

CS2102 Final Report

Online Booking System

Movie Tickets Booking System

Hu Wenyan (A0119397R)

Lu Yuehan (A0119387U)

Luan Wenhao (A0119541J)

07 November 2014

Objectives

In this project, we aim to design a online movie tickets booking application to facilitate convenient searching and booking for users. User will be allowed to search catalogue of movie based on the **movie title,movies attributes**(such as actors, director and movie descriptions),**time slots, cinema name, cinema location, and ticket price range**. In addition, after logging in, users are able to book movie tickets, and modify or cancel their booking afterwards.

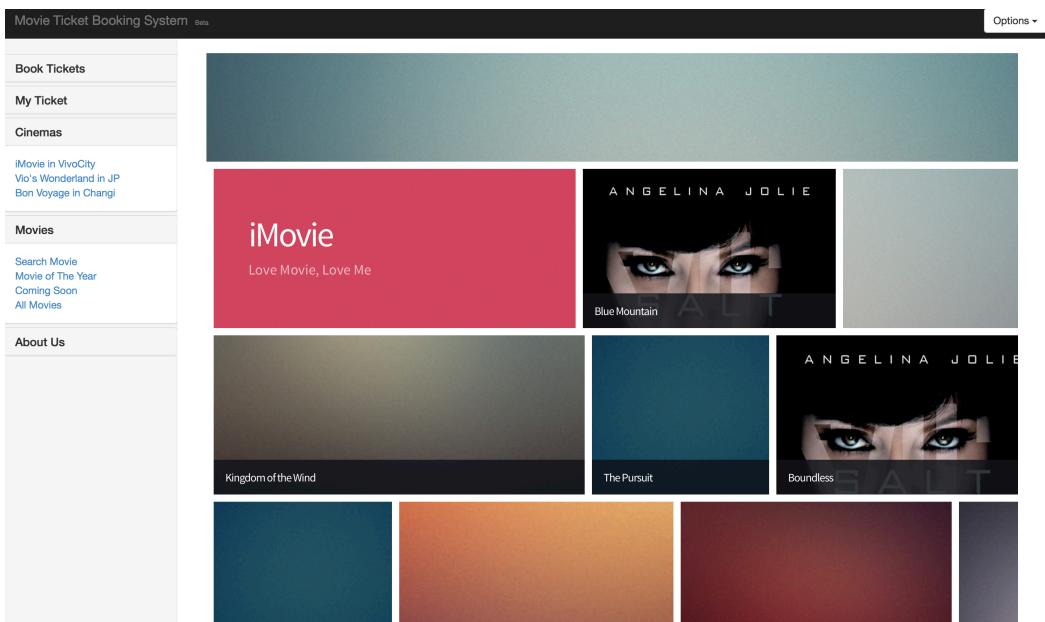
Implementation

The structure of our web application can be categorised into two major part including front end components and back end components. The front end, basically user interface, is responsible for directly interacting with users including taking in issued commands and displaying feedback results. The back end, which is hidden from users' aspects, processes issued command passed by UI and returns feedback.

Components

User Interface

In order to construct an effective user interface for our web applications, we have applied a right mix of a variety of web development languages, tools and platforms including *HTML, CSS, Javascript* and *AJAX*. We aim to implement a well-thought-out interaction design that reflects the perspective of our users and curtail to their needs.



Web Server

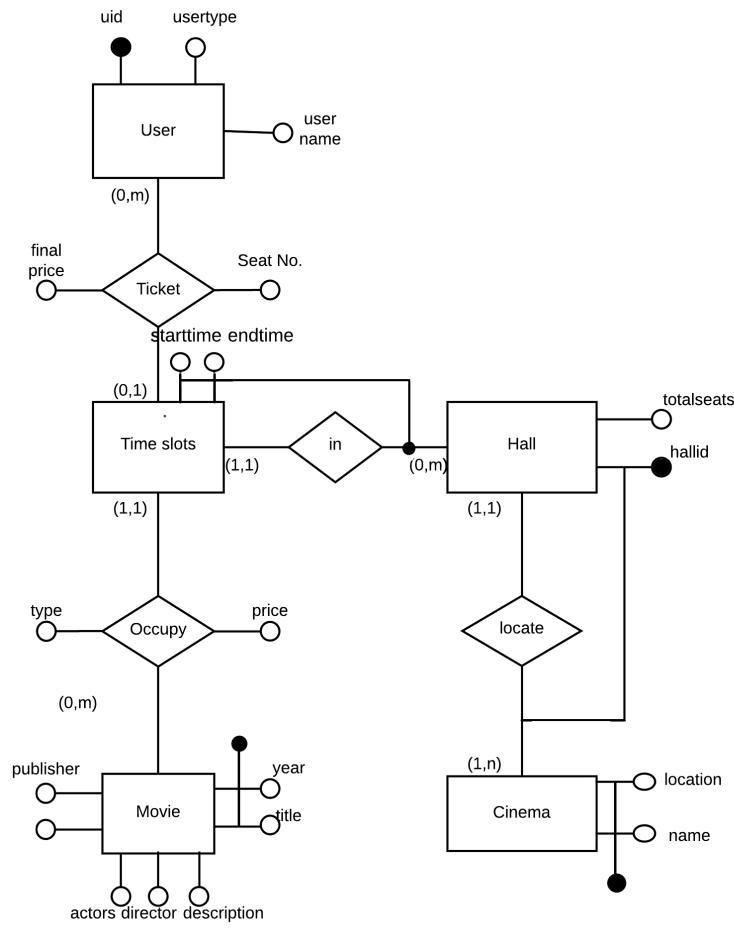
We use the SoC sub-zone to host our website. The interaction medium between the user interface and the server with our database is PHP language which will query or manipulate the database according to input from the user interface using `POST` method. With the information from users and the database, our PHP scripts will generate HTML codes that will later serve as part of the user interface.

Database

We stay with Oracle SQL which is discussed in the module lectures. We try to apply the knowledge we learnt from the course to this website project and it is also good for us to reflect on what we have learnt with this hands-on practice.

Database Schema

We designed the database schema following our ER Diagram which has been discussed in our progress report.

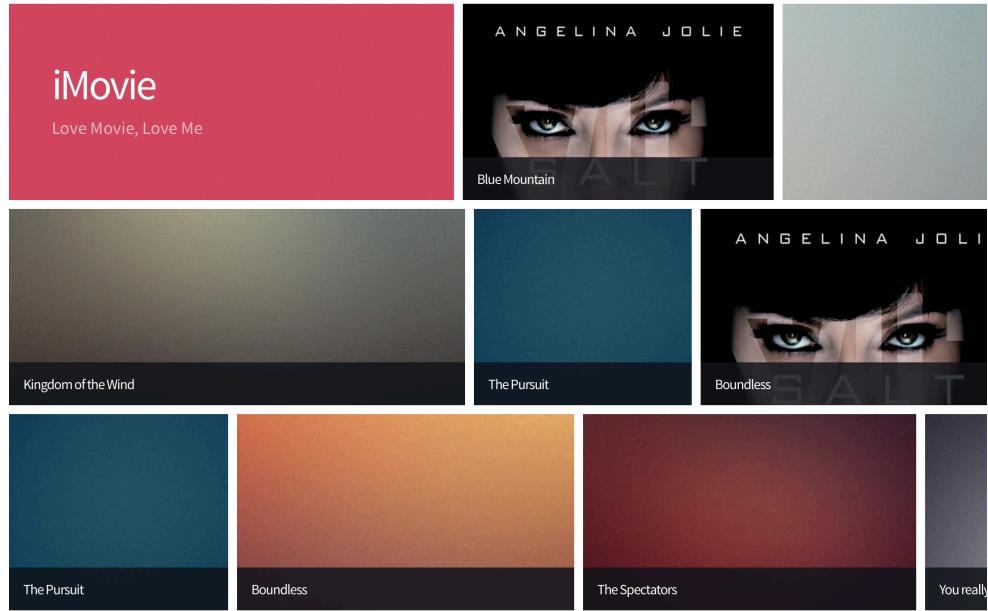


Functionalities & SQL implementation

Browsing

- **Browsing Cinema**

Users are able to view all the cinemas and the movies showing in the selected cinemas by simply click on any of the three cinema links in the navigation bar, the page will display information about the cinema and its movies:

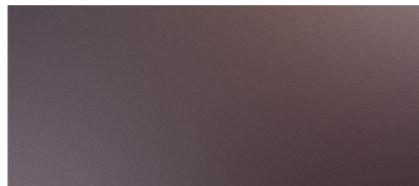


```
SELECT * FROM CINEMA;
```

- **Display Movie list**

Users are able to view all the movies by simply click the All Movie button in the navigation bar, the page will display the whole list of movies:

ENJOY MOVIE

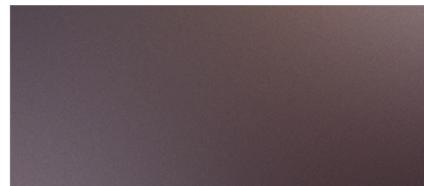


CRAS

Director: Jeremy

Actors: Yuri

Description: metus. Aliquam erat

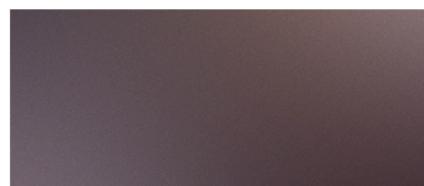
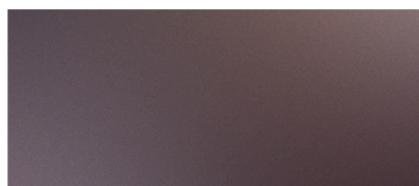


CONVALLIS CONVALLIS

Director: Hanna

Actors: Talon

Description: consequat nec, mollis



```
SELECT DISTINCT TITLE FROM MOVIE;
```

Searching

- **Search for Movie**

To facilitate this search, we have implemented the SQL query code as follows:

```
SELECT * FROM MOVIE WHERE TITLE LIKE '%"$Key"%';
```

- **Search for Booked Ticket**

User can search for their booked ticket by simply enter their unique user ID.



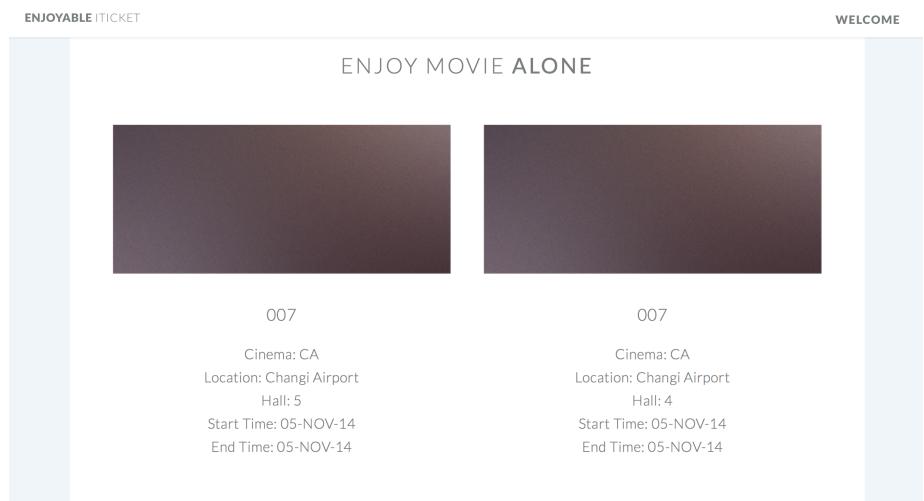
- When user click submit button with input field blank:

The interface features a light gray background with a white central search area. At the top center is a small bar chart icon. Below it is a text input field labeled "MY ID:" followed by a large empty rectangular input field. A prominent button at the bottom of the search area is labeled "SEARCH MY TICKET". The entire search area is enclosed in a thin black border. In the center of the page, below the search area, is a message: "PLEASE ENTER YOUR ID". Below this message is a smaller button labeled "SEE MORE".

- When the issued User ID has no corresponding entry in ticket database situation, the page will prompt:

The interface is identical to the one above, featuring a light gray background with a white central search area. The bar chart icon is at the top center. Below it is a text input field labeled "MY ID:" followed by a large empty rectangular input field containing the number "10000". A prominent button at the bottom of the search area is labeled "SEARCH MY TICKET". The entire search area is enclosed in a thin black border. In the center of the page, below the search area, is a message: "YOU HAVE NO TICKET YET, START BOOKING NOW!". Below this message is a smaller button labeled "SEE MORE".

- For successful search, the result will be displayed as:



To facilitate this search, we have implemented the SQL query code as follows

```

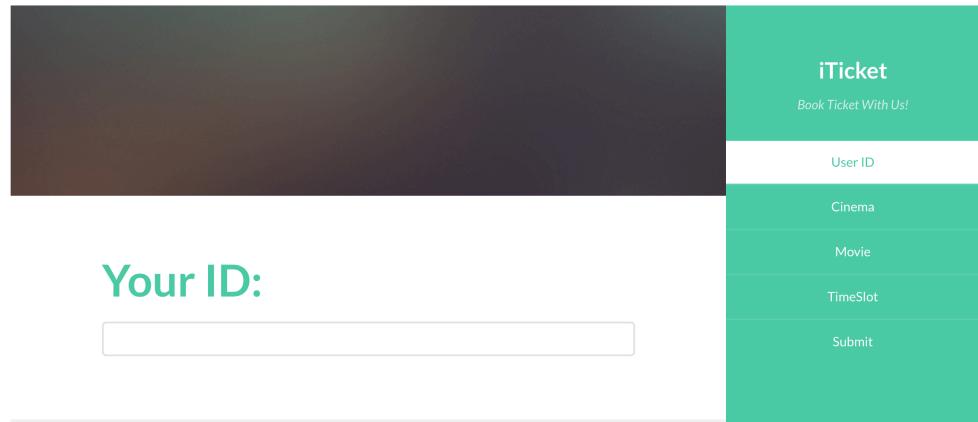
SELECT
T.SUBSCRIBERID,S.USERNAME,
O.MOVIETITLE,Cn.NAME,
Cn.LOCATION,H.HALLID,
T.STARTTIME,T.ENDTIME
FROM
Ticket T, Occupy O, Subscriber S,
Cinema Cn,Hall H
WHERE
(T.STARTTIME = O.STARTTIME AND T.ENDTIME =
O.ENDTIME AND T.HALLID = O.HALLID)
AND (H.NAMEOFCINEMA =
Cn.NAME AND H.LOCATIONOFCINEMA = Cn.LOCATION)
AND T.SUBSCRIBERID = S.SUBSCRIBERID
AND S.SUBSCRIBERID = $USER_ID;

```

Booking

- **Book Ticket**

User can book ticket with us



Use Case Flow:

- ID Check
 - Note that the User ID validation check is performed immediately when user finishes typing (input field loses focus). It is done by querying the database with the user ID input.

Your ID:

✓

Your ID:

✗

```
SELECT COUNT(*) AS NUM FROM SUBSCRIBER WHERE SUBSCRIBERID = '$cName';
```

- o Fetch Cinema List

Cinema

Select Cinema

✓ CA

JP

VC

```
SELECT DISTINCT name FROM cinema;
```

- o Fetch Movie List with specified cinema

- Note that whenever the selection in the "Select Cinema" option menu is changed, the Movie List options will change accordingly and immediately by performing an SQL query with AJAX.

Movie

✓ Select Movie

Gone Girl

title1

I love U

King Kong

title3

title2

Hello World

Fighter

Blue Mountain

Blue Dream

007

Bye and Hi

Farewell

```
SELECT
DISTINCT MOVIETITLE
FROM OCCUPY, HALL
WHERE OCCUPY.HALLID = HALL.HALLID
AND HALL.NAMEOFCINEMA = "$cName";
```

- o Fetch available timeslots
 - Note that whenever the selection in the "Select Movie" option menu is changed, the Timeslot List options will change accordingly and immediately by performing an SQL query with AJAX.

Time Slot

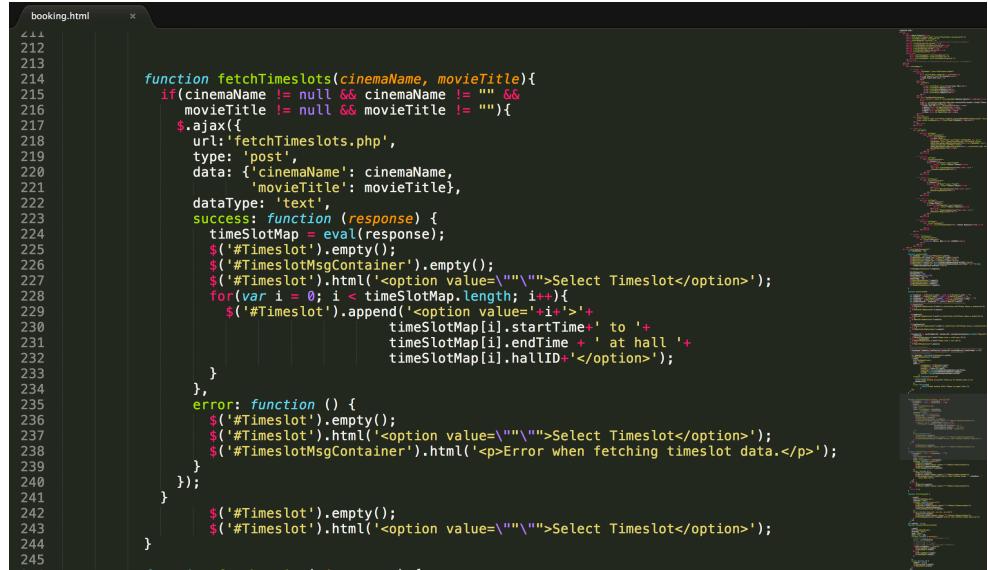
✓ Select Timeslot
00:00:00 05/11/2014 to 00:00:00 05/11/2014 at hall 2

```
INSERT INTO TICKET(SUBSCRIBERID, STARTTIME, ENDTIME,
HALLID)
VALUES
('$userID',
TO_DATE("$sTime", 'HH24:Mi:SS dd/mm/yyyy'),
TO_DATE("$eTime", 'HH24:Mi:SS dd/mm/yyyy'),
"$hallID");
```

Technical Specifications

JavaScript

- With library **jQuery**.
- With data format support **JSON**.
- With client-server communication technique **AJAX**.
- Sample code:
 - o Using **jQuery** with **AJAX** to fetch timeslots of movies.



```

212
213
214     function fetchTimeslots(cinemaName, movieTitle){
215         if(cinemaName != null && cinemaName != "" &&
216             movieTitle != null && movieTitle != ""){
217             $.ajax({
218                 url:'fetchTimeslots.php',
219                 type: 'post',
220                 data: {'cinemaName': cinemaName,
221                       'movieTitle': movieTitle},
222                 dataType: 'text',
223                 success: function (response) {
224                     timeSlotMap = eval(response);
225                     $('#Timeslot').empty();
226                     $('#TimeslotMsgContainer').empty();
227                     $('#Timeslot').html('<option value="">Select Timeslot</option>');
228                     for(var i = 0; i < timeSlotMap.length; i++){
229                         $('#Timeslot').append('<option value="'+i+'>' +
230                             timeSlotMap[i].startTime + ' to ' +
231                             timeSlotMap[i].endTime + ' at hall ' +
232                             timeSlotMap[i].hallID+ '</option>');
233                     }
234                 },
235                 error: function () {
236                     $('#Timeslot').empty();
237                     $('#Timeslot').html('<option value="">Select Timeslot</option>');
238                     $('#TimeslotMsgContainer').html('<p>Error when fetching timeslot data.</p>');
239                 }
240             });
241         }
242         $('#Timeslot').empty();
243         $('#Timeslot').html('<option value="">Select Timeslot</option>');
244     }

```

PHP

- With library **OCI8**
- Sample code:
 - Using **PHP** to query database and echo **JSON** to **AJAX**

```

<?php
putenv('ORACLE_HOME=/oraclient');
$dbh = ociLogon('A0119541','crse1410','sid3');

$cName = $_POST["cinemaName"];
$mName = $_POST["movieTitle"];
$sql = "SELECT DISTINCT TO_CHAR(O.STARTTIME, 'HH24:Mi:SS dd/mm/yyyy') AS STARTTIME,
$stid = oci_parse ($dbh,$sql);
oci_execute($stid, OCI_DEFAULT);

echo "[";
$startBit = 1;
while($row = oci_fetch_array($stid)){
    if($startBit != 1){
        echo ',';
    }
    $startBit = 0;
    echo '{"startTime": "'.$row['STARTTIME'].'", "endTime": "'.$row['ENDTIME'].'", "'
}
echo "]";
oci_free_statement($stid);
oci_close($dbh);
?>

```