



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ



ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2020-2021

ΑΛΕΞΑΝΔΡΟΣ  
ΠΑΛΗΑΜΠΕΛΟΣ  
E16099

ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΑ ΠΛΑΙΣΙΑ  
ΤΟΥ ΜΑΘΗΜΑΤΟΣ  
ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

# Πίνακας περιεχομένων

<b>Ενότητα 1: Θεματολογία Συστήματος</b> .....	<b>3</b>
1.1 Αναλυτική περιγραφή διαδικασίας υλοποίησης συστήματος .....	3
1.2 Επεξήγηση του προβλήματος που επιλύεται.....	3
<b>Ενότητα 2: Υλοποίηση Συστήματος</b> .....	<b>4</b>
2.1 Η αρχιτεκτονική που χρησιμοποιήθηκε στο σύστημα .....	4
2.2 Περιγραφή των Web Services του συστήματος .....	5
2.3 Περιγραφή των Third Party Web Services του συστήματος.....	13
<b>Ενότητα 3: Εγχειρίδιο Χρήσης Συστήματος</b> .....	<b>14</b>
3.1 Περιγραφή όλων των τεχνολογιών που χρησιμοποιήθηκαν .....	14
3.2 Περιγραφή της Βάσης Δεδομένων που χρησιμοποιήθηκε .....	14
3.3 Οδηγίες εγκατάστασης και εκτέλεσης του συστήματος .....	17
3.4 Επεξήγηση διεπαφών .....	18

# 1. Θεματολογία Συστήματος

## 1.1 Αναλυτική περιγραφή διαδικασίας υλοποίησης συστήματος

Δεδομένου ότι έχουμε γνωστές απαιτήσεις και μικρή μεταβλητότητα κατά την ανάπτυξη, για την υλοποίηση του συστήματος επιλέχθηκε το μοντέλο λειτουργικής επαύξησης. Συνεπώς έχουμε κατάτμηση σε τμήματα ανεξάρτητα και παράλληλα, όπου το κάθε τμήμα ακολουθεί ακολουθιακή ανάπτυξη.

Το πλεονέκτημα μας εδώ

είναι ότι σε πολύ μικρότερο χρόνο ανάπτυξης έχουμε δυνατότητα παράλληλης ανάπτυξης και διαδοχικό εμπλουτισμό λειτουργικών χαρακτηριστικών. Συγκεκριμένα τα παρακάτω τμήματα αναπτύχθηκαν με αυτήν την σειρά και παράλληλα

→T1 : Login – SignUP

→T2 : Μέρος διαχειριστή

    →T2.1 : CHAT διαχειριστή

    →T2.2 : Σύστημα διαχείρισης σινεμά

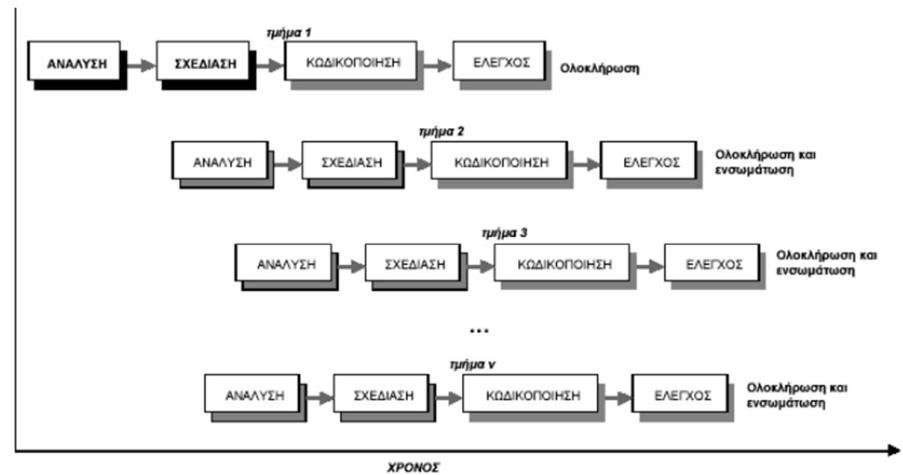
→T3 : Μέρος απλού χρήστη

    →T3.1 : Προφίλ χρήστη

    →T3.2 : CHAT απλού χρήστη

    →T3.3 : Σύστημα κράτηση εισιτήριών

    →T3.4 : Λοιπές δευτερεύουσες σελίδες, index, contactus, aboutus



## 1.2 Επεξήγηση του προβλήματος που επιλύεται

Η συγκεκριμένη εργασία επιλύει ένα βασικό πρόβλημα συστήματος κρατήσεων. Συγκεκριμένα αυτοματοποιείται η ρεαλιστική διαδικασία για ένα σύστημα εκδόσεων εισιτηρίων κινηματογράφου.

Η επιτακτική ανάγκη στις μέρες μας για αγορές/κρατήσεις εισιτηρίων για συναυλίες, κινηματογράφους κ.ο.κ είναι γεγονός και είναι πλέον αδιανότητο μία επιχείρηση που λειτουργεί στα σημερινά δρώμενα να μην έχει έστω ένα απλό σύστημα για αυτοματοποίηση διαδικασιών να εξυπηρετούνται οι πελάτες του. Ο ιστορικός κινηματογράφου του Θησείου αποτελεί ένα μέρος του προβλήματος και δεν προσφέρει τέτοιες υπηρεσίες.

---

## 2. Υλοποίηση Συστήματος

---

### 2.1 Η αρχιτεκτονική που χρησιμοποιήθηκε στο σύστημα

Στο εν λόγω σύστημα χρησιμοποιήθηκε Υπηρεσιοστρεφής Αρχιτεκτονική (Service-Oriented Architecture – SOA). Ο λόγος που χρησιμοποιήθηκε SOA είναι ότι μας δίνεται η δυνατότητα να χωρίσουμε τα επιμέρους components του συστήματος σε διαφορετικές υπηρεσίες που κάθε μια από αυτή εκτελεί μία ξεχωριστή λειτουργία.

Μερικά από τα πλεονεκτήματα της Υπηρεσιοστρεφής Αρχιτεκτονικής:

- Μείωση χρόνου ανάπτυξης επαναχρησιμοποιώντας τις ήδη υλοποιημένες υπηρεσίες μας μέσα στο σύστημα
- Δεδομένου ότι οι υπηρεσίες είναι ανεξάρτητες μεταξύ τους έχουμε εύκολη συντήρηση καθώς μπορούμε να τις τροποποιήσουμε και να τις επεξεργαστούμε χωρίς να επηρεαστεί το σύστημα.
- Αποφυγή μονολιθικού κώδικα-εφαρμογής και κατά συνέπεια εύκολος εντοπισμός σφαλμάτων, άρα και καλύτερη αξιοπιστία
- Μέσω ενός απλού αιτήματος οι υπηρεσίες καθίστανται διαθέσιμες σε οποιονδήποτε
- Εύκολο integration web services από third party προγραμματιστές
- Δεδομένου πως υπάρχει ένα τυπικό communication protocol, μειώνεται δραστικά ο χρόνος αλληλεπίδρασης πελάτη-παρόχου γεγονός που αυξάνει την επεκτασιμότητα

## 2.2 Περιγραφή των Web Services του συστήματος

Για όλα τα web services του συστήματος χρησιμοποιείτε REST αρχιτεκτονική. Συνεπώς όλες οι υπηρεσίες είναι προσβάσιμες μέσω URI. Η κλήση μιας Υπηρεσίας Ιστού χρησιμοποιώντας το HTTP πρωτόκολλο και των GET/PUT/POST/DELETE μεθόδων καθιστά δυνατή την χειρισμό των πόρων του συστήματος.

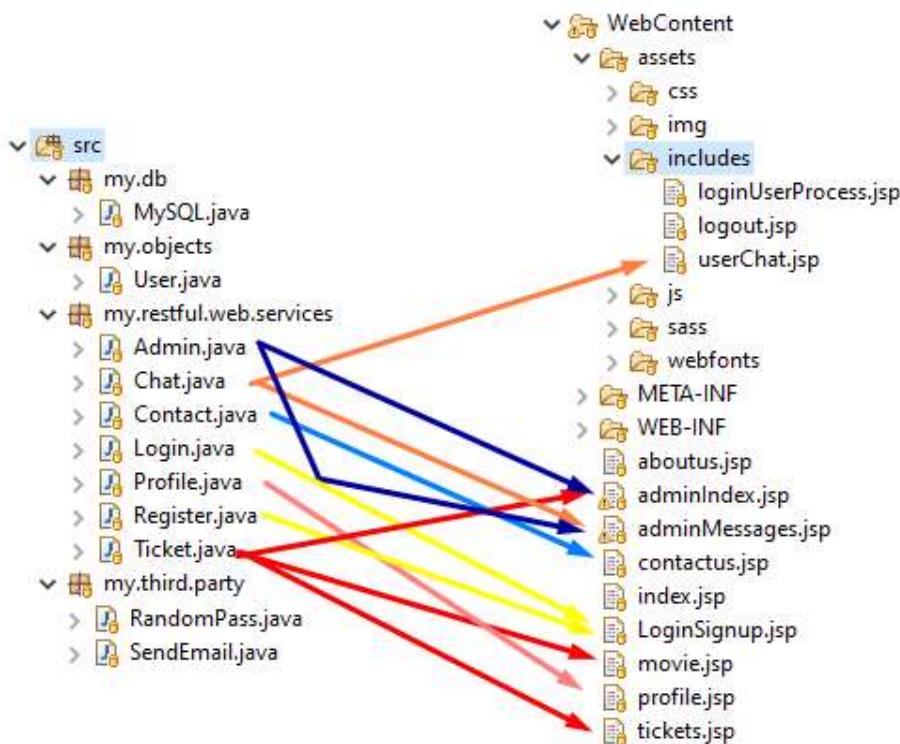
Γιατί REST και όχι SOAP?

- Απλή κλήση URL για να γίνει invoke κάποια υπηρεσία
- Καλύτερη απόδοση – λιγότερη χρησιμοποίηση CPU – λιγότερη ανάγκη bandwidth
- Το SOAP μας δίνει την δυνατότητα μόνο για XML format, σε αντίθεση με το REST που έχουμε plain text, html, xml, json, κλπ.
- Σε περίπτωση που χρειαστούμε SOAP μπορούμε να το συμπεριλάβουμε μέσα στο REST καθώς το REST είναι αρχιτεκτονική ενώ το SOAP πρωτόκολλο, το αντίθετο δεν μπορεί να συμβεί.

Το API που χρησιμοποιήθηκε για την ανάπτυξη **όλων** των Web Services είναι το **JAX-RS**. Συγκεκριμένα χρησιμοποιήθηκε **Jersey\***, το open-source framework για την ανάπτυξη RESTful web services σε Java.

\* Το Jersey 1.x έχει την version JEE6 του JAX-RS

Με μια ματιά τα web-services και οι σελίδες που φαίνονται στον τελικό χρήστη:



Αναλυτικότερα:

## Admin.java

Endpoint: .../rest/AdminService

'Όλα τα WS που υλοποιούνται εδώ χρησιμοποιούνται στην σελίδα adminIndex.jsp

Για αρχή εχουμε 4 GET μεθόδους μια για κάθε “κουτάκι” στο admin dashboard  
Και οι 4 χρησιμοποιούνται στην σελίδα adminIndex.jsp και επαναχρησιμοποιούνται στην adminMessages.jsp



- **public String getUsers()**  
Endpoint : .../rest/AdminService/users  
HTTP Request : GET  
Description : Επιστρέφει τους συνολικούς χρήστες του συστήματος.
- **public String getMovies()**  
Endpoint : .../rest/AdminService/movies  
HTTP Request : GET  
Description : Επιστρέφει τον αριθμό των ταινιών που έχουν εκχωρηθεί στην βάση.
- **public String getTheaters ()**  
Endpoint : .../rest/AdminService/theaters  
HTTP Request : GET  
Description : Επιστρέφει τον αριθμό των αιθουσών του κινηματογράφου που έχουν εκχωρηθεί στην βάση.
- **public String getTodaysMessages(@PathParam("sysadmin") String sysadmin)**  
Endpoint : .../rest/AdminService/messages  
HTTP Request : GET  
Description : Επιστρέφει τον αριθμό των σημερινών μηνυμάτων που έχουν στείλει οι απλοί χρήστες στον συγκεκριμένο διαχειριστή.
- **public String selectMovie()**  
Endpoint: .../AdminService/selectMovie  
HTTP Request : GET  
Description : μας επιστρέφει σε μορφή html select, τις διαθέσιμες ταινίες, Επαναχρησιμοποιείται δύο φορές στην σελίδα του διαχειριστή. Στο section για επεξεργασία ταινία και στο section για προσθήκη προγράμματος.
- **public String selectTheater ()**  
Endpoint: .../AdminService/selectTheater  
HTTP Request : GET  
Description : μας επιστρέφει σε μορφή html select, τις διαθέσιμες αιθουσες (id)
- **public String selectSchedule()**  
Endpoint: .../AdminService/selectSchedule  
HTTP Request : GET  
Description : μας επιστρέφει σε μορφή html select, τα διαθέσιμα προγράμματα (id)

- ```
public String getTheaterSpots(@PathParam("tID") String tID)
```

Endpoint: .../AdminService/theater/{tID}  
HTTP Request : GET  
Description : Μας επιστρέφει τις συνολικές θέσεις που έχει η αίθουσα κινηματογράφου για κάποιο id. Το χρησιμοποιούμε στο section "edit theater"
- ```
public String addMovie(@PathParam("mtitle") String mtitle, @PathParam("myear") String myear, @PathParam("mstars") String mstars, @PathParam("mdesc") String mdesc, @PathParam("mmin") String mmin, @PathParam("mage") String mage)
```

Endpoint:  
.../AdminService/movie/{mtitle}/{myear}/{mstars}/{mdesc}/{mmin}/{mage}  
HTTP Request : POST  
Description : To webservice αυτό κάνει INSERT στον πίνακα movie την αντίστοιχη ταινία που ο admin πληκτρολογεί στην φόρμα.
- ```
public String addTheater(@PathParam("tID") String tID, @PathParam("tSpots") String tSpots)
```

Endpoint: .../AdminService/theater/{tID}/{tSpots}  
HTTP Request : POST  
Description : To webservice αυτό κάνει INSERT στον πίνακα theater, τη νέα αίθουσα του κινηματογράφου με την αντίστοιχη χωρητικότητα που έχει.
- ```
public String addSchedule(@PathParam("tID") String tID, @PathParam("mID") String mID, @PathParam("dateOfAir") String dateOfAir, @PathParam("movieUP") String movieUP)
```

Endpoint: .../AdminService/schedule/{tID}/{mID}/{dateOfAir}/{movieUP}  
HTTP Request : POST  
Description : To webservice αυτό κάνει INSERT στον πίνακα schedule την ταινία που θα παιχτεί σε κάποια μέρα σε κάποια αίθουσα.
- ```
public String editMovie(@PathParam("mmID") String mmID, @PathParam("mtitle") String mtitle, @PathParam("myear") String myear, @PathParam("mstars") String mstars, @PathParam("mdesc") String mdesc, @PathParam("mmin") String mmin, @PathParam("mage") String mage)
```

Endpoint: .../AdminService  
/movie/{mmID}/{mtitle}/{myear}/{mstars}/{mdesc}/{mmin}/{mage}  
HTTP Request : PUT  
Description : To webservice αυτό κάνει UPDATE στον πίνακα movie την αντίστοιχη ταινία που ο admin πληκτρολογεί στην φόρμα.
- ```
public String editSchedule(@PathParam("sID") String sID, @PathParam("date") String date, @PathParam("mUP") String mUP)
```

Endpoint: .../AdminService/schedule/{sID}/{date}/{mUP}  
HTTP Request : PUT  
Description : To webservice αυτό κάνει UPDATE στον πίνακα schedule, μόνο την ημερομηνία προβολής και το αν η ταινία είναι διαθέσιμη για το κοινό
- ```
public String getScheduleInfo(@PathParam("sID") String sID, @PathParam("type") String type)
```

Endpoint: .../AdminService/schedule/{sID}/{type}  
HTTP Request : GET  
Description : To webservice αυτό το χρησιμοποιούμε στο section "edit schedule" προκειμένου να συμπληρωθεί η φόρμα με τα δύο πεδία.

## Chat.java

Endpoint: .../rest/ChatService

- `public String pushMessage(@PathParam("id") int id, @PathParam("type") String type, @PathParam("user") String user, @PathParam("sysadmin") String sysadmin, @PathParam("msg") String msg)`  
Endpoint :  
.../rest/ChatService/pushMessage/{id}/{type}/{user}/{sysadmin}/{msg}  
HTTP Request : POST  
Description : Τοποθετεί το μήνυμα που θα στείλει ο χρήστης μέσω της αντίστοιχης διεπαφής στον πίνακα chatrooms.  
msg = το μήνυμα που στέλνει ο χρήστης  
user = το username του απλού χρήστη  
sysadmin = το username του διαχειριστή  
type = "U" for user || "A" for admin → ποιός από τους δύο στέλνει το μήνυμα  
id = το id του chatroom, κάθε ζευγάρι χρήστη-διαχειριστή μπορούν να έχουν μόνο ένα chatroom για να βρούμε το κατάλληλο id χρησιμοποιούνται οι βοηθητικές μέθοδοι getChatNo() & openChat(String user, String sysadmin).  
Το συγκεκριμένο webservice επαναχρησιμοποιείται σε όλες τις σελίδες που ο χρήστης έχει κάνει login.
- `public String openChat(@PathParam("user") String user, @PathParam("sysadmin") String sysadmin)`  
Endpoint : .../rest/ChatService/openChat/{user}/{sysadmin}  
HTTP Request : GET  
Description : Ελέγχει αν υπάρχει ήδη chatroom μεταξύ user-admin. Αν υπάρχει επιστρέφει το id του chatroom σε διαφορετική περίπτωση επιστρέφει "0".  
Το συγκεκριμένο webservice επαναχρησιμοποιείται σε όλες τις σελίδες που ο χρήστης έχει κάνει login.
- `private int getChatNo()`  
Description : Επιστρέφει τον αριθμό των chatrooms. Αν η openChat() μας επιστρέψει "0", η pushMessage() θα την καλέσει για να βρει το επόμενο διαθέσιμο id για την δημιουργία του chatroom
- `public String getAdmins()`  
Endpoint : .../rest/ChatService/admins  
HTTP Request : GET  
Description : Επιστρέφει σε μορφή html <select> όλους τους διαχειριστές προκειμένου ο απλός χρήστης να επιλέξει έναν για να συνομιλήσει  
Το συγκεκριμένο webservice επαναχρησιμοποιείται σε όλες τις σελίδες που βλέπει ο απλός χρήστης.
- `public String getAdmins2()`  
Description : Ιδια λογική απλώς αλλάζει το id του select tag. Αυτό γίνεται γιατί υπήρχε κάποιο πρόβλημα σχετικά με την επαναχρησιμοποιήσει του ίδιου select στην ίδια form. Προτιμήθηκε αυτός ο "κακός"τρόπος για να μην αλλάξει τελείως ο σχεδιασμός του chat.
- `public String getMsgs (@PathParam("id") int chatid)`  
Endpoint : .../rest/ChatService/chat/{id}

### HTTP Request : GET

Description : Επιστρέφει τα μηνύματα για το αντίστοιχο chatroom. Κάθε μήνυμα που βρίσκεται στην βάση τοποθετείται σε διαφορετικό <div> αναλόγως αν έχει σταλθεί από user ή από admin (type == "U" || type == "A").  
Το συγκεκριμένο webservice επαναχρησιμοποιείται σε όλες τις σελίδες που ο χρήστης έχει κάνει login.

- `public String getAdminMsgList(@PathParam("sysadmin") String sysadmin)`

Endpoint : .../adminMsgList/{sysadmin}

### HTTP Request : GET

Description : Επιστρέφει στη πλευρά του διαχειριστή μία html form προκειμένου να επιλέξει τον χρήστη που επιθυμεί να του απαντήσει. Μπορεί να επιλέξει μόνο χρήστες που του έχουν στείλει πρώτα αυτοί μήνυμα και όχι να δημιουργήσει νέο chat με κάποιον χρήστη όπως συμβαίνει στην πλευρά του απλού χρήστη.  
Αρχικά το web-service ψάχνει ποιοι χρήστες έχουν στείλει στον admin και τους τοποθετεί σε μία λίστα, έπειτα από αυτή την λίστα κάθε χρήστης μπαίνει σε option html tag.

Το συγκεκριμένο webservice επαναχρησιμοποιείται στις σελίδες που έχει πρόσβαση μόνο ο διαχωριστής (adminIndex.jsp & adminMessages.jsp)

## Contact.java

Endpoint: .../rest/ContactService

- `public String contactUs(@PathParam("cName") String cName, @PathParam("cEmail") String cEmail, @PathParam("cMsg") String cMsg)`

Endpoint : .../rest/ContactService/contactUs/{cName}/{cEmail}/{cMsg}

### HTTP Request : POST

Description : Από την form που βρίσκεται στην σελίδα contactus.jsp ο χρήστης πληκτρολογεί τα δικά του στοιχεία:

cName = όνομα χρήστη

cEmail = email χρήστη

cMsg = το μήνυμα που θέλει να στείλει

Τα στοιχεία αυτά στέλνονται στο [cinethisio7@gmail.com](mailto:cinethisio7@gmail.com), ένα email που φτιάχτηκε για την εργασία αυτή.

Στη συνέχεια το webservice φτιάχνει ένα auto reply μήνυμα και του στέλνει στο email που πληκτρολογήθηκε από τον χρήστη.

## Login.java

Endpoint: .../rest/LoginService

- `public Response userAuth(@PathParam("user") String user, @PathParam("pass") String pass)`

Endpoint : .../rest/LoginService/userAuth/{user}/{pass}

### HTTP Request : GET

Description : To webservice αυτό χρησιμοποιείται μόνο στην σελίδα

LoginSignup.jsp και υλοποιεί την διαδικασία της αυθεντικοποίησης του χρήστη για την είσοδο του στο σύστημα. Μας επιστρέφει ένα Response με το αντίστοιχο status code και το string της μορφής `"ADMIN:TRUE" || "ADMIN:FALSE" || "Wrong username or password!"`

Συνεπώς,

Αν η αυθεντικοποίηση ήταν επιτυχής status code = 200 και ADMIN:TRUE ή ADMIN:FALSE  
Αλλιώς status code = 400 και wrong username or password!

- `private boolean isSysAdmin(String user)`

Για τον καθορισμό του string ADMIN:TRUE ή ADMIN:FALSE χρησιμοποιείται η συγκεκριμένη private Boolean method.

## Register.java

Endpoint: .../rest/RegisterService

- `public String userAuth(@PathParam("fname") String fname, @PathParam("lname") String lname, @PathParam("email") String email, @PathParam("usr") String usr, @PathParam("pwd") String pwd)`

Endpoint : .../ RegisterService/user/{fname}/{lname}/{email}/{usr}/{pwd}

HTTP Request : POST

Description : To webservice αυτό χρησιμοποιείται μόνο στην σελίδα LoginSignup.jsp και υλοποιεί την διαδικασία της εγγραφής του χρήστη.

Κάνει ένα INSERT στον πίνακα users χρησιμοποιώντας όλα τα στοιχεία που ζητούνται από την αντίστοιχη φόρμα.

Για το τυχαίο 6ψήφιο password χρησιμοποιείται 3<sup>rd</sup> party webservice το οποίο βρίσκεται παρακάτω στην αντίστοιχη ενότητα.

## Profile.java

Endpoint: .../rest/ProfileService

- `public List<User> userInfo(@PathParam("user") String user)`

Endpoint : .../ ProfileService/userInfo/{user}

HTTP Request : GET

Description : To webservice αυτό χρησιμοποιείται μόνο στην σελίδα profile.jsp και ο ρόλος του είναι να φορτώσουμε την φόρμα που εμφανίζεται στον χρήστη με τα δεδομένα του. Συνεπώς επιστρέφεται μία λίστα του αντικειμένου User. Η κλάση User βρίσκεται στο πακέτο my.objects και έχει τους κατάλληλους geters και seters για τον ορισμό του αντικειμένου από την userInfo(). Στην αρχή είχε προτιμηθεί να επιστρέψει ένα xml <user id=""><name>...</name>....</user> αλλά αλλάχτηκε. Ουσιαστικά με τον υλοποιημένο τρόπο επιστρέφουμε και πάλι xml element, περισσότερες πληροφορίες παρακάτω στο User.java

- `public String userInfo(@PathParam("user") String user, @PathParam("pass") String pass, @PathParam("fname") String fname, @PathParam("lname") String lname, @PathParam("email") String email)`

Endpoint : .../ ProfileService/userInfo/{user} /{pass}/{fname}/{lname}/{email}

HTTP Request : PUT

Description : To webservice αυτό χρησιμοποιείται μόνο στην σελίδα profile.jsp και ο ρόλος του είναι να κάνει update τα στοιχεία του user στην βάση από την φόρμα που έχει συμπληρωθεί από την προηγούμενη συνάρτηση και έχει ενημερωθεί από τον χρήστη.

- `public String logs(@PathParam("user") String user)`

Endpoint : .../ ProfileService/logs/{user}

HTTP Request : POST

Description : To webservice αυτό χρησιμοποιείται μόνο στην σελίδα LoginSignup.jsp και ο ρόλος του είναι να κάνει insert στην DB, το όνομα του χρήστη και την ώρα που έγινε το login. Η συνάρτηση γίνεται invoke έπειτα από επιτυχημένο login.

- `public String logs(@PathParam("user") String user, @PathParam("type") String type)`  
Endpoint : .../ProfileService/logs/{user}/{type}  
HTTP Request : GET  
Description : To webservice αυτό χρησιμοποιείται μόνο στην σελίδα profile.jsp και μας επιστρέφει τα logs του χρήστη για ένα συγκεκριμένο type.  
Αν type.equals("\*") τότε μας επιστρέφει όλα τα log στον αντίστοιχο πίνακα.  
Σε διαφορετική περίπτωση υποθέτουμε ότι στο URI έχουμε κάποιον int x, και επιστρέφουμε τα x τελευταία logs.  
Πάντα γυρνάμε τα logs από το νεότερο προς το παλαιότερο (DESC).  
Στο profile.jsp έχω βάλει να μας γυρνάνε τα 15 τελευταία logs.

## Ticket.java

Endpoint: .../rest/TicketService

- `public String getSchedule(@PathParam("dateOfAir") String dateOfAir, @PathParam("type") String type)`  
Endpoint: .../TicketService/schedule/{dateOfAir}/{type}  
HTTP Request : GET  
Description : To webservice αυτό χρησιμοποιείται στις σελίδες που έχουμε πρόσβαση στο πρόγραμμα του κινηματογράφου. Επαναχρησιμοποιείται και στην πλευρά του διαχειριστή και στου χρήστη. Μας επιστρέφει (html table) το πρόγραμμα του κινηματογράφου για κάποια επιλεγμένη μέρα. Αν το χρησιμοποιούμε στην πλευρά του χρήστη τότε (type=="U") έτσι ώστε να μας δείξει μόνο τις ταινίες που είναι διαθέσιμες για το κοινό (movieUP==1)
- `public String book(@PathParam("uID") String uID, @PathParam("fname") String fname, @PathParam("lname") String lname, @PathParam("email") String email, @PathParam("spots") String spots, @PathParam("sID") String sID)`  
Endpoint: .../TicketService/book/{uID}/{fname}/{lname}/{email}/{spots}/{sID}  
HTTP Request : POST  
Description : To webservice αυτό χρησιμοποιείται μόνο στην μεριά του απλού χρήστη και συγκεκριμένα στην σελίδα tickets.jsp όπου ο χρήστης καλείται να κάνει κράτηση εισιτηρίων. Αν η κράτηση εισιτηρίων είναι επιτυχής τότε γίνεται insert στον πίνακα booking και μετά γίνεται update στον πίνακα schedule (για τις νέες διαθέσιμες θέσεις). Στον χρήστη έρχεται email με τα εισητήρια που έκλεισε. Σε διαφορετική περίπτωση (ανεπιτυχούς κράτησης) εμφανίζονται αντίστοιχα μηνύματα.
- `public String getMovieFullInfo(@PathParam("mID") String mID, @PathParam("type") String type)`  
Endpoint: .../TicketService/movie/{mID}/{type}  
HTTP Request : GET  
Description : To webservice αυτό χρησιμοποιείται μόνο στην μεριά του απλού χρήστη και συγκεκριμένα στην σελίδα movie.jsp.

Σε αυτήν την σελίδα ο χρήστης δεν έχει έμμεση πρόσβαση για να ανακατευθυνθεί σε αυτή πρέπει να πατήσει πάνω στο όνομα της ταινία από το πρόγραμμα (tickets.jsp). Σε αυτή την σελίδα δίνουμε το id της ταινίας μέσω του URI (movie.jsp?movieID=5) και ουσιαστικά είναι μία δυναμική σελίδα που παρουσιάζουμε όλα τα στοιχεία της ταινίας που έχουμε στην βάση. Επαναχρησιμοποιείται στην ίδια σελίδα το ίδιο webservice 6 φόρες, για κάθε type μας επιστρέφει διαφορετικό στοιχείο της ταινίας.

- `private String getMovieName(int mID)`

Description : Είναι μία βοηθητική private συνάρτηση που για κάποιο id ταινίας μας επιστρέφει τον τίτλο της. Χρησιμοποιείται σε δύο webservices getSchedule(), book() για να την βελτιστοποίηση της αναγνωσιμότητας του κώδικα.

## User.java

Δεν υλοποιείται κάποιο webservice (my.object package)

- `public class User`

Όπως συζητήθηκε και προηγουμένως σε αυτή την κλάση υλοποιείται το object που αρχικοποιείται από το web-service στο userInfo (Profile.java). Χρησιμοποιείται το `@XmlElement(name="User")` για να μαραρουμε την κλάση σε ένα xml element. Η κλάση αυτή έχει τους seters και geters για τον ορισμό του αντικειμένου User() για τα παιδία που μπορούμε να βρούμε στον πίνακα users της DB.

## MySQL.java

Δεν υλοποιείται κάποιο webservice (my.db package) αλλά ουσιαστικά χρησιμοποιείται από όλα τα webservices για την επικοινωνία με την DB.

- `public class MySQL`

Αρχικοποιούνται όλες οι απαραίτητες μεταβλητές JDBC Driver, username, pass, etc. Στη συνέχεια έχουμε 3 μεθόδους `getProperties()`, `connectDB()`, `disconnectDB()` όπου και πραγματοποιούνται οι αντίστοιχες λειτουργίες.

## 2.3 Περιγραφή των Third Party Web Services του συστήματος

### RandomPass.java

Endpoint : ../rest/RandomPass  
(my.third.party package)

- `public String getRandomString()`  
Endpoint: .../RandomPass/pass  
HTTP Request : GET  
Description : To webservice αυτό χρησιμοποιείται κατά την διαδικασία της εγγραφής του χρήστη στο σύστημα. Επιστρέφει έναν τυχαίο 6-ψηφιο string που μπορεί να χρησιμοποιηθεί σαν password.  
Πηγή: <https://stackoverflow.com/questions/20536566/creating-a-random-string-with-a-z-and-0-9-in-java>

### SendEmail.java

Τελικά προτιμήθηκε να μην χρησιμοποιηθεί σαν web service έτσι ώστε να αποφευχθεί να μεταφέρω τεράστια string στο URI και να πρέπει να κάνω escape όλους τους χαρακτήρες που χρειάζονται  
(my.third.party package)

- `public boolean sendEmail(String recipient, String subject, String body )`  
Description : Επιστρέφει μία Boolean τιμή true/false ανάλογα αν το μήνυμα έχει σταλθεί ή όχι. Χρησιμοποιείται ο smtp relay του mail.com δηλ. του email που φτιάχτηκε για τους σκοπούς αυτής της εργασίας ([cinethisio7@gmail.com](mailto:cinethisio7@gmail.com)).  
Αυτή η μέθοδος επαναχρησιμοποιείται στις εξής περιπτώσεις:
  - ➔ send tickets after purchase
  - ➔ send mail to [cinethisio7@gmail.com](mailto:cinethisio7@gmail.com) when the contactUs form submits
  - ➔ send auto reply mail when someone submit the contactUs formΧρησιμοποιώ τις 3 παραμέτρους recipient, subject, body για να την καλώ από οπουδήποτε.  
Πηγή: <https://www.javatpoint.com/example-of-sending-email-using-java-mail-api>

### Σαν σύνολο υλοποιούνται:

- my.db
  - MySQL.java | 1 private method, 2 public methods
- my.objects
  - User.java | 14 public methods (get + set)
- my.restful.web.services = **31 WEB SERVICES**
  - Admin.java | 15 Web-Services
  - Chat.java | 6 Web-Services, 1 private method
  - Contact.java | 1 Web-Service
  - Login.java | 1 Web-Service, 1 private method
  - Profile.java | 4 Web-Services
  - Register.java | 1 Web-Services
  - Ticket.java | 3 Web-Services, 1 private method
- my.third.party
  - RandomPass.java | 1 Web-Service
  - SendEmail.java | 1 public method

### 3. Εγχειρίδιο Χρήσης Συστήματος

#### 3.1 Περιγραφή όλων των τεχνολογιών που χρησιμοποιήθηκαν

Χρησιμοποιήθηκαν οι γλώσσες JSP, HTML, JS, CSS, Bootstrap, και JAVA 8 για την υλοποίηση του συστήματος στο Eclipse (v.2018-09) και Postman για debugging.

Το dynamic project του eclipse τρέχει τοπικά σε έναν Apache Tomcat (v9).

Η βάση δεδομένων είναι MySQL εγκαταστημένη και αυτή τοπικά.

#### 3.2 Περιγραφή της Βάσης Δεδομένων που χρησιμοποιήθηκε

Στην MySQL έχει δημιουργηθεί η database "cinethisio" η οποία είναι η μόνη που αλληλοεπιδράει με το σύστημα. Στα δεξιά φαίνονται και οι πίνακες που έχουν δημιουργηθεί για την υλοποίηση των endpoints του συστήματος. Στη συνέχεια θα αποτυπωθεί μία σύντομη περιγραφή για κάθε έναν πίνακα.

```
mysql> SELECT DATABASE();
+-----+
| DATABASE() |
+-----+
```

```
mysql> SHOW TABLES;
+-----+
| Tables_in_cinethisio |
+-----+
| booking
| chatrooms
| logs
| movie
| schedule
| theater
| users
+-----+
```

**DESC users;** Εδώ αποθηκεύονται όλοι οι χρήστες του συστήματος

uid = μοναδικό ID user

password = MD5(pass)

sysadmin = boolean

```
+-----+
| Field      | Type       | Null | Key | Default | Extra           |
+-----+
uid	int unsigned	NO	PRI	NULL	auto_increment
username	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
firstname	varchar(255)	YES		NULL	
lastname	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
sysadmin	tinyint(1)	YES		NULL	
+-----+
```

```
mysql> select * from users;
+-----+
| uid | username | password          | firstname | lastname | email           | sysadmin |
+-----+
1	alex	b75bd008d5fecb1f50cf026532e8ae67	Alexandros	Paliampelos	alexpap18@gmail.com	1
2	nick	4156b5c0fdc96160b1da20d1d54175b9	Nikolas	Papadopoulos	nick@gmail.com	0
3	xristros	fef96d637ccc8977e9c9ff894cf43e1d	Xristos	Zaxos	chris@gmail.com	1
4	mixalis	854904d67069867fc672105e3415572f	Mike	Papazoglou	alex-otypas@hotmail.com	0
+-----+
```

**DESC logs;** Εδώ αποθηκεύονται τα logs έπειτα από κάθε επιτυχημένο login

```
+-----+
| Field      | Type       | Null | Key | Default | +-----+
| username   | varchar(255) | YES  |      | NULL    | | username | date
| date       | datetime   | YES  |      | NULL    | +-----+
+-----+   | alex      | 2021-08-18 23:05:57 |
alex	2021-08-18 23:39:18
alex	2021-08-19 00:12:06
alex	2021-08-19 00:16:26
```

**DESC chatrooms;** Εδώ αποθηκεύονται τα ασύγχρονα μηνύματα που ανταλλάσσουν χρήστης και διαχειριστής.

chatID = το chat που θέλουμε να ανοίγουμε κάθε φορά και να τοποθετούμε μηνύματα  
 type = συμβολίζει ποιος έχει στείλει το μήνυμα U=user, A=admin  
 user = username απλού χρήστη  
 sysadmin = username admin  
 msg = το μήνυμα που στάλθηκε  
 date = ημερομηνία αποστολής

| Field    | Type         | Null | Key | Default |
|----------|--------------|------|-----|---------|
| chatID   | int unsigned | YES  |     | NULL    |
| type     | char(1)      | YES  |     | NULL    |
| user     | varchar(255) | YES  |     | NULL    |
| sysadmin | varchar(255) | YES  |     | NULL    |
| message  | varchar(255) | YES  |     | NULL    |
| date     | datetime     | YES  |     | NULL    |

mysql> select \* from chatrooms;

| chatID | type | user    | sysadmin | message                   | date                |
|--------|------|---------|----------|---------------------------|---------------------|
| 1      | U    | nick    | xristos  | kalispera sas             | 2021-08-22 01:47:24 |
| 1      | U    | nick    | xristos  | exw mia aporia            | 2021-08-22 02:26:55 |
| 3      | U    | nick    | alex     | hello alex!               | 2021-08-22 02:32:09 |
| 3      | A    | nick    | alex     | pes mou nick              | 2021-08-22 02:32:41 |
| 3      | U    | nick    | alex     | ola kala                  | 2021-08-23 13:06:18 |
| 3      | A    | nick    | alex     | kali sinexeia             | 2021-08-23 13:06:42 |
| 7      | U    | mixalis | alex     | Hello mr Alexandros       | 2021-08-25 20:53:39 |
| 7      | U    | mixalis | alex     | eiste edw                 | 2021-08-25 20:55:49 |
| 7      | A    | mixalis | alex     | Pos mporo na sas voithiso | 2021-08-25 20:58:47 |

**DESC theater;** Ο πίνακας αυτός αντιπροσωπεύει τις διαθέσιμες αίθουσες προβολής ταινιών που έχει ο κινηματογράφος

| Field      | Type         | Null | Key | Default | Extra          |
|------------|--------------|------|-----|---------|----------------|
| tID        | int unsigned | NO   | PRI | NULL    | auto_increment |
| totalSpots | int          | YES  |     | NULL    |                |

Ο κινηματογράφος προς το παρών έχει μία αίθουσα με 200 θέσεις σαν χωρητικότητα

| tID | totalSpots |
|-----|------------|
| 1   | 200        |

**DESC movie;** Ο πίνακας αυτός αντιπροσωπεύει τις διαθέσιμες ταινίες που προβάλλονται στις εκάστοτε αίθουσες

| Field       | Type         | Null | Key | Default | Extra          |
|-------------|--------------|------|-----|---------|----------------|
| mID         | int unsigned | NO   | PRI | NULL    | auto_increment |
| title       | varchar(255) | YES  |     | NULL    |                |
| year        | int          | YES  |     | NULL    |                |
| stars       | varchar(255) | YES  |     | NULL    |                |
| description | varchar(255) | YES  |     | NULL    |                |
| duration    | int          | YES  |     | NULL    |                |
| age         | int          | YES  |     | NULL    |                |

| mID | title      | year | stars                       | description                                                                                               |
|-----|------------|------|-----------------------------|-----------------------------------------------------------------------------------------------------------|
| 1   | La La Land | 2016 | Ryan Gosling and Emma Stone | La La Land is a 2016 American musical romantic comedy-drama film written and directed by Damien Chazelle. |
| 2   | Il Postino | 1994 | Massim Troisi               | A simple Italian postman learns to love poetry while delivering mail to a famous poet.                    |

| duration | age |
|----------|-----|
| 130      | 13  |
| 110      | 13  |

### DESC schedule;

Σε αυτόν τον πίνακα αποθηκεύομε το πρόγραμμα του κινηματογράφου για κάποια ημέρα sID = μοναδικό id προγράμματος

| Field          | Type         | Null | Key | Default | Extra          |
|----------------|--------------|------|-----|---------|----------------|
| sID            | int unsigned | NO   | PRI | NULL    | auto_increment |
| theaterID      | int          | YES  |     | NULL    |                |
| movieID        | int          | YES  |     | NULL    |                |
| dateOfAir      | datetime     | YES  |     | NULL    |                |
| availableSpots | int          | YES  |     | NULL    |                |
| totalSpots     | int          | YES  |     | NULL    |                |
| movieUP        | tinyint(1)   | YES  |     | NULL    |                |

theaterID = αιθουσα προβολής

movieID = η ταινία που προβάλλεται

dateOfAir = η ημερομηνία που προβάλλεται

availableSpots = οι διαθέσιμες θέσεις

totalSpots = οι συνολικές θέσεις (της αιθουσας)

movieUP = μεταβλητή ελέγχου Boolean π.χ όταν θα θέλει ο διαχειριστής να αποκρύψει το πρόγραμμα αυτό (ταινία προσεχώς, ή ταινία που σταματάει να την προβάλει ο κινηματογράφος).

```
mysql> select * from schedule;
```

| sID | theaterID | movieID | dateOfAir           | availableSpots | totalSpots | movieUP |
|-----|-----------|---------|---------------------|----------------|------------|---------|
| 1   | 1         | 1       | 2021-08-24 21:45:00 | 198            | 200        | 1       |
| 2   | 1         | 2       | 2021-08-24 23:45:00 | 200            | 200        | 1       |

### DESC booking; Σε αυτόν

τον πίνακα έχουμε τα εισιτήρια που έχουν κάνει κράτηση οι χρήστες.

bID = αριθμός booking

uID = ID user

| Field | Type         | Null | Key | Default | Extra          |
|-------|--------------|------|-----|---------|----------------|
| bID   | int unsigned | NO   | PRI | NULL    | auto_increment |
| uID   | int          | YES  |     | NULL    |                |
| sID   | int          | YES  |     | NULL    |                |
| spots | int          | YES  |     | NULL    |                |

sID = ID προγράμματος (schedule)

spots = αριθμός κρατημένων θέσεων

```
mysql> select * from booking;
```

| bID | uID | sID | spots |
|-----|-----|-----|-------|
| 1   | 2   | 1   | 2     |

### 3.3 Οδηγίες εγκατάστασης και εκτέλεσης του συστήματος

Τι θα χρειαστούμε:

- Να έχουμε κάνει εγκατάσταση MySQL 8 και τα credentials για τον διαχειριστή να είναι user="root" και pass="root". Σε αντίθετη περίπτωση θα πρέπει να αλλαχτούν στο MySQL.java

Θα χρειαστεί να δημιουργήσουμε μία βάση με όνομα "cinethisio" και έπειτα να κάνουμε import το sql αρχείο με την εξής εντολή:

```
mysql -u root -p cinethisio < PATH/cinethisio.sql
```

όπου PATH είναι το σημείο που βρίσκεται το αρχείο, θα το βρείτε στον φάκελο my\_db αυτής της εργασίας.

- Να έχουμε java 8 έκδοση εγκαταστημένη. Εδώ θα βρούμε το [JDK 8](#)
- Να έχουμε κάνει εγκατάσταση της [2018-09 JAVA EE](#) έκδοσης του Eclipse
- Να έχουμε βάλει Apache tomcat v9 στο eclipse και στον φάκελο που έχουμε κάνει εγκατάσταση τον tomcat να αντιγράψουμε το αρχείο my\_jar\external\mysql-connector-java-5.1.49 στον φάκελο lib της εγκατάστασης του tomcat
- Να αντιγράψουμε το project ολόκληρο στο worckspace του eclipse. Το project βρίσκεται στον φάκελο my\_files\CINEThisio. Και έπειτα να κάνουμε import το project
- Να ελέγξουμε το build path του project.

Θα πρέπει τα παρακάτω jar να υπάρχουν στο build path :

javax.mail , jersey-bundle-1.19.4, jsr311-api-1.1.1, mysql-connector-java-5.1.49

Σημ: όλα τα external jar υπάρχουν και στον φάκελο my\_jar\external\ αλλά και στον φάκελο WebContent\WEB-INF\lib του project!

Σε περίπτωση που μας βγάζει error τότε διαγράφουμε από το build path όλα τα external jars και τα ξανά προσθέτουμε.

Για να τρέξουμε την εργασία απλά τρέχουμε το project από το eclipse στον apache server μας και έπειτα από τον browser μας πάμε στην σελίδα

<http://localhost:8080/CINEThisio> όπου θα μας μεταφέρει στην κεντρική index.jsp

Παρακάτω θα βρούμε τους συνδυασμούς user-pass για όλους τους χρήστες του συστήματος:

|                |  |                   |
|----------------|--|-------------------|
| user: alex     |  | pass: alex123     |
| user: xristsos |  | pass: xristsos123 |
| user: nick     |  | pass: 0b4y81      |
| user: mixalis  |  | pass: uxc04p      |

Να σημειωθεί πως οι 2 πρώτοι (alex, xristsos) έχουν δικαιώματα διαχειριστή στο σύστημα.

### 3.4 Επεξήγηση διεπαφών

#### Register.java – RandomPass.java

Ανοίγοντας την σελίδα loginSignup.jsp (μέσω του μενού – πάνω δεξιά) ο χρήστης μπορεί είτε να κάνει signup είτε login.

Παρακάτω φαίνεται η υλοποίηση των δύο διεπαφών της εγγραφής και της δημιουργίας αυτοματοποιημένου password

The screenshot shows a Java web application interface. On the left, there is a sign-up form with fields for First Name (Miltiadis), Last Name (Papazoglou), Email (alex-otypas@hotmail.com), Username (mixalis), and Password. A red arrow points from the 'Password' field to a message at the bottom right of the page: "το password έχει ήδη δημιουργηθεί και θα μας εμφανιστεί στο επόμενο βήμα". Below the form are two buttons: LOGIN and SIGN UP (which is highlighted in blue).

On the right, a modal window titled "localhost:8080 says" displays the generated password: "To password σας είναι: uxc04p" followed by "Sign Up SUCCESSFUL". An OK button is visible in the bottom right corner of the modal.

## Login.java

Ανοίγοντας την σελίδα loginSignup.jsp (μέσω του μενού – πάνω δεξιά) ο χρήστης μπορεί είτε να κάνει signup είτε login.

Παρακάτω φαίνεται η υλοποίηση της διεπαφής για την αυθεντικοποίηση για την είσοδο του χρήστη στο σύστημα.

Αν κάνει είσοδο στο σύστημα ένας διαχειριστής τότε μεταφέρεται στο admin dashboard

**Login**

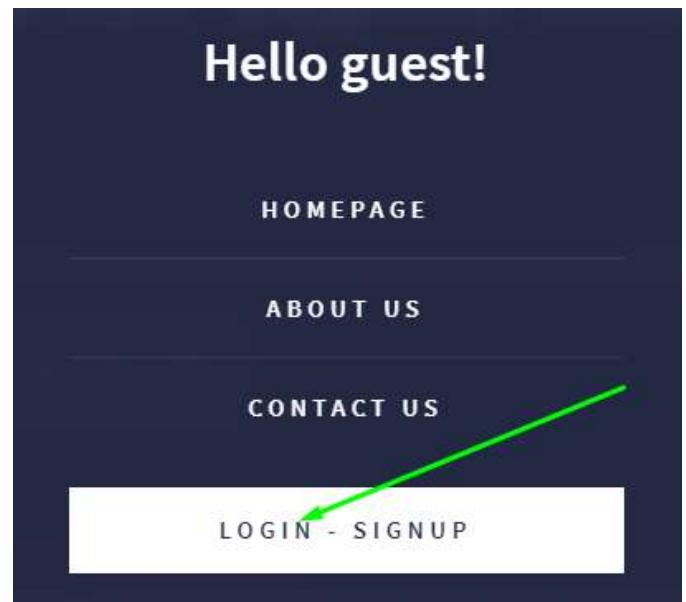
Για να αποκτήσετε πρόσβαση στον λογαριασμό σας παρακαλώ εισάγετε τα απαραίτημα στοιχεία.

Username:  isSysAdmin() == 1

Password:

**SIGN UP** **LOGIN**

localhost:8080/CINEThisio/adminIndex.jsp



Σημ: κατά το login καλείται και το αντίστοιχο webservice που αποθηκεύει τα logs

To dashboard του administrator:

**Dashboard**

CINE Thisio Admin

Total Users: 4      Total Movies: 2      Total Theaters: 1      Today's Messages: 0

**Dashboard**    **Messages**    **Profile**    **Homepage**    **Logout**

Add new movie

Movie Title..: 2018  
Movie Stars..: Movie Description..  
Duration of the movie in minutes: 120  
Min age of the viewers (years old): 10  
**Add Movie**

Add new theater

New theater ID: 2  
Add the capacity of the theater: 150  
**Add Theater**

Add to Schedule

Εισαγωγή προβολής

Επιλογή αίθουσας προβολής: 1  
Επιλογή ταινίας προβολής: --  
Επιλογή ώρας προβολής: mm/dd/yyyy --:-- --  
 Η ταινία είναι διαθέσιμη για το κοινό

**Add to Schedule**

Schedule

mm/dd/yyyy  DATE: 2021-08-24

| ID | Theater | Movie      | Air Date         | Available Spots | Total Spots | Is Movie Available? |
|----|---------|------------|------------------|-----------------|-------------|---------------------|
| 1  | 1       | La La Land | 2021-08-24 21:45 | 198             | 200         | YES                 |
| 2  | 1       | Il Postino | 2021-08-24 23:45 | 200             | 200         | YES                 |

Αν κάνει είσοδο στο σύστημα ένας απλός χρήστης τότε μεταφέρεται στην αρχική σελίδα και το μενού του πλέον είναι ανανεωμένο :

Welcome Back,  
mixalis!

HOMEPAGE

E-TICKETS

ABOUT US

CONTACT US

PROFILE

LOG OUT

## Profile.java

Ας δούμε τώρα αναλυτικότερα τα webservices που ορίζονται στην κλάση Profile. Όπως έχει ήδη αναφερθεί στην αριστερή πλευρά φαίνονται τα 15 τελευταία logins του χρήστη, ενώ στην αριστερή πλευρά ένα webservice έχει ήδη ενημερώσει την φόρμα και περιμένει τον χρήστη για υποβολή.

The screenshot shows the 'Profile' page with two main sections: 'Log File' and 'Account Information'. In the 'Log File' section, a red arrow points from the text 'Κάθε φορά που ο χρήστης πατάει login (πάνω πάνω φαίνονται το τέλευταίο)' to the log entries: 2021-08-25 20:44:14.0, 2021-08-25 20:44:07.0, and 2021-08-25 20:42:30.0. In the 'Account Information' section, another red arrow points from the text 'Χρειάζεται ένα password, οτι και να βάλουμε αυτό θα κάνει UPDATE στην βάση' to the 'Password (put the old one or a new one)' input field. A third red arrow points from the 'SUBMIT' button to the 'CHAT' button.

## Account Information

Παρακαλώ υποβάλετε την φόρμα προκειμένου να ανανεώσετε τα στοιχεία σας!

Αλλάζοντας λοιπόν όλα τα στοιχεία που θέλουμε, ένα τρίτο WS αναλαμβάνει το UPDATE του χρήστη στον αντίστοιχο πίνακα

The screenshot shows the 'Account Information' page. A red arrow points from the 'SUBMIT' button to the 'CHAT' button. Another red arrow points from the 'SUBMIT' button to a success message box. The message box contains the text: 'localhost:8080 says Your personal info have been updated!' and an 'OK' button.

## Contact.java

Στην σελίδα contactus.jsp καλείται το μοναδικό WS της κλάσης όπου δίνει την δυνατότητα στον χρήστη να στείλει email μέσω της html φόρμας.

# Contact Us

NAME                            EMAIL

Giannis Papaxristou        f5890620@gmail.com

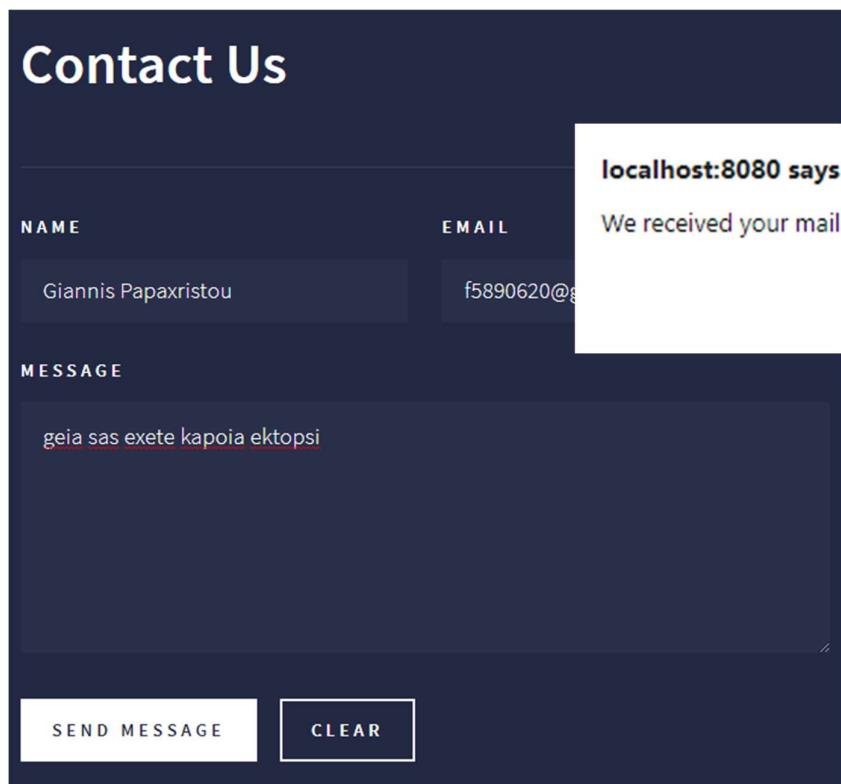
MESSAGE

geia sas exete kapoia ektopsi

SEND MESSAGE              CLEAR

localhost:8080 says  
We received your mail and we will reply you as soon as we can

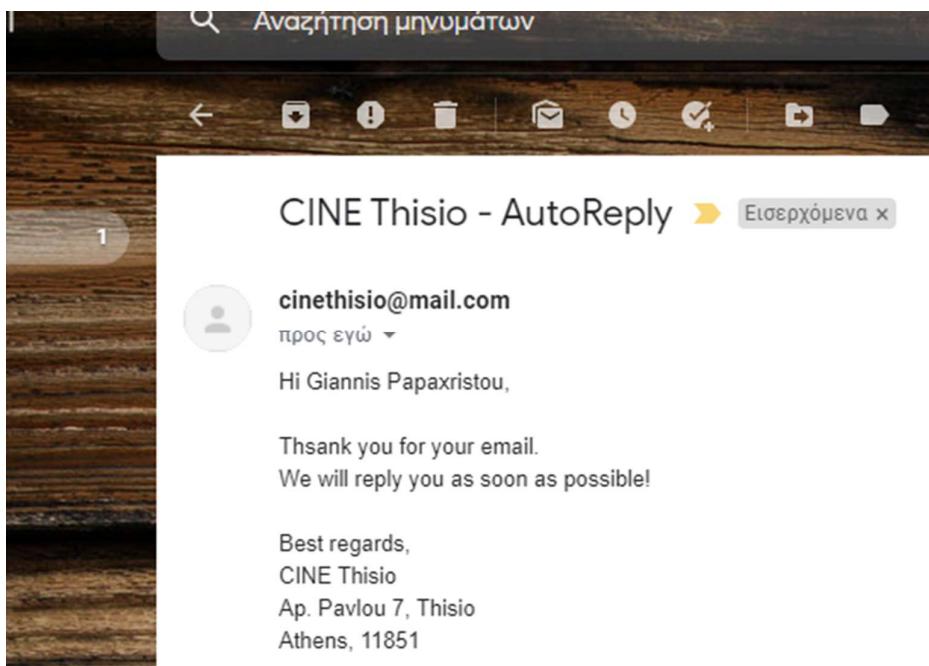
OK



To email που έρχεται στο [cinethisio7@gmail.com](mailto:cinethisio7@gmail.com) :

New mail from the website contact form! Εισερχόμενα X

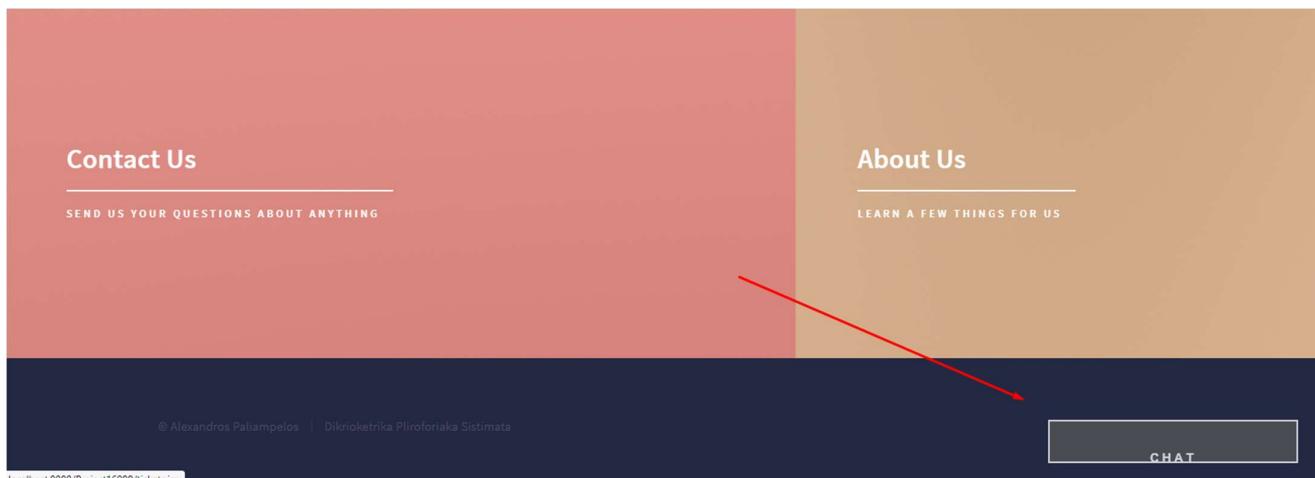
 cinethisio7@gmail.com  
προς εγώ ▾  
**Name:** Giannis Papaxristou  
**Email:** [f5890620@gmail.com](mailto:f5890620@gmail.com)  
**Message:** geia sas exete kapoia ekptosi



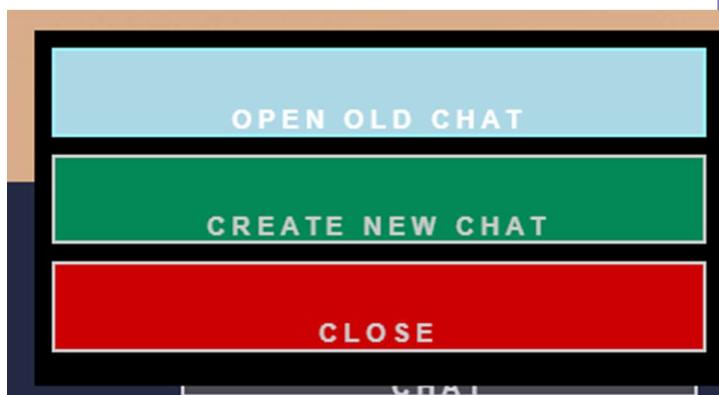
To autoreply mail που πηγαίνει στο δηλωμένο email

## Chat.java

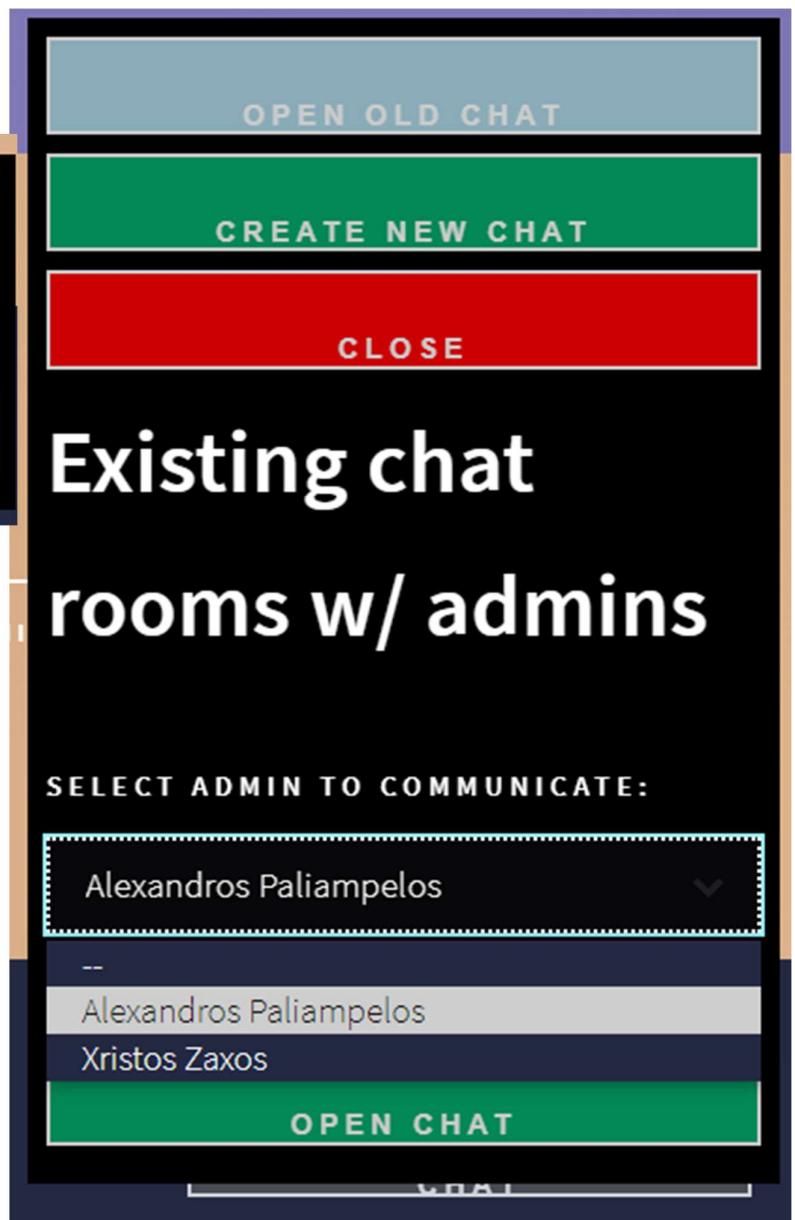
Μεταβαίνοντας σε οποιαδήποτε σελίδα ο απλός χρήστης μπορεί από το κάτω δεξιά μέρος της οθόνης να ανοίξει το chat.



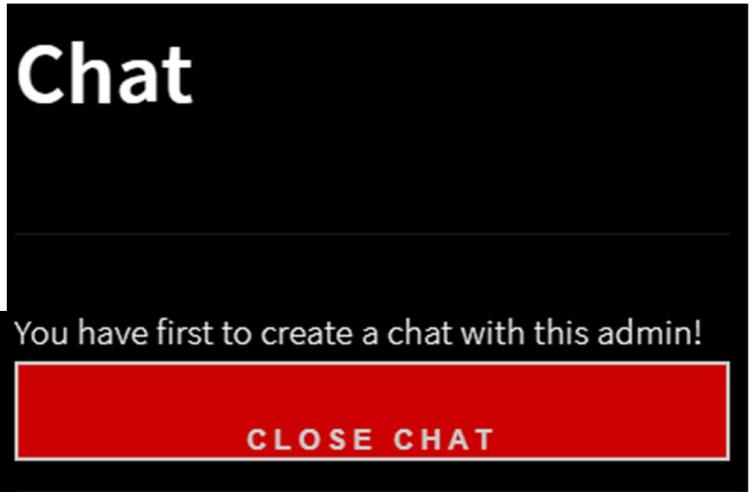
Πατώντας λοιπόν στο κουμπί μπορεί είτε να ανοίξει κάποιο παλιό chat είτε να δημιουργήσει ένα καινούργιο (αριστερή είκόνα). Στα δεξιά βλέπουμε ότι πάτησε να ανοίξει OLD CHAT



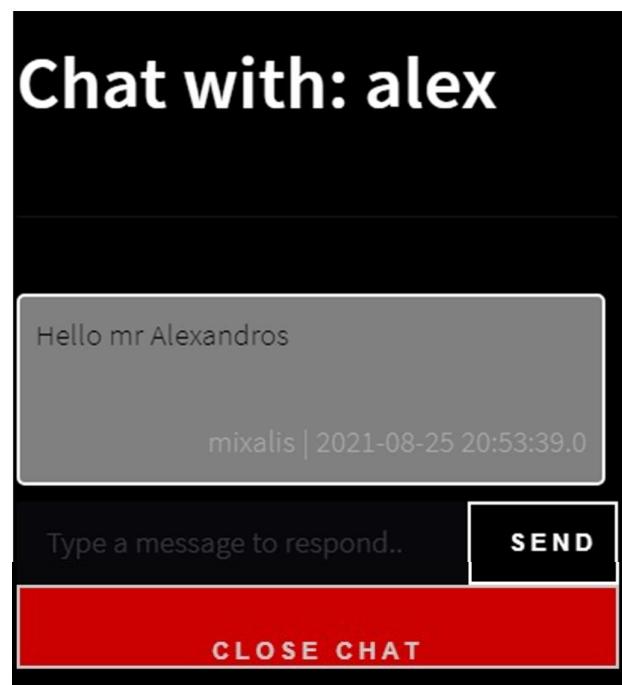
'Ενα webservice getAdmins μάς προτρέπει να επιλέξουμε έναν από τους διαθέσιμους διαχειριστές προκειμένου να συνομιλήσουμε μαζί του.



Πατώντας σε χρήστη που δεν υπάρχει κάποιο chatroom, έχουμε ένδειξη ότι θα πρέπει πρώτα να δημιουργήσουμε ένα chat, στο επόμενο βήμα πατάμε το κουμπί "CREATE NEW CHAT"



Ένα διαφορετικό webservice getAdmins μάς προτρέπει να επιλέξουμε έναν από τους διαθέσιμους διαχειριστές προκειμένου να συνομιλήσουμε μαζί του ενώ στην συνέχεια πληκτρολογούμε το μήνυμα μας στο πεδίο, ύστερα πατάμε "SEND MSG". Το ws pushMessage() αναλαμάνει να αποθηκευεθεί στην βάση και άμα ξανα ανοίξουμε το chat όπως στο πρώτο βήμα θα δούμε το τελικό αποτέλεσμα (ws – getMsgs()).



Ας στείλουμε άλλο ένα μήνυμα πριν πάμε στην μεριά του διαχειριστή

# Chat

Hello mr Alexandros

mixalis | 2021-08-25 20:53:39.0

eiste edw?

SEND

CLOSE CHAT

Κάνοντας τώρα login με τα στοιχεία του alex, μεταβαίνουμε αυτόματα στο admin dashboard και βλέπουμε ότι σήμερα έχουμε 2 μηνύματα:

# Chat with: alex

Hello mr Alexandros

mixalis | 2021-08-25 20:53:39.0

eiste edw

mixalis | 2021-08-25 20:55:49.0

Type a message to respond..

SEND

CLOSE CHAT

Dashboard

CINE Thisio Admin

Dashboard

Messages

Profile

Homepage

Logout

Total Users: 4

Total Movies: 2

Total Theaters: 1

Today's Messages: 2

Add new movie

Movie Title..: 2018

Movie Stars..: Movie Description..:

Duration of the movie in minutes: 120

Min age of the viewers (years old): 10

Add Movie

Add new theater

New theater ID: 2

Add the capacity of the theater: 150

Add Theater

Add to Schedule

Επιλογή αιθουσας προβολής: 1

Επιλογή ταινιας προβολής: --

Schedule

mm/dd/yyyy

Get Schedule

The screenshot shows the CINE Thisio Admin dashboard. On the left, there's a sidebar with links for Dashboard, Messages (which is currently selected), Profile, Homepage, and Logout. The main area displays four key statistics: Total Users (4), Total Movies (2), Total Theaters (1), and Today's Messages (2). Below these stats are two forms: one for adding a new movie (Movie Title, Movie Stars, Duration, Min age) and one for adding a new theater (New theater ID, Add the capacity of the theater). At the bottom, there are sections for 'Add to Schedule' (with dropdowns for movie and theater selection) and 'Schedule' (with a date input field and a 'Get Schedule' button).

Από το αριστερό μενού μεταβαίνουμε στην σελίδα adminMessages.jsp όπου ο διαχειριστής καλείται να επιλέξει μέσω ενός WS έναν από τους χρήστες που του έχουν στείλει μήνυμα. Στο παράδειγμα μας τα μηνύματα τα έστελνε ο "mixalis"

The screenshot shows the 'Messages' section of the admin interface. On the left, there's a sidebar with links for Dashboard, Messages (which is currently selected), Profile, Homepage, and Logout. The main area has four summary boxes: Total Users (4), Total Movies (2), Total Theaters (1), and Today's Messages (2). Below these is a dropdown menu titled 'Select User to open chat' with two options: 'Nikolas Papadopoulos' and 'Mike Papazoglou'. A large green button labeled 'Open Chat' is positioned below the dropdown. Red arrows point from the text labels 'Nikolas Papadopoulos' and 'Open Chat' to their respective elements.

Πατώντας openChat, καλείτε το WS getMsgs() και πλέον μπορούμε να δούμε το chat στα δεξιά και να απαντήσουμε:

The screenshot shows the same admin interface after opening a chat with 'alex'. The right side of the screen displays the chat window with the title 'Chat with: alex'. It shows the following messages:  
User: Hello mr Alexandros  
User: eiste edw  
alex: Pos mporo na sas voithiso  
The user's message input field is shown at the bottom left, and a 'Send' button is at the bottom right. Red arrows point from the user's message input field and the 'Send' button towards the corresponding elements in the screenshot above.

Τελικό chat διαχειριστή – απλού χρήστη

The screenshot shows the final chat interface. The administrator's messages are on the left, and the user's messages are on the right:  
Administrator: Hello mr Alexandros  
Administrator: eiste edw  
User: Pos mporo na sas voithiso  
Administrator: Type a message to respond..  
Administrator: SEND  
Administrator: CLOSE CHAT  
Red arrows point from the administrator's message input field and the 'SEND' button to the corresponding elements in the screenshot above.

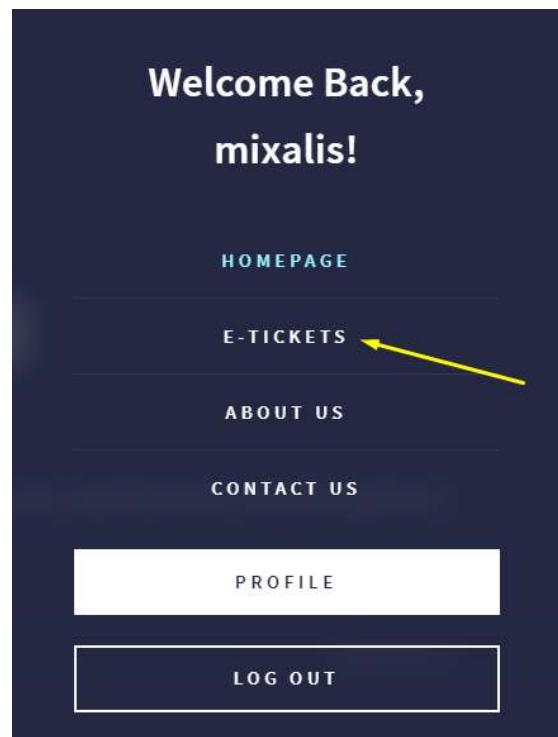
Και εδώ φαίνονται τα μηνύματα στην βάση:

| chatID | type | user    | sysadmin | message                   | date                |
|--------|------|---------|----------|---------------------------|---------------------|
| 1      | U    | nick    | xristos  | kalispera sas             | 2021-08-22 01:47:24 |
| 1      | U    | nick    | xristos  | exw mia aporia            | 2021-08-22 02:26:55 |
| 3      | U    | nick    | alex     | hello alex!               | 2021-08-22 02:32:09 |
| 3      | A    | nick    | alex     | pes mou nick              | 2021-08-22 02:32:41 |
| 3      | U    | nick    | alex     | ola kala                  | 2021-08-23 13:06:18 |
| 3      | A    | nick    | alex     | kali sinexeia             | 2021-08-23 13:06:42 |
| 7      | U    | mixalis | alex     | Hello mr Alexandros       | 2021-08-25 20:53:39 |
| 7      | U    | mixalis | alex     | eiste edw                 | 2021-08-25 20:55:49 |
| 7      | A    | mixalis | alex     | Pos mporo na sas voithiso | 2021-08-25 20:58:47 |

### Ticket.java

Για την μετάβαση στο σύστημα έκδοσης εισιτηρίων ο χρήστης αρκεί να πάει μέσω του menu στην αντίστοιχη σελίδα (tickets.jsp) (Προϋπόθεση να έχει κάνει login)

Στην σελίδα αυτή έχουμε ένα WS που θα μας επιστρέψει το πρόγραμμα για κάποια ημερομηνία. Εισάγουμε την αντίστοιχη ημερομηνία και στην συνέχεια πατάμε "GET SCHEDULE".



## Book Tickets

ΔΙΑΛΕΞΤΕ ΗΜΕΡΟΜΗΝΙΑ

Συμπληρώστε τα στοιχεία για κράτηση εισητηρίων

ΣΤΟΙΧΕΙΑ ΕΠΙΚΟΙΝΩΝΙΑΣ

Mike

Papazoglou

Ο χρήστης πλέον μπορεί να πάρει όλες τις πληροφορίες που χρειάζεται προκειμένου να κλείσει τα εισιτήρια. Θα χρειαστεί να γνωρίζει τον ID του προγράμματος μόνο ο οποίος φαίνεται στα αριστερά.

## Book Tickets

ΔΙΑΛΕΞΤΕ ΗΜΕΡΟΜΗΝΙΑ

mm/dd/yyyy

DATE: 2021-08-24

| ID | Theater | Movie      | Air Date         | Available Spots | Total Spots |
|----|---------|------------|------------------|-----------------|-------------|
| 1  | 1       | La La Land | 2021-08-24 21:45 | 198             | 200         |
| 2  | 1       | Il Postino | 2021-08-24 23:45 | 192             | 200         |

Σε περίπτωση που θέλει να δει περισσότερες πληροφορίες για την ταινία, πατώντας στο όνομα της ανοίγεται ένα νέο παράθυρο με όλες τις πληροφορίες που φέρνει το αντίστοιχο WS από τον πίνακα movie.

The screenshot shows a browser window with the URL `localhost:8080/CINEThisio/movie.jsp?movieID=2`. A yellow box highlights the URL bar, and a yellow arrow points from it to the movie title 'Il Postino (1994)' displayed prominently below the header.

# Il Postino (1994)

## Description

A simple Italian postman learns to love poetry while delivering mail to a famous poet.

**Stars:** Massim Troisi      **Duration:** 110'      **Age:** 13

Συνεχίζοντας τώρα στην σελίδα tickets.jsp και αφού ο χρήστης έχει δει το σημερινό πρόγραμμα, τον προτρέπουμε να συμπληρώσει την φόρμα για να κλείσει τα εισιτήρια του. Ένα webservice αναλαμβάνει την συμπλήρωση των στοιχείων επικοινωνίας (μπορούν να αλλάξουν) και ο χρήστης πρέπει απλά να εισάγει επιθυμητό αρ. εισιτηρίων και id προγράμματος. Ένα άλλο ws αναλαμβάνει τις ενέργειες μετά το submit της φόρμας. Εννοείται ότι αν το πρόγραμμα δεν υπάρχει ή αν το πλήθος των εισιτηρίων είναι περισσότερο από το διαθέσιμο, εμφανίζεται το αντίστοιχο μήνυμα.

## Συμπληρώστε τα στοιχεία για κράτηση εισητηρίων

| ΣΤΟΙΧΕΙΑ ΕΠΙΚΟΙΝΩΝΙΑΣ   |                 |
|-------------------------|-----------------|
| Mike                    | Papazoglou      |
| alex-otypas@hotmail.com |                 |
| ΠΛΗΘΟΣ ΕΙΣΙΤΗΡΙΩΝ       | ID ΠΡΟΓΡΑΜΜΑΤΟΣ |
| 4                       | 2               |
| <b>SUBMIT</b>           |                 |

CINE Thisio - Successful booking!



**localhost:8080 says**

Successful booking! More details at your email!

## BOOKING DETAILS

**OK**

Name: Mike Papazoglou

Date: 2021-08-24 23:45

Movie: Il Postino

Theater: 1

Number of tickets: 4

Σε περίπτωση που όλα είναι σωστά τότε μας έρχεται email με μία ανασκόπηση καθώς και μία προπροπή για να πληρώσουμε τα εισιτήρια στο σινεμά.

Total cost: 4x10 = 40 EUR

Pay at cinema with the following QR code:



## Admin.java

Τώρα θα αναλύσουμε την μεριά του διαχειριστή για τα webservices που χρησιμοποιούνται στο σύστημα έκδοσης εισιτηρίων

**Η πρώτη** γραμμή (bootstrap row) στο Admin Dashboard έχει 3 παιδία για add δηλαδή HTTP POST ή INSERT (in DB).

The screenshot shows three separate input forms arranged horizontally:

- Add new movie:** Fields include "Movie Title.." (La vita è bella), "Year.." (1997), "Stars.." (Roberto Benigni), "Description.." (When an open-minded J), "Duration of the movie in minutes" (116), and "Min age of the viewers (years old)" (13). A button "Add Movie" is at the bottom.
- Add new theater:** Fields include "New theater ID:" (3) and "Add the capacity of the theater" (150). A button "Add Theater" is at the bottom.
- Add to Schedule:** Fields include "Εισαγωγή προβολής" (Movie title dropdown), "Επιλογή αιθουσας προβολής" (Theater dropdown), "Επιλογή ωρας προβολής" (Date dropdown), and a checkbox "Η ταινία είναι διαθέσιμη για το κοινό". A button "Add to Schedule" is at the bottom.

Προσθέτοντας μία νέα ταινία καλείται μόνο ένα WS το οποίο κάνει INSERT στην βάση όπως φαίνεται παρακάτω:

### Add new movie

A confirmation message from "localhost:8080 says" stating "Movie added SUCCESSFUL" with an "OK" button. The input fields for the movie "La vita è bella" are shown again: title, year, stars, description, duration, and age.

```
mysql> select * from movie;
+----+-----+-----+-----+-----+-----+
| mID | title          | year | stars           | description                                | duration | age  |
+----+-----+-----+-----+-----+-----+
1	La La Land	2016	Ryan Gosling and Emma Stone	La La Land is a 2016 American musical film about a young woman who falls in love with a man she meets on a road trip.	130	13
2	Il Postino	1994	Massim Troisi	A simple Italian postman learns to love again after his wife leaves him.	110	10
3	La vita e bella	1997	Roberto Benigni	When an open-minded Jewish librarian falls in love with a Nazi officer, and imagination to protect his son from the dangers around their camp.	116	13
+----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Προσθέτοντας μια νέα αίθουσα κινηματογράφου έχουμε ένα πεδίο readonly για το τη νέα αίθουσα που προκύπτει από προηγούμενη κληση WS που μας γυρνάει τον αριθμό των αιθουσών.

### Add new theater

New theater ID: 2

Add the capacity of the theater 150

**Add Theater**

### Add new theater

New theater ID: 2

Add the capacity of the theater 150

**Add Theater**

localhost:8080 says

Theater added SUCCESSFUL

**OK**

```
mysql> select * from theater;
+-----+-----+
| tID | totalSpots |
+-----+-----+
| 1   |      200   |
| 2   |      150   |
+-----+-----+
```

Τέλος για την προσθήκη ταινίας στο πρόγραμμα ένα WS μας δείχνει τις διαθέσιμες ταινίες για προβολή μέσω select tag και έπειτα μόλις πατήσουμε add to schedule ένα άλλο WS θα κάνει το insert στον πίνακα.

### Add to Schedule

#### Εισαγωγή προβολής

Επιλογή αίθουσας προβολής 1

Επιλογή ταινίας προβολής --

Επιλογή ώρας προβολής

mm/dd/yyyy --:-- --

Η ταινία είναι διαθέσιμη για το κοινό

**Add to Schedule**

### Add to Schedule

#### Εισαγωγή προβολής

Επιλογή αίθουσας προβολής 2

Επιλογή ταινίας προβολής

La vita e bella

--

La La Land

Il Postino

La vita e bella

πλήρης

0 PM

προβολή

διαθέσιμη για το κοινό

**Add to Schedule**

localhost:8080 says

Movie added to schedule

OK

```
mysql> select * from schedule;
+-----+-----+-----+-----+-----+-----+-----+
| sID | theaterID | movieID | dateOfAir           | availbleSpots | totalSpots | movieUP |
+-----+-----+-----+-----+-----+-----+-----+
1	1	1	2021-08-24 21:45:00	198	200	1
2	1	2	2021-08-24 23:45:00	192	200	1
3	2	3	2021-08-28 22:00:00	150	150	1
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

**Η δεύτερη** γραμμή (bootstrap row) στο Admin Dashboard έχει 3 παιδία για edit δηλαδή HTTP PUT ή UPDATE (in DB).

Πάμε τώρα να επεξεργαστούμε μια ταινία, με την χρήση ενός WS θα επιλέξουμε μια από τις διαθέσιμες ταινίες.

Στο επόμενο βήμα ένα άλλο WS θα μας βοηθήσει για το συμπλήρωμα της φόρμας και ο χρήστης το μόνο που αλλάζει είναι η ηλικία

'Ένα τρίτο WS τώρα θα κάνει update στην βάση μας

Edit Movie

Edit Movie

Επιλογή ταινίας

Il Postino

Select movie



Il Postino

1994

Massim Troisi

A simple Italian postmar

Duration of the movie in minutes

110

Min age of the viewers (years old)

10

Edit Movie



localhost:8080 says

Movie has been updated!

OK

```
mysql> select * from movie;
+-----+-----+-----+-----+-----+-----+-----+
| mID | title    | year   | stars          | description                                | duration | age  |
+-----+-----+-----+-----+-----+-----+-----+
| 1   | La La Land | 2016 | Ryan Gosling and Emma Stone | La La Land is a 2016 American musical romantic comedy-drama film written and directed by Damien Chazelle. | 130     | 13   |
| 2   | Il Postino | 1994 | Massim Troisi   | A simple Italian postman learns to love poetry while delivering mail to a famous poet.          | 110     | 13   |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from movie;
+-----+-----+-----+-----+-----+-----+-----+
| mID | title    | year   | stars          | description                                | duration | age  |
+-----+-----+-----+-----+-----+-----+-----+
| 1   | La La Land | 2016 | Ryan Gosling and Emma Stone | La La Land is a 2016 American musical romantic comedy-drama film written and directed by Damien Chazelle. | 130     | 13   |
| 2   | Il Postino | 1994 | Massim Troisi   | A simple Italian postman learns to love poetry while delivering mail to a famous poet.          | 110     | 10   |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Στο αμέσως επόμενο κουτάκι στα δεξιά μπορούμε να αλλάξουμε την χωρητικότητα μιας αίθουσας.

Ακολουθείται ίδια λογική. Ένα web service μας δείχνει τις διαθέσιμες επιλογές, στη συνέχεια ένα άλλο μας δείχνει την υπάρχουσα χωρητικότητα, και ο χρήστης αλλάζει την χωρητικότητα από 150 ήταν σε 120.

Το 3<sup>o</sup> WS θα κάνει UPDATE τον πίνακα με τα στοιχεία της φόρμας.

Edit Theater

Επιλογή αίθουσας

--  
ID: 1  
ID: 2  
ID: 3



Edit Theater

Edit the capacity of the theater

120

localhost:8080 says

Theater has been updated!

```
mysql> select * from theater;
+----+-----+
| tID | totalSpots |
+----+-----+
1	200
2	150
3	120
+----+-----+
```

Στο δεξιότερο section της γραμμής μπορούμε να αλλάξουμε μόνο την ημερομηνία προβολής και το αν θα είναι μια ταινία διαθέσιμη για κράτηση εισιτηρίων. Ομοίως χρησιμοποιούνται άλλα 3 ws.

### Edit Schedule

Επιλογή προγράμματος ID: 3 ▾

Select Schedule

- 
- ID: 1
- ID: 2
- ID: 3**
- ID: 4

### Edit Schedule

Edit the air date of selected schedule

08/28/2021 10:30 PM

Η ταινία είναι διαθέσιμη για το κοινό

**Edit Schedule**

localhost:8080 says

Schedule has been updated!

OK

```
mysql> select * from schedule;
+----+-----+-----+-----+-----+-----+-----+
| sID | theaterID | movieID | dateOfAir           | availbleSpots | totalSpots | movieUP |
+----+-----+-----+-----+-----+-----+-----+
1	1	1	2021-08-24 21:45:00	198	200	1
2	1	2	2021-08-24 23:45:00	192	200	1
3	2	3	2021-08-28 22:00:00	150	150	1
4	1	1	2021-08-28 22:00:00	200	200	1
+----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from schedule;
+----+-----+-----+-----+-----+-----+-----+
| sID | theaterID | movieID | dateOfAir           | availbleSpots | totalSpots | movieUP |
+----+-----+-----+-----+-----+-----+-----+
1	1	1	2021-08-24 21:45:00	198	200	1
2	1	2	2021-08-24 23:45:00	192	200	1
3	2	3	2021-08-28 22:30:00	150	150	0
4	1	1	2021-08-28 22:00:00	200	200	1
+----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Θα μπορούσαμε τώρα να είχαμε μία 3<sup>η</sup> γραμμή (bootstrap row) που να υλοποιούσα διαδικασίες διαγραφής, δηλαδή HTTP DELETE ή DELETE (in DB).

Όμως δεν υλοποιούνται τέτοιες διακασίες διαγραφής ταινίας, αιθουσας προγράμματος ή χρήστη.

Στην πραγματικότητα η 3<sup>η</sup> γραμμή μας δείχνει το πρόγραμμα για κάποια ημέρα, όπως στο tickets.jsp Η διαφορά είναι ότι εδώ μας δείχνει όλες τις ταινίες, δηλ. και αυτές που έχουν σήμανση movieUP==0.

#### Schedule

| mm/dd/yyyy |         | Get Schedule    | DATE: 2021-08-28 |                 |             |  | Is Movie Available? |
|------------|---------|-----------------|------------------|-----------------|-------------|--|---------------------|
| ID         | Theater | Movie           | Air Date         | Available Spots | Total Spots |  |                     |
| 3          | 2       | La vita e bella | 2021-08-28 22:30 | 150             | 150         |  |                     |
| 4          | 1       | La La Land      | 2021-08-28 22:00 | 200             | 200         |  |                     |

```
mysql> select * from schedule;
+-----+-----+-----+-----+-----+-----+-----+
| sID | theaterID | movieID | dateOfAir           | availbleSpots | totalSpots | movieUP |
+-----+-----+-----+-----+-----+-----+-----+
1	1	1	2021-08-24 21:45:00	198	200	1
2	1	2	2021-08-24 22:45:00	102	200	1
3	2	3	2021-08-28 22:30:00	150	150	0
4	1	1	2021-08-28 22:00:00	200	200	1
+-----+-----+-----+-----+-----+-----+-----+
```