_Toc378370329)

[6.2.2  Χάρτης θέσεων προς κράτηση- Διαδικασία κράτησης (figure 4) 48](#_Toc378370330)

[6.3.1  Προεπισκόπηση κράτησης εισιτηρίου (figure 5) 49](#_Toc378370331)

[6.3.2  Εκτύπωση/Κράτηση εισιτηρίου   (figure 6) 50](#_Toc378370332)

**Έντυπο Προδιαγραφών Απαιτήσεων Λογισμικού**

**για το Έργο**

**Σύστημα έκδοσης Θεατρικών Εισιτηρίων**

**Από:** Παναγιώτης - Νεκτάριος Δημητρακόπουλος

Χρήστος Πετρόπουλος

Γιώργος Χαλικιόπουλος

**<14/01/2014>**

**Αναθεωρήσεις**

|  |  |  |  |
| --- | --- | --- | --- |
| **Συγγραφέας** | **Ημερομηνία** | **Αίτια αναθεωρήσεων** | **Έκδοση κειμένου** |
| Παναγιώτης Δημητρακόπουλος | 20/01/2014 | Συμπλήρωση λειτουργικών απαιτήσεων - λειτουργιών | 1.03 |
| Χρήστος Πετρόπουλος | 23/01/2014 | Τελικές αλλαγές | 1.04 |

# 

# 

# 

# 

# 

# **1. Εισαγωγή**

## 1.1 Σκοπός του κειμένου

Η κύρια λειτουργία της εφαρμογής είναι η κράτηση θέσεων ενός θεάτρου για θεατρικές παραστάσεις μέσω του διαδικτύου. Ο σκοπός του παρόν εγγράφου είναι η ανάλυση και περιγραφή των λειτουργιών της εφαρμογής που μας ενδιαφέρει.

## 1.2 Συμβάσεις του κειμένου

Η γραμματοσειρά που χρησιμοποιήθηκε ήταν Arial με μέγεθος 12px.

## 1.3 Προορισμένο ακροατήριο και προτάσεις ανάγνωσης

Οι οδηγίες/λεπτομέρειες που παρουσιάζονται σε αυτό το έγγραφο προορίζονται για τον υπεύθυνο του θεάτρου, τους ταμίες που θα κλιθούν να κάνουν χρήση του κατά την διάρκεια της σεζόν και του Διευθύνων συμβούλου της εταιρείας.

## 1.4 Στόχος και εύρος του προϊόντος

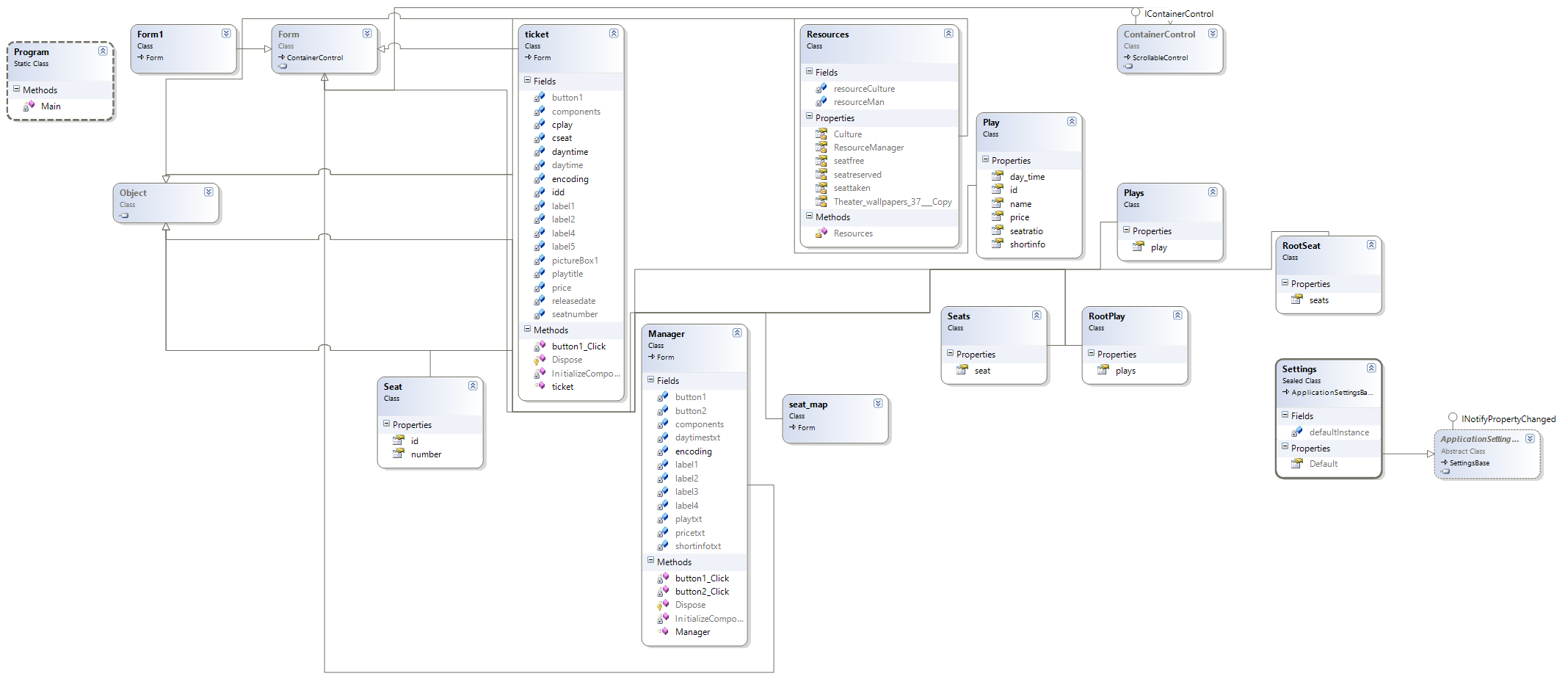
Όπως προαναφέρθηκε, το πρόγραμμα είναι μια απλή εφαρμογή κράτησης θέσεων θεάτρου μέσω διαδικτύου. Γενικότερος στόχος είναι η Ανάπτυξη νέου λογισμικού. Απευθύνεται στο προσωπικό του θεάτρου και έχει ως σκοπό την άμεση κράτηση των θέσεων καθώς και την διευκόλυνση της δράσης αυτής. Ο πελάτης μπορεί να κλείσει θέσεις είτε με την χρήση πιστωτικής κάρτας, είτε με παραλαβή από το θέατρο την ημέρα της παράστασης ή πιο πριν αλλά και να πληροφορηθεί για την κατανομή των θέσεων, ημέρες και ώρες παραστάσεων καθώς και διαθεσιμότητα των εισητηρίων.

# 2. **Γενική Περιγραφή**

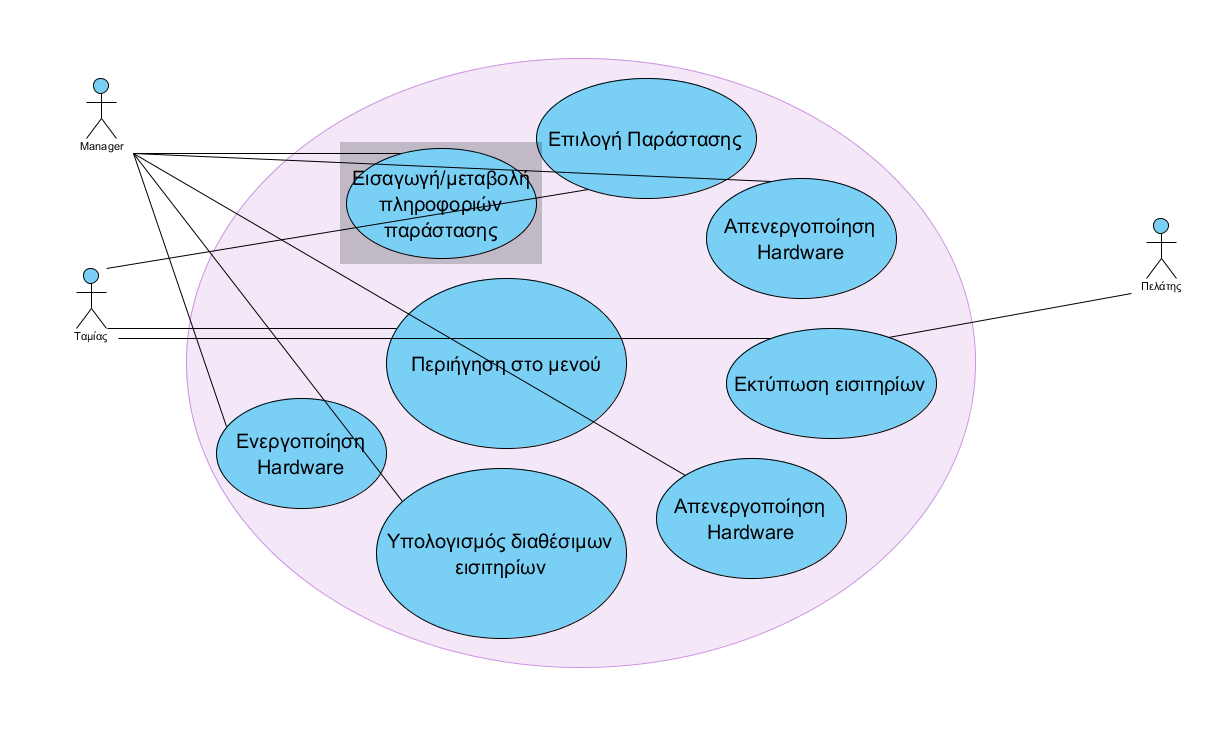
## 2.1 Προέλευση και προοπτική του προϊόντος

Το προϊόν αυτό είναι ένα νέο, ανεξάρτητο πρόγραμμα, ειδικά κατασκευασμένο για να καλύψει τις άμεσες ανάγκες ενός θεάτρου ή γενικότερα τις ανάγκες θεάτρων και κινηματογράφων Πρόκειται για την δουλειά μιας ομάδας ανεξάρτητων προγραμματιστών αφοσιωμένων στην διεκπεραίωση τέτοιων εφαρμογών που απευθύνονται κυρίως σε μικροεπιχειρήσεις.

### 2.2.1     Class Diagram



### 2.2.2     User Classes and Characteristics



Συστημα Έκδοσης Εισητηρίων

## 2.2 Γενική περιγραφή των λειτουργιών του προϊόντος

Οι λειτουργίες που αυτή η εφαρμογή καλείται να φέρει εις πέρας είναι:

1. Επιλογή θέσεων
2. Εκτύπωση Εισιτηρίων
3. Καταχώρηση των πληροφοριών μιας παράστασης

## 2.3 Κατηγορίες χρηστών

Οι κατηγορίες των χρηστών είναι δύο. Καθώς μιλάμε για θέατρο, αυτές οι κατηγορίες χωρίζονται στους ταμίες και τον υπεύθυνο του θεάτρου. Ο κάθε τομέας αναλαμβάνει και κάποια συγκεκριμένη λειτουργία από αυτές που αναφερθήκαμε στην προηγούμενη παράγραφο. Πιο συγκεκριμένα, οι ταμίες είναι υπεύθυνοι για την επιλογή των θέσεων και την εκτύπωση των εισιτηρίων ενώ ο υπεύθυνος του θεάτρου αναλαμβάνει την καταχώρηση των πληροφοριών μιας παράστασης.

## 2.4 Περιβάλλον λειτουργίας

Το λογισμικό αυτό θα λειτουργεί σε λειτουργικό σύστημα Windows XP, Vista, 7 και θα στηρίζεται σε Microsoft Windows Server. Η διεπαφή με τον χρήστη θα γίνεται μέσω οθόνης αφής είτε από υπολογιστή είτε από Microsoft Tablet.

## 2.5 Περιορισμοί στη σχεδίαση και την υλοποίηση

Κατά την σχεδίαση και την υλοποίηση του λογισμικού, σαν υπεύθυνοι σχεδίασης κληθήκαμε να προσέξουμε ορισμένους περιορισμούς σχετικά με τα εισιτήρια. Κάποιοι από αυτούς τους περιορισμούς ήταν ο αριθμός των θέσεων που πρέπει να είναι μέγιστος 123 θέσεις, να μπορεί να γίνει κράτηση από έναν πελάτη, η τιμή να καθορίζεται από τον αριθμό και την τοποθεσία των θέσεων και να υπάρχει ένδειξη για τα κομμένα εισιτήρια. Επίσης, το λογισμικό πρέπει να λειτουργεί κάτω από την πλατφόρμα των Windows(ΧΡ και μετά). Τέλος, γίνεται χρήση τοπικής σύνδεσης(LAN) στα 10ΜBps.

## 2.6 Τεκμηρίωση για το χρήστη

Θα παραδοθούν 3 σεμινάρια σχετικά με την εισαγωγή στο λογισμικό και την λειτουργία του, εκμάθηση πάνω σε αυτές τις λειτουργίες και την εξάσκηση πάνω σε αυτό. Επιπλέον, θα δοθεί ένα εγχειρίδιο χρήσης προκειμένου αν υπάρχουν απορίες να μπορεί ο χρήστης να ανατρέξει και να αναζητήσει περισσότερες πληροφορίες.

# **3. Απαιτήσεις για τις εξωτερικές διεπαφές**

## 3.1 Διεπαφή χρήστη

Η διεπαφή του ταμία και του υπευθύνου θα γίνεται μέσω γραφικού περιβάλλοντος με οθόνη αφής είτε από Η/Υ είτε από τάμπλετ.

## 3.2 Διεπαφές υλικού

Oι συσκευές που αλληλεπιδρούν με το πρόγραμμά μας είναι ο εκτυπωτής που θα μας βγάζει τα εισιτήρια και η οθόνη αφής.

# **4. Λειτουργίες συστήματος**

## 4.1 Επιλογή Παράστασης και θέσεων

### 4.1.1 Περιγραφή και προτεραιότητα

Είναι μια λειτουργία υψηλής προτεραιότητας γιατί καθορίζει την ορθή λειτουργία και βιωσιμότητα του θεάτρου. Εκτελείται από τους ταμίες μέσω του λογισμικού και της οθόνης αφής.

### 4.1.2 Ακολουθίες ερεθίσματος / απόκρισης

Ο χρήστης θα επιλέγει την λειτουργία που θέλει να εκτελέσει. Μπορεί να επιλέξει τον αριθμό θέσης, το πλήθος των θέσεων, την τοποθεσία τους, τίτλο παράστασης και μέρα και ώρα της αντίστοιχης παράστασης με το πάτημα του σχετικού “κουμπιού” στην εφαρμογή.

### 4.1.3 Ανάλυση σε λειτουργικές απαιτήσεις

1. **REQ-1**: Το λογισμικό θα πρέπει να περιέχει τις διαθέσιμες θέσεις.
2. **REQ-2**: Το λογισμικό να περιέχει τις διαθέσιμες παραστάσεις, την ώρα και την ημέρα.
3. **REQ-3**: Θα πρέπει να εμπεριέχεται μέσα στην βάση δεδομένων ο χάρτης του θεάτρου με τις θέσεις και το που βρίσκονται αυτές.
4. **REQ-4**: Αν δεν υπάρχει διαθεσιμότητα να εμφανίζεται σχετικό μήνυμα.
5. **REQ-5**: Αν γίνει λάθος επιλογή/πληκτρολόγηση από τον χρήστη να δίνεται σχετικό μήνυμα καθώς και η δυνατότητα αναίρεσης.

## 4.2 Εκτύπωση εισιτηρίων

### 4.2.1 Περιγραφή και προτεραιότητα

Είναι επίσης, μια λειτουργία υψηλής προτεραιότητας. Εκτελείται από τον ταμία και πρέπει να υπάρχει επικοινωνία μεταξύ υλικού υποστήριξης του λογισμικού και ενεργού εκτυπωτή.

### 4.2.2 Ακολουθίες ερεθίσματος / απόκρισης

Η λειτουργία που θα επιλέγεται θα είναι το τελικό αποτέλεσμα της προηγούμενης. Αφού ο χρήστης επιλέξει τα στοιχεία που θέλει και επιβεβαιώσει την ορθότητα τους θα στέλνει εντολή εκτύπωσης των εισιτηρίων στον εκτυπωτή.

### 4.2.3 Ανάλυση σε λειτουργικές απαιτήσεις

1. **REQ-1**: Το λογισμικό θα πρέπει να συνδέεται με έναν ενεργό εκτυπωτή.
2. **REQ-2**: Μήνυμα αν η μεταφορά έγινε επιτυχώς.
3. **REQ-3**: Μήνυμα με την ολοκλήρωση της εκτύπωσης.
4. **REQ-4**: Ο εκτυπωτής θα πρέπει να υποστηρίζεται από το υλικό και το λογισμικό.

## 4.3 Καταχώρηση πληροφοριών Παράστασης

### 4.3.1 Περιγραφή και προτεραιότητα

Είναι επίσης, μια λειτουργία μέσης προτεραιότητας. Εκτελείται από τον υπεύθυνο του θεάτρου αλλά εδώ υπάρχουν ορισμένα δικαιώματα τα οποία έχουν δοθεί μόνο σε αυτόν. Πρόκειται για την καταχώρηση πληροφοριών μιας παράστασης όπως το είδος της, μια περίληψη του έργου, ώρες και μέρες που παίζεται και διαθεσιμότητα εισιτηρίων.

### 4.3.2 Ακολουθίες ερεθίσματος / απόκρισης

Ο χρήστης θα εισάγει στοιχεία σχετικά με το εισιτήριο και την παράσταση με ειδική πρόσβαση στο λογισμικό, περιορισμένη μόνο σε εκείνον. Έτσι δίνεται η δυνατότητα μεταβολής μεταβλητών και στοιχείων σχετικών με την παράσταση.

### 4.3.3 Ανάλυση σε λειτουργικές απαιτήσεις

1. **REQ-1**: Θα πρέπει να δοθούν ειδικά δικαιώματα χρήσης.
2. **REQ-2**: Άμεση σύνδεση με το υλικό των υπολοίπων συσκευών που συνδέονται στο σύστημα. .
3. **REQ-3**: Μήνυμα επιβεβαίωσης αλλαγών.

# **5. Μη λειτουργικές απαιτήσεις**

## 5.1 Επιδόσεις

Οι κυριότερες απαιτήσεις για το λογισμικό μας είναι ότι η συναλλαγή πρέπει να διαρκεί κάτω από πέντε(5) δευτερόλεπτα και να είναι ικανό να λειτουργεί ικανοποιητικά για δύο(2) ταυτόχρονους χρήστες.

## 5.2 Φυσική ασφάλεια

Το λογισμικό αυτό δεν πρέπει να τροποποιείται στις λειτουργίες του με κανέναν τρόπο. Το υλικό που χρησιμοποιείται για την διεπαφή με τον χρήστη καθώς και τα περιφερειακά που συνδέονται με το λογισμικό θα πρέπει να προστατεύονται και να συντηρούνται ανά τακτά χρονικά διαστήματα για την ορθή λειτουργία.

## 5.3 Ασφάλεια πληροφορίας

Καθώς πρόκειται για τοπική σύνδεση δεν υπάρχει κίνδυνος διαρροής πληροφοριών εκτός του συστήματος του θεάτρου. Ωστόσο, θα πρέπει να μην επιτρέπεται η πρόσβαση στο λογισμικό και το υλικό του σε τρίτους για την διαφύλαξη των δεδομένων.

## 5.4 Ποιότητα λογισμικού

Το λογισμικό που παραδίδεται είναι αρκετά ελαφρύ και φορητό αφού μπορεί να χρησιμοποιηθεί από φορητούς υπολογιστές (λάπτοπ, ταμπλετ κλπ) με ευκολία. Η συντήρηση του είναι εύκολη. Σε περίπτωση, λαθών ή δυσλειτουργιών του προγράμματος θα υπάρχουν οι απαραίτητες αναβαθμίσεις που θα περιέχουν επιδιορθώσεις τυχόν σφαλμάτων.

## 5.5 Επιχειρησιακοί κανόνες

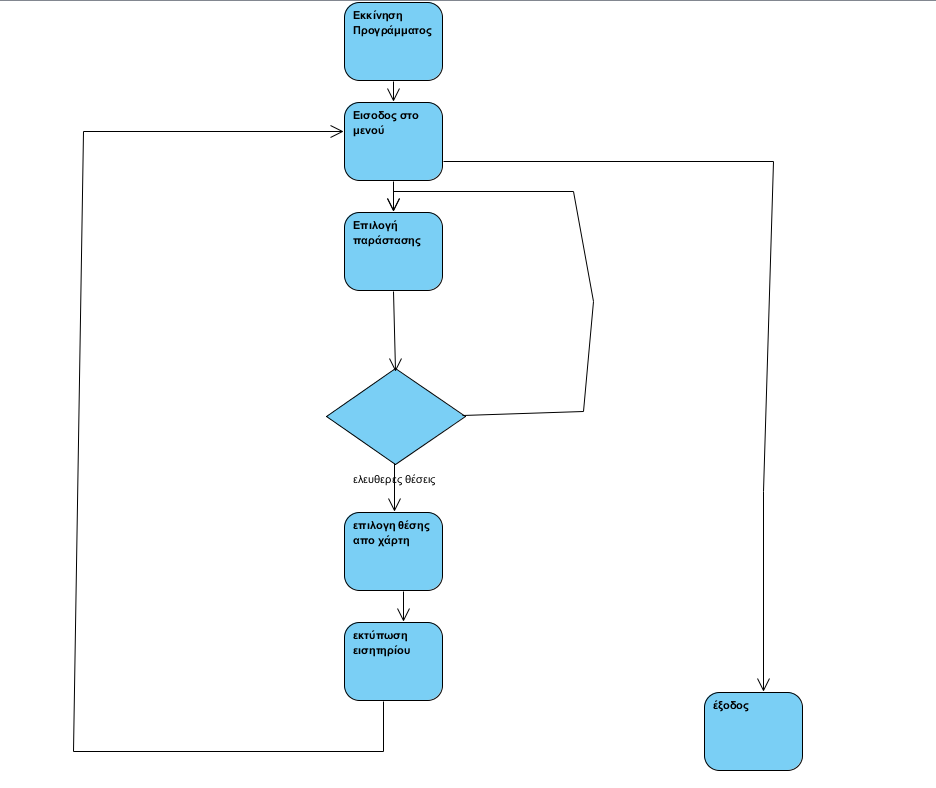
Για τον κάθε χρήστη θα υπάρχουν και τα σχετικά δικαιώματα χρήσης. Για τους ταμίες θα είναι δυνατή η χρήση μόνο για την κράτηση των θέσεων και την εκτύπωση των εισιτηρίων ενώ για τον υπεύθυνο θα δοθεί η δυνατότητα μεταβολής των πληροφοριών σε εισιτήρια ή/και παραστάσεις.

# **Περιπτώσεις Χρήσης**

**Αναθεωρήσεις**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ονοματεπώνυμο** | **Ημερομηνία** | **Αιτία Αναθεώρησης** | **Έκδοση** |
| Παναγιώτης Δημητρακόπουλος | 19/01/2014 | Διόρθωση κομματιών του εγγράφου | 1.01 |
|  |  |  |  |
|  |  |  |  |

# **Διάγραμμα Περιπτώσεων Χρήσης**



# 

# 

# 

# 

# 

# 

# 

# **Κατάλογος Περιπτώσεων Χρήσης**

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι** | **Περιπτώσεις Χρήσης** |
| Ταμίας | ΖΧ-1. Περιήγηση στο θεατρικό χάρτη  ΖΧ-2. Περιήγηση στο μενού  ΖΧ-3. Επιλογή Παράστασης  ΖΧ-4. Επιλογή Θέσεων  ΖΧ-5. Εκτύπωση εισιτηρίων |
| Υπεύθυνος | ΖΧ-6. Ενεργοποίηση Hardware  ΖΧ-7. Υπολογισμός διαθέσιμων εισιτηρίων  ΖΧ-8. Εισαγωγή/μεταβολή πληροφοριών παράστασης  ΖΧ-9. Απενεργοποίηση Hardware |

**Πρότυπο Περιπτώσεως Χρήσης**

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-1 |  |  |
| **Ονομασία:** | Περιήγηση στο θεατρικό χάρτη |  |  |
| **Δημιουργήθηκε από:** | Παναγιώτης  Δημητρακόπουλος | **Τελευταία ενημέρωση από:** | Παναγιώτης  Δημητρακόπουλος |
| **Ημερομηνία Συγγραφής:** | 19/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 19/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Ταμίας |
| **Περιγραφή:** | Ο ταμίας μπορεί να έχει πρόσβαση στο χάρτη των θέσεων του θεάτρου |
| **Γεγονός Εκκίνησης:** | Ο ταμίας επιλέγει προβολή θέσεων |
| **Προϋποθέσεις:** | 1. Φόρτωση του χάρτη στο σύστημα |
| **Τελική Κατάσταση:** | 1. Επιλογή του χάρτη 2. Εμφάνιση στην κύρια οθόνη μαζί με τις επιπρόσθετες πληροφορίες |
| **Φυσιολογική Ροή:** | 1. ΖΧ-2 2. Ο ταμίας επιλέγει την προβολή του χάρτη 3. Το σύστημα φορτώνει τον διαθέσιμο χάρτη από την βάση δεδομένων 4. Εμφάνιση του χάρτη στην οθόνη |
| **Εναλλακτική Ροή:** | 1. ΖΧ-2 2. Ο ταμίας επιλέγει την προτίμηση αυτή 3. ΖΧ-5 |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | ΖΧ-1, ΖΧ-2, ΖΧ-5 |
| **Προτεραιότητα:** | 2 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 121 χρήσεις ανά ημέρα |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να υπάρχει αναβαθμισμένη έκδοση του χάρτη μετά από κάθε κράτηση |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-2 |  |  |
| **Ονομασία:** | Περιήγηση στο μενού |  |  |
| **Δημιουργήθηκε από:** | Παναγιώτης  Δημητρακόπουλος | **Τελευταία ενημέρωση από:** | Παναγιώτης  Δημητρακόπουλος |
| **Ημερομηνία Συγγραφής:** | 19/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 19/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Ταμίας |
| **Περιγραφή:** | Ο ταμίας μπορεί να έχει πρόσβαση στο μενού του λογισμικού |
| **Γεγονός Εκκίνησης:** | Μετάβαση σε κανονική λειτουργία (ανάδραση οθόνης), εκκίνηση Η/Υ, ενεργοποίηση οθόνης |
| **Προϋποθέσεις:** | 1. Επαφή οθόνης με Η/Υ, ύπαρξη ηλεκτρικής ενέργειας |
| **Τελική Κατάσταση:** | 1. Μετάβαση στο κεντρικό μενού   2. Επιλογή διαθέσιμων παραθύρων |
| **Φυσιολογική Ροή:** | 1. ΖΧ-6 2. Ο ταμίας ενεργοποιεί την οθόνη 3. Πραγματοποιεί περιήγηση στο κεντρικό μενού |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | ΖΧ-2, ΖΧ-6 |
| **Προτεραιότητα:** | 3 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 3 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το Software |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-3 |  |  |
| **Ονομασία:** | Επιλογή παράστασης |  |  |
| **Δημιουργήθηκε από:** | Γιώργος Χαλικιόπουλος | **Τελευταία ενημέρωση από:** | Γιώργος Χαλικιόπουλος |
| **Ημερομηνία Συγγραφής:** | 21/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 21/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Ταμίας |
| **Περιγραφή:** | Ο ταμίας επιλέγει την παράσταση. |
| **Γεγονός Εκκίνησης:** | Ανακοίνωση της παράστασης, πελάτης στο ταμείο |
| **Προϋποθέσεις:** | 1. Επαφή οθόνης με Η/Υ, ύπαρξη ηλεκτρικής ενέργειας  2. Διαθεσιμότητα θέσεων. |
| **Τελική Κατάσταση:** | 1. Μετάβαση στο μενού εισιτηρίων |
| **Φυσιολογική Ροή:** | 1. ΖΧ-8  2. Ο ταμίας επιλέγει την παράσταση που το ζήτησε ο  πελάτης |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | ΖΧ-3, ΖΧ-8 |
| **Προτεραιότητα:** | 4 |
| **Συχνότητα χρήσης:** | Μέγιστο 123 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | 1. Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το  Software  2. Να υπάρχουν διαθέσιμα εισιτήρια. |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-4 |  |  |
| **Ονομασία:** | Επιλογή εισιτήριων |  |  |
| **Δημιουργήθηκε από:** | Γιώργος Χαλικιόπουλος | **Τελευταία ενημέρωση από:** | Γιώργος Χαλικιόπουλος |
| **Ημερομηνία Συγγραφής:** | 21/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 21/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Ταμίας |
| **Περιγραφή:** | Ο ταμίας μπορεί να επιλέξει τις θέσεις για την παράσταση. |
| **Γεγονός Εκκίνησης:** | Επιλογή παράστασης |
| **Προϋποθέσεις:** | 1. Να έχει επιλεχθεί η παράσταση |
| **Τελική Κατάσταση:** | 1. Επιλογή θέσεων   1. Μη διαθέσιμες θέσεις |
| **Φυσιολογική Ροή:** | 1. ΖΧ-6 2. Ο ταμίας ενεργοποιεί την οθόνη 3. Πραγματοποιεί περιήγηση στο κεντρικό μενού |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | ΖΧ-2, ΖΧ-6 |
| **Προτεραιότητα:** | 5 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 123 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το Software |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-5 |  |  |
| **Ονομασία:** | Εκτύπωση Εισιτηρίων |  |  |
| **Δημιουργήθηκε από:** | Γιώργος Χαλικιόπουλος | **Τελευταία ενημέρωση από:** | Γιώργος Χαλικιόπουλος |
| **Ημερομηνία Συγγραφής:** | 21/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 21/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Ταμίας |
| **Περιγραφή:** | Εκτυπώνονται τα εισιτήρια για την επιλεχθείσα παράσταση |
| **Γεγονός Εκκίνησης:** | Επιλογή εισιτηρίων |
| **Προϋποθέσεις:** | 1. Να έχουν επιλεχθεί τα εισιτήρια  2. Να έχει πληρώσει ο πελάτης το εισιτήριο. |
| **Τελική Κατάσταση:** | 1. Παράδοση του εισιτηρίου στον πελάτη. |
| **Φυσιολογική Ροή:** | 1. ΖΧ-3  2. ZX-4  3. ZX-7  4. Εκτύπωση Εισιτηρίων και παράδοση τους στον πελάτη. |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | - |
| **Προτεραιότητα:** | 7 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 123 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το Software |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-6 |  |  |
| **Ονομασία:** | Ενεργοποίηση Hardware |  |  |
| **Δημιουργήθηκε από:** | Γιώργος Χαλικιόπουλος | **Τελευταία ενημέρωση από:** | Γιώργος Χαλικιόπουλος |
| **Ημερομηνία Συγγραφής:** | 21/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 21/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Υπεύθυνος |
| **Περιγραφή:** | Ο υπεύθυνος στην αρχή της βάρδιας του ανοίγει το μηχάνημα |
| **Γεγονός Εκκίνησης:** | Έναρξη βάρδιας |
| **Προϋποθέσεις:** | 1. Ύπαρξη ηλεκτρικής ενέργειας |
| **Τελική Κατάσταση:** | 1. Ενεργοποιημένο σύστημα |
| **Φυσιολογική Ροή:** | 1. ΖΧ-3  2. Έλεγχος εισιτηρίων. |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | ΖΧ-2, ΖΧ-6 |
| **Προτεραιότητα:** | 1 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 2 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το Software |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-7 |  |  |
| **Ονομασία:** | Υπολογισμός διαθέσιμων εισιτηρίων |  |  |
| **Δημιουργήθηκε από:** | Γιώργος Χαλικιόπουλος | **Τελευταία ενημέρωση από:** | Γιώργος Χαλικιόπουλος |
| **Ημερομηνία Συγγραφής:** | 21/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 21/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Υπεύθυνος |
| **Περιγραφή:** | Ο υπεύθυνος ελέγχει αν υπάρχουν διαθέσιμα εισιτήρια. |
| **Γεγονός Εκκίνησης:** | Επιλογή εισιτηρίων. |
| **Προϋποθέσεις:** | 1. Ύπαρξη ηλεκτρικής ενέργειας.  2. Ζήτηση για τα εισιτήρια. |
| **Τελική Κατάσταση:** | 1. Εμφάνιση διαθεσιμότητας εισιτηρίων |
| **Φυσιολογική Ροή:** | 1. Άνοιγμα υπολογιστή. |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | ΖΧ-3, ΖΧ-7 |
| **Προτεραιότητα:** | 6 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 123 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το Software |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-8 |  |  |
| **Ονομασία:** | Εισαγωγή/μεταβολή πληροφοριών παράστασης |  |  |
| **Δημιουργήθηκε από:** | Γιώργος Χαλικιόπουλος | **Τελευταία ενημέρωση από:** | Γιώργος Χαλικιόπουλος |
| **Ημερομηνία Συγγραφής:** | 21/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 21/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Υπεύθυνος |
| **Περιγραφή:** | Ο υπεύθυνος εισάγει ή τροποποιεί τα δεδομένα της παράστασης |
| **Γεγονός Εκκίνησης:** | Ενεργοποίηση Hardware. |
| **Προϋποθέσεις:** | 1. Η μετατροπή/εισαγωγή των δεδομένων να είναι πριν την ημερομηνία της παράστασης  2. Ζήτηση για τα εισιτήρια. |
| **Τελική Κατάσταση:** | 1. Εμφάνιση νέων πληροφοριών παράστασης |
| **Φυσιολογική Ροή:** | 1. ΖΧ-6  2. ΖΧ-8  3. Αλλαγή στα δεδομένα της παράστασης. |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | ΖΧ-6, ΖΧ-8 |
| **Προτεραιότητα:** | 6 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 123 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το Software |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Κωδικός Περίπτωσης:** | ΖΧ-9 |  |  |
| **Ονομασία:** | Απενεργοποίηση Hardware |  |  |
| **Δημιουργήθηκε από:** | Γιώργος Χαλικιόπουλος | **Τελευταία ενημέρωση από:** | Γιώργος Χαλικιόπουλος |
| **Ημερομηνία Συγγραφής:** | 21/1/2014 | **Ημερομηνία τελευταίας ενημέρωσης:** | 21/1/2014 |

|  |  |
| --- | --- |
| **Εμπλεκόμενοι Ρόλοι:** | Υπεύθυνος |
| **Περιγραφή:** | Ο υπεύθυνος απενεργοποιεί το Hardware. |
| **Γεγονός Εκκίνησης:** | Απενεργοποίηση του Hardware. |
| **Προϋποθέσεις:** | 1. Ενεργοποιημένο Hardware. |
| **Τελική Κατάσταση:** | 1. Κλειστο σύστημα. |
| **Φυσιολογική Ροή:** | 1.  2.  3. |
| **Εναλλακτική Ροή:** | - |
| **Εξαιρέσεις:** | - |
| **Ενσωματώνει:** | - |
| **Προτεραιότητα:** | 9 |
| **Συχνότητα χρήσης:** | Μέγιστο περίπου 1 χρήσεις την ημέρα. |
| **Business Rules:** | - |
| **Ειδικές απαιτήσεις:** | Να είναι ενεργοποιημένο το Hardware και διαθέσιμο το Software |
| **Υποθέσεις:** | - |
| **Σημειώσεις και ζητήματα:** |  |

**Stage: Acceptance**

**User Acceptance Test (UAT) Plan**

**Document Version: 1.2**

**Date: 07-Nov-2007**

# **Document Management**

## 1.1 Contributors

Please provide details of all contributors to this document.

|  |  |  |
| --- | --- | --- |
| **Role** | **Department** | **Name** |
| Programmer | Development | Christos Petropoulos |
| Tester | QA | George Halikiopoulos |
| Writer | Report | Panos Dimitrakopoulos |
|  |  |  |

## 1.2 Version Control

Please document all changes made to this document since initial distribution.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Section** | **Amendment** |
| 15/01/2014 | 1.01 | Panos D. |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# **USER ACCEPTANCE TESTING**

## 2.1 User Acceptance Definition

User Acceptance Testing should ensure that the application performs at an acceptable level for the Customer.

## 2.2 UAT Responsibilities

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Responsibilities** |
| Ταμίας | Το όνομα του ταμία | Εκτύπωση εισιτήριων, Έλεγχος θέσεων, Επιλογή παράστασης, Επιλογή εισιτηρίου. |
| Υπεύθυνος |  | Άνοιγμα Hardware, Κλείσιμο Hardware, Καταχώρηση παραστάσεων. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **UAT STRATEGY**

|  |  |  |
| --- | --- | --- |
| **Consideration** | **Date Agreed** | **Details** |
| Test Approach | 20/01/2014 |  |
| Assumptions & Constraints | 20/01/2014 |  |
| Setup of Test Environment | 21/01/2014 |  |
| Test Scenarios | 22/01/2014 |  |
| Test Period | 22/01/2014 |  |

# **4. USER TESTING**

The following explicit tests cover the main areas of work as load out in section 3 above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **1.1** | Kράτηση Εισητηριου απο την εφαρμογή και επιβεβέωση της ύπαρξης του στη βάση | Να έχει καταχωρηθει το εισητηριο στη βάση |  | Pass |
| **1.2** | Φόρτωση των Παραστάσεων στην εφαρμογή απο τη βάση | Φόρτωση των Παραστάσεων |  | Pass |
| **1.3** |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **2.1** | Ελεγχος Θέσεων εκάστοτε παράστασης | Εμφάνιση ελέυθερων θέσεων |  | Pass |
| **2.2** | Εμφάνιση Χάρτη θέσεων εκάστοτε παράστασης | Εμφάνιση χάρτη με κοκκινισμένες τις πιασμένες |  | Pass |
| **2.3** | Καταχώρηση Νέων Παραστάσεων | Καταχωρηση των παραστάσεων στη βάση |  | Fail |

# **ΚΩΔΙΚΑΣ:**

|  |
| --- |
| Play.cs Play Class   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;    namespace TheaterTickets.classes  {     public class Play     {         public string id { get; set; }         public string name { get; set; }         public string day\_time { get; set; }         public string shortinfo { get; set; }         public string seatratio { get; set; }         public string price { get; set; }     }       public class Plays     {         public List<Play> play { get; set; }     }       public class RootPlay     {         public Plays plays { get; set; }     }  }   seats.cs Seats Class   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;    namespace TheaterTickets.classes  {     public class Seat     {         public string id { get; set; }         public string number { get; set; }     }       public class Seats     {         public List<Seat> seat { get; set; }     }       public class RootSeat     {         public Seats seats { get; set; }     }  }   **Form1.cs Main Window**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Windows.Forms;  using System.Net;  using System.IO;  using Newtonsoft.Json;  using TheaterTickets.classes;    namespace TheaterTickets  {     public partial class Form1 : Form     {         public static RootSeat rootseat = new RootSeat();         public static RootPlay rootplay = new RootPlay();         List<String> temp = new List<String>();         string id;             public Form1()         {             InitializeComponent();             //listBox1.SelectionMode =SelectionMode.One;             //listBox1.Items.Add("JANIS JOPLIN:ΔΩΜΑΤΙΟ 105");             //listBox1.Items.Add("4.48 ΕΝΑ ΤΡΑΓΟΥΔΙ");             //listBox1.Items.Add("Ο ΘΕΟΣ ΒΑΡΙΕΤΑΙ ΤΩΡΑ ΤΕΛΕΥΤΑΙΑ");             //listBox1.Items.Add("ΒΡΥΚΌΛΑΚΕΣ");             //listBox1.Items.Add("BEDTIME STORIES");             //listBox1.Items.Add("Η ΑΥΞΗΣΗ");               //listBox2.SelectionMode = SelectionMode.One;             //listBox2.Items.Add("10-12");             //listBox2.Items.Add("12-2");             //listBox2.Items.Add("2-4");             //listBox2.Items.Add("4-6");             //listBox2.Items.Add("6-8");             //listBox2.Items.Add("9-11");             HttpWebRequest req = (HttpWebRequest)WebRequest.Create("http://83.212.118.71/theaters/get.php");             req.Method = "GET";             if (req.Headers == null)             {                 req.Headers = new WebHeaderCollection();             }            // req.Headers[HttpRequestHeader.IfModifiedSince] = DateTime.UtcNow.ToString();               req.BeginGetResponse(new AsyncCallback(getresponseforplays), req);           }           void getresponseforplays(IAsyncResult MyResultAsync)         {             HttpWebResponse response;             do             {                 HttpWebRequest request = (HttpWebRequest)MyResultAsync.AsyncState;                   response = (HttpWebResponse)request.EndGetResponse(MyResultAsync);             }             while (response.StatusCode != HttpStatusCode.OK);               if (response.StatusCode == HttpStatusCode.OK && response.ContentLength > 0)             {                 using (StreamReader sr = new StreamReader(response.GetResponseStream()))                 {                     string result = sr.ReadToEnd();                       rootplay = JsonConvert.DeserializeObject<RootPlay>(result);                 BeginInvoke((Action)(() =>                          {                       for (int i = 0; i < rootplay.plays.play.Count; i++)                     {                         if (temp.Contains(rootplay.plays.play[i].name) )continue;                        temp.Add(rootplay.plays.play[i].name);                     }                         temp.Distinct().ToList();                     foreach (string th in temp)                     {                         if (listBox1.Items.Contains(th)) continue;                           listBox1.Items.Add(th);                     }                       for (int i = 0; i < rootplay.plays.play.Count; i++)                     {                         if(rootplay.plays.play[i].name==rootplay.plays.play[0].name)listBox2.Items.Add(rootplay.plays.play[i].day\_time);                     }                    // listBox1.Items.Add(temp);                          }));                   }             }           }           private void button1\_Click(object sender, EventArgs e)         {             seat\_map seatmap = new seat\_map(listBox1.SelectedItem.ToString(), listBox2.SelectedItem.ToString(),rootseat,rootplay);             seatmap.Text = listBox1.SelectedItem.ToString();             seatmap.Tag = listBox1.SelectedItem.ToString();               seatmap.Show();             Hide();         }           private void play\_changed(object sender, EventArgs e)         {               int index = listBox1.SelectedIndex;             //desc.Text = rootplay.plays.play.Where(x => x.name == temp[index]).First().shortinfo;             // id = rootplay.plays.play.Where(x => x.id == temp[index]).First().id;             listBox2.Items.Clear();             for (int i = 0; i < rootplay.plays.play.Count; i++)             {                 if (rootplay.plays.play[i].name == temp[index]) listBox2.Items.Add(rootplay.plays.play[i].day\_time);             }             }           void getresponseforseats(IAsyncResult MyResultAsync)         {             HttpWebResponse response;             do             {                 HttpWebRequest request = (HttpWebRequest)MyResultAsync.AsyncState;                   response = (HttpWebResponse)request.EndGetResponse(MyResultAsync);             }             while (response.StatusCode != HttpStatusCode.OK);               if (response.StatusCode == HttpStatusCode.OK && response.ContentLength > 0)             {                 using (StreamReader sr = new StreamReader(response.GetResponseStream()))                 {                     string result = sr.ReadToEnd();                       rootseat = JsonConvert.DeserializeObject<RootSeat>(result);                     BeginInvoke((Action)(() =>                     {                         rootseat.seats.seat = rootseat.seats.seat.Where(x => x.id == id).ToList();                         availableseats.Text = (122 - rootseat.seats.seat.Count).ToString()+"/122";                       }));                   }             }           }           private void day\_time\_changed(object sender, EventArgs e)         {             id = rootplay.plays.play.Where(x => x.day\_time == listBox2.SelectedItem).First().id;             desc.Text = rootplay.plays.play.Where(x => x.day\_time == listBox2.SelectedItem).First().shortinfo;               HttpWebRequest req = (HttpWebRequest)WebRequest.Create("http://83.212.118.71/theaters/getseats.php");             req.Method = "GET";             if (req.Headers == null)             {                 req.Headers = new WebHeaderCollection();             }             // req.Headers[HttpRequestHeader.IfModifiedSince] = DateTime.UtcNow.ToString();               req.BeginGetResponse(new AsyncCallback(getresponseforseats), req);           }           private void button2\_Click(object sender, EventArgs e)         {             Manager manager = new Manager();             manager.Show();             this.Hide();         }     }  }   Manage.cs Manager Window   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Windows.Forms;  using System.Net;  using System.IO;    namespace TheaterTickets  {     public partial class Manager : Form     {           ASCIIEncoding encoding = new ASCIIEncoding();             public Manager()         {             InitializeComponent();         }             private void button1\_Click(object sender, EventArgs e)         {             MessageBox.Show("Εισαγωγή: πχ. Δευτέρα,14-16  για πολλαπλες παραστάσεις: πχ: Δευτερα,10-12,Τρίτη,13-17");         }           private void button2\_Click(object sender, EventArgs e)         {               if (playtxt.Text == "" || daytimestxt.Text == "" || shortinfotxt.Text == "" || pricetxt.Text == "")             {                 MessageBox.Show("Παρακαλώ συμπληρώστε όλα τα στοιχεία πριν καταχωρήσετε την παράσταση.");             }else             {                 string[] words=daytimestxt.Text.Split(',');                 for (int i = 0; i < words.Count(); i += 2)                 {                     HttpWebRequest req = (HttpWebRequest)WebRequest.Create("http://83.212.118.71/theaters/Add.php");                     req.Method = "POST";                     req.ContentType = "application/x-www-form-urlencoded";                       string forupload;                     string[] linesfortext;                       string postData = "name="+playtxt.Text+"&shortinfo="+shortinfotxt.Text+"&day\_time="+ words[i]+"\_"+words[i+1]+"&seatratio=1&price="+pricetxt.Text;                       //byte[] data = encoding.GetBytes(postData);                     byte[] data = Encoding.UTF8.GetBytes(postData);                     req.ContentLength = data.Length;                       using (Stream stream = req.GetRequestStream())                     {                         stream.Write(data, 0, data.Length);                     }                       HttpWebResponse response = (HttpWebResponse)req.GetResponse();                       string responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();                       if (responseString == "1")                     {                         MessageBox.Show("Η καταχώρηση ολοκληρώθηκε.");                         Form1 newform1 = new Form1();                         newform1.Show();                           Close();                     }                     else                     {                         MessageBox.Show("H καταχώρηση δεν ολοκληρώθηκε,δοκιμάστε ξανά η ενημερώστε τον υπέυθυνο για το θέμα.");                         Form1 newform1 = new Form1();                         newform1.Show();                           Close();                     }                   }               }         }     }  } **seat\_map.cs Seat Map Window**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Windows.Forms;  using System.Net;  using System.IO;    namespace TheaterTickets  {     public partial class Manager : Form     {           ASCIIEncoding encoding = new ASCIIEncoding();             public Manager()         {             InitializeComponent();         }             private void button1\_Click(object sender, EventArgs e)         {             MessageBox.Show("Εισαγωγή: πχ. Δευτέρα,14-16  για πολλαπλες παραστάσεις: πχ: Δευτερα,10-12,Τρίτη,13-17");         }           private void button2\_Click(object sender, EventArgs e)         {               if (playtxt.Text == "" || daytimestxt.Text == "" || shortinfotxt.Text == "" || pricetxt.Text == "")             {                 MessageBox.Show("Παρακαλώ συμπληρώστε όλα τα στοιχεία πριν καταχωρήσετε την παράσταση.");             }else             {                 string[] words=daytimestxt.Text.Split(',');                 for (int i = 0; i < words.Count(); i += 2)                 {                     HttpWebRequest req = (HttpWebRequest)WebRequest.Create("http://83.212.118.71/theaters/Add.php");                     req.Method = "POST";                     req.ContentType = "application/x-www-form-urlencoded";                       string forupload;                     string[] linesfortext;                       string postData = "name="+playtxt.Text+"&shortinfo="+shortinfotxt.Text+"&day\_time="+ words[i]+"\_"+words[i+1]+"&seatratio=1&price="+pricetxt.Text;                       //byte[] data = encoding.GetBytes(postData);                     byte[] data = Encoding.UTF8.GetBytes(postData);                     req.ContentLength = data.Length;                       using (Stream stream = req.GetRequestStream())                     {                         stream.Write(data, 0, data.Length);                     }                       HttpWebResponse response = (HttpWebResponse)req.GetResponse();                       string responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();                       if (responseString == "1")                     {                         MessageBox.Show("Η καταχώρηση ολοκληρώθηκε.");                         Form1 newform1 = new Form1();                         newform1.Show();                           Close();                     }                     else                     {                         MessageBox.Show("H καταχώρηση δεν ολοκληρώθηκε,δοκιμάστε ξανά η ενημερώστε τον υπέυθυνο για το θέμα.");                         Form1 newform1 = new Form1();                         newform1.Show();                           Close();                     }                   }               }         }     }  }   ticket.cs Ticket Window   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Windows.Forms;  using System.Net;  using System.IO;    namespace TheaterTickets  {     public partial class ticket : Form     {         ASCIIEncoding encoding = new ASCIIEncoding();           string idd;         string dayntime;         string cplay;         string cseat;         public ticket(string currplay,string day\_time,string id,string currseat,string baseprice)         {             InitializeComponent();             idd = id;             dayntime = day\_time;             cseat = currseat;               releasedate.Text = "Ημερομηνία έκδοσης" + DateTime.Now;             playtitle.Text = currplay;             daytime.Text = day\_time;             seatnumber.Text = currseat;             if (Convert.ToInt32(currseat) <= 10)             {                 double ma = 0.4;                 double extra=(Convert.ToDouble(baseprice) \* ma);                 price.Text = (Convert.ToDouble(baseprice) +extra ).ToString() + "€";             }else               if (Convert.ToInt32(currseat) <= 20)                 {                     double ma = 0.35;                     double extra = (Convert.ToDouble(baseprice) \* ma);                     price.Text = (Convert.ToDouble(baseprice) + extra).ToString() + "€";                  }             else                   if (Convert.ToInt32(currseat) <= 30)                   {                       double ma = 0.30;                       double extra = (Convert.ToDouble(baseprice) \* ma);                       price.Text = (Convert.ToDouble(baseprice) + extra).ToString() + "€";                   }                   else                       if (Convert.ToInt32(currseat) <= 40)                       {                           double ma = 0.25;                           double extra = (Convert.ToDouble(baseprice) \* ma);                           price.Text = (Convert.ToDouble(baseprice) + extra).ToString() + "€";                       }                       else                           if (Convert.ToInt32(currseat) <= 50)                           {                               double ma = 0.20;                               double extra = (Convert.ToDouble(baseprice) \* ma);                               price.Text = (Convert.ToDouble(baseprice) + extra).ToString() + "€";                           }                           else                               if (Convert.ToInt32(currseat) <= 60)                               {                                   double ma = 0.15;                                   double extra = (Convert.ToDouble(baseprice) \* ma);                                   price.Text = (Convert.ToDouble(baseprice) + extra).ToString() + "€";                               }                               else                           if (Convert.ToInt32(currseat) <= 72)                           {                               double ma = 0.10;                               double extra = (Convert.ToDouble(baseprice) \* ma);                               price.Text = (Convert.ToDouble(baseprice) + extra).ToString() + "€";                           }         }           private void button1\_Click(object sender, EventArgs e)         {             HttpWebRequest req = (HttpWebRequest)WebRequest.Create("http://83.212.118.71/theaters/Addseats.php");             req.Method = "POST";             req.ContentType = "application/x-www-form-urlencoded";               string forupload;             string[] linesfortext;               string postData = "id=" + idd + "&number=" +cseat;               byte[] data = encoding.GetBytes(postData);             req.ContentLength = data.Length;               using (Stream stream = req.GetRequestStream())             {                 stream.Write(data, 0, data.Length);             }               HttpWebResponse response = (HttpWebResponse)req.GetResponse();               string responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();               if (responseString == "1")             {                 MessageBox.Show("Η καταχώρηση ολοκληρώθηκε.");                 Form1 newform1 = new Form1();                 newform1.Show();                   Close();             }             else             {                 MessageBox.Show("H καταχώρηση δεν ολοκληρώθηκε,δοκιμάστε ξανά η ενημερώστε τον υπέυθυνο για το θέμα.");                 Form1 newform1 = new Form1();                 newform1.Show();                   Close();               }             // textBox1.Text = responseString;                 }     } |

# **6. ΜΟΝΤΕΛΟ ΑΝΑΛΥΣΗΣ**

## 

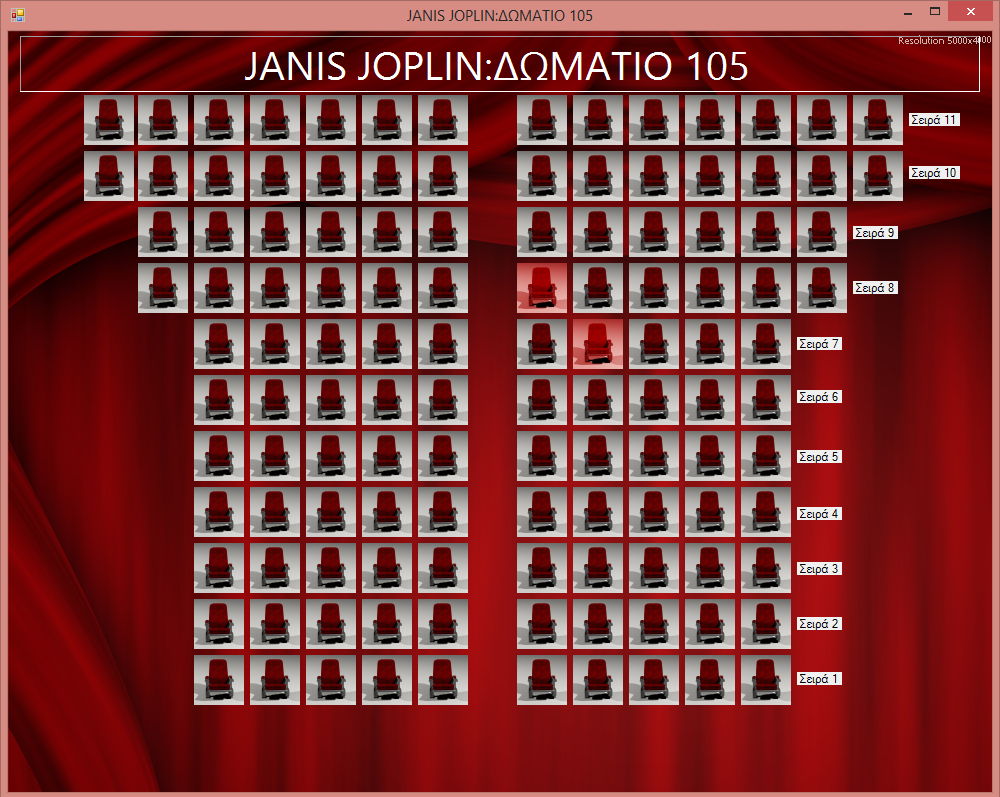
6.1.1  Αρχικό μενού (figure 1)



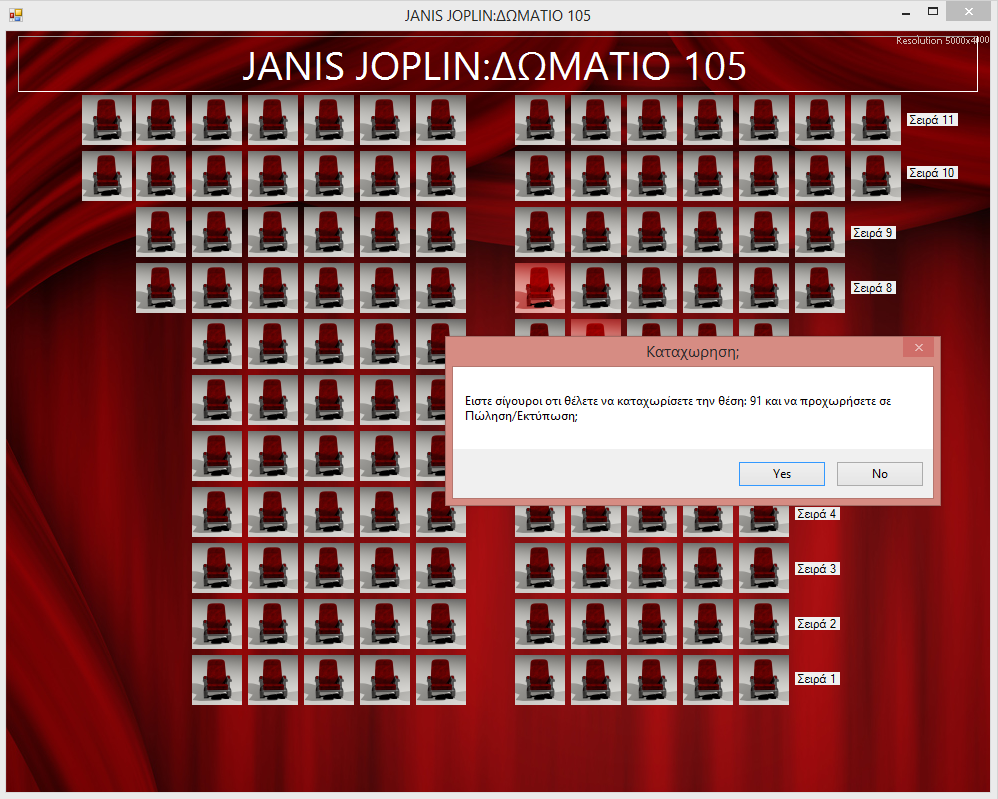
## 6.1.2  Αρχικό μενού-Επιλογή Παράστασης (figure 2)



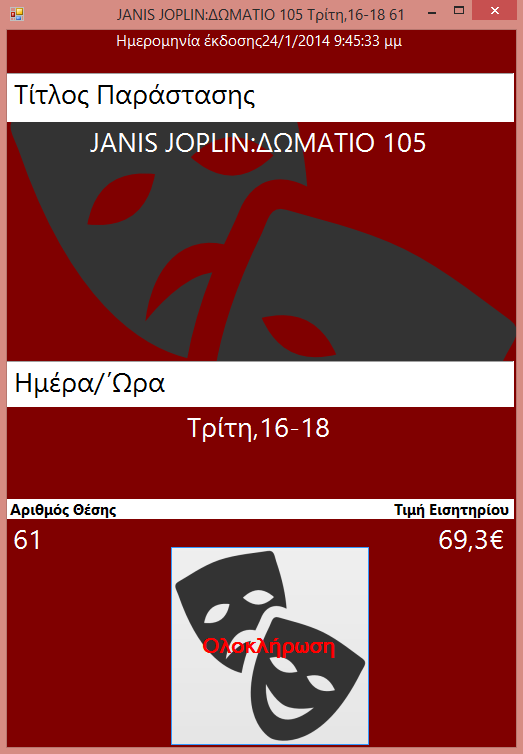
6.2.1  Χάρτης θέσεων προς κράτηση (figure 3)



## 6.2.2  Χάρτης θέσεων προς κράτηση- Διαδικασία κράτησης (figure 4)



6.3.1  Προεπισκόπηση κράτησης εισιτηρίου (figure 5)



## 6.3.2  Εκτύπωση/Κράτηση εισιτηρίου   (figure 6)



**Y.Γ.: Για το server-side κομμάτι της εργασίας εχει στηθεί στον *okeanos* ενα μηχανημα και μας παρεχει webservices για τα δεδομένα.Η εφαρμογή ειναι cloudbased.**