

## Project Requirements

You are required to develop a database that can be used to support the Online Movie Ticket Service (OMTS), which is an application for the advance purchase of movie tickets from any local theatre. Customers use the service to find out information about movies currently playing in their city and to order advance tickets for specific showings of the movies. Your task is to provide the database and related functionality for this application.

### Data Requirements

Your database must, at a minimum, contain information about the following:

- There are multiple theatre complexes in the city. Each theatre complex contains some number of theatres and has a name, address and a phone number. Each theatre in a theatre complex has a theatre number, a maximum number of seats and a screen size (small, medium or large). You must have at least 3 theatre complexes represented in your project.
- Each current movie has a title, a running time, a rating (G, PG, AA, 14A, R, etc), a plot synopsis, a list of main actors, a director, a production company, the name of the supplier and the start and end dates for the movie's run at the theatre complex. The movie has one or more daily showings at the theatre complex specified by a start time. Each showing has the number of the theatre for the showing, the start time and the number of seats still available.
- A movie supplier has a company name, a company address, a phone number and the name of the contact person at the company.
- Movie information remains in the database even if the movie is no longer playing at any theatres.
- Each OMTS customer must register with the service. Once they have done so, they use the account number and password created to conduct transactions with the service. Each customer has a name, address, phone number, email address, account number, password, credit card number and credit card expiry date.
- Customers make reservations with the service (ie. they purchase movie tickets). Each reservation contains an account number, a showing, the number of tickets reserved. Assume that reservation records will be retained in the database for later analysis of customer movie viewing patterns.
- Customer reviews for movies.

### Functional Requirements

The OMTS application needs to support 2 categories of functions - one for members and one for administrators.

Members:

- make an account including a login id and password
- browse movies playing at the various theatre complexes.
- purchase some number of tickets for one or more movies showing at one or more theatres
- view their purchases
- cancel a purchase
- update their personal details -- ie. modify their profile
- browse their past rentals.
- add a review for a movie.

Administrators

- list all the members
- remove a member
- add or update the information for a theatre complex/theatre
- add movies to the database
- update where/when movies are showing
- for a particular customer, show their rental history (including current tickets held)
- find the movie that is the most popular (ie. has sold the most tickets across all theatres).
- find the theatre complex that is most popular (ie. has sold the most tickets across all movies)

These requirements are a minimum. You may find that you need to add additional data and or functionality to make your application realistic, or to demonstrate that it works.

## Project Deliverables

Projects should be done in groups of 2 or 3 individuals. The groups will remain the same for all project deliverables. Group members will be assigned the same mark for each project deliverable. Sign up for groups on OnQ prior to the first deliverable. Students may choose to work alone but no compensation will be made for this when marking.

**Hand in each of the database design and final report deliverables as single pdf files using OnQ. One submission per team is sufficient.**

### Deliverable 1: Database Design and Implementation. (30% of final mark)

Design the ER schema for your database. Show all constraints and state any assumptions that you make which are not specified in the requirements. As you design your schema, keep in mind the Functional Requirements. A good design both represents the necessary data and facilitates effective use of the data. Capture as many of the constraints as possible in your design. Your ER diagram may be hand-drawn, but it must be legible.

Convert the ER schema for your application to a relational schema. Make sure you indicate the primary key and any foreign keys, as well as NOT NULL constraints, for each table. Create a database instance for your schema in MySQL and populate your database with some reasonable data. A few tuples for each table is sufficient, however you will save yourself time later if you add a reasonable amount of data at this point.

Hand in the following:

- A list of all your assumptions
- Your ER schema.
- The DDL for your relational schema.

### Deliverable 2: Application Demonstration/Final Report. (70% of final mark)

Write PHP programs that implement the functional requirements outlined above. You will need to add more data to your database to adequately demonstrate all the functionality and to make the results more interesting. You may assume that user input is correct so input syntax checking can be minimal. You must, however, handle cases where queries return no results. As the minimum, your program can simply use prompted input and formatted output from the browser. Higher marks will be given to more sophisticated Web applications.

You will be required to demonstrate your application during the final 2 weeks of class. All team members should be present and participate in the demo. A demo schedule will be established later in the term but you should plan to complete your project by the first day of scheduled demos (which will be determined later in the term once we know how many project groups there are).

For the demo you should

- Create a sample database with enough data to adequately demonstrate all of the required functionality.
- Prepare a script to follow that allows you to show some of the features of your application in a logical sequence in a 5 minute demo. You do not have to show everything, but the marker will ask to see some functionality of his/her choice.
- Be prepared to answer questions about the capabilities of your application and the development process you followed.

#### Demo Rubric:

	A+ to A-	B+ to B-	C+ to C-	D+ to D-
<b>Quality of application</b>	Application satisfies all project requirements. All functional requirements implemented correctly. Provides a superior user experience in terms of ease of use, logical flow and quality of web pages. Demonstrates excellent knowledge of course concepts and relevant technologies.	Application satisfies all project requirements. All functional requirements implemented correctly. Provides a good user experience in terms of ease of use, logical flow and quality of web pages. Demonstrates good knowledge of course concepts and relevant technologies.	Application satisfies most project requirements. Most functional requirements implemented correctly. Provides a good user experience in terms of ease of use, logical flow and quality of web pages. Demonstrates adequate knowledge of course concepts and relevant technologies.	Application satisfies some project requirements. Some functional requirements implemented correctly. Provides a poor user experience in terms of ease of use, logical flow and quality of web pages. Does not demonstrate adequate knowledge of course concepts and relevant technologies.
<b>Quality of demo</b>	Excellent organization and preparation. Shows all required functionality and features within allotted time. Excellent answers to questions. Demonstrates excellent knowledge of course concepts and relevant technologies.	Good organization and preparation. Shows all required functionality and features within allotted time. Good answers to questions. Demonstrates good knowledge of course concepts and relevant technologies.	Some organization and preparation. Shows most required functionality and features within allotted time. Adequate answers to questions. Demonstrates adequate knowledge of course concepts and relevant technologies.	Little or no organization and preparation. Shows some required functionality and features within allotted time. Adequate answers to some questions. Does not demonstrate adequate knowledge of course concepts and relevant technologies.

Hand in the following to dropbox in a single pdf file:

- List of your assumptions.
- The final versions of your ER and relational schemas.
- A discussion including the following: problems encountered during the development and how you solved them; important design and implementation decisions; the technologies and tools you used in developing your application, why you chose them and what your experience was using them; things you would like to go back and change or do differently if you had the chance.
- A user's guide to your application. This should be a document you could give to a new OMTS member or administrator to help them use your application.

#### Report Rubric

	A+ to A-	B+ to B-	C+ to C-	D+ to D-
--	----------	----------	----------	----------

<b>Project Documentation</b>	All assumptions stated. ER and relational schemas included and readable. State machine diagram included and readable. All SQL for interactions with database in each state included and sample output provided.	All assumptions stated. ER and relational schemas included and readable. State machine diagram included and readable. All SQL for interactions with database in each state included and sample output provided.	Some assumptions stated. Some of ER and relational schemas, state machine diagram and SQL for interactions with database missing.	Some assumptions stated. Some of ER and relational schemas, state machine diagram and SQL for interactions with database missing.
<b>Discussion</b>	A thorough and thoughtful discussion of all of following issues: problems encountered during the development and how you solved them; important design and implementation decisions; the technologies and tools you used; possible improvements and extensions. Shows an excellent understanding of course concepts and application development.	A thorough discussion of all of following issues: problems encountered during the development and how you solved them; important design and implementation decisions; the technologies and tools you used; possible improvements and extensions. Shows a good understanding of course concepts and application development.	A discussion of some of following issues: problems encountered during the development and how you solved them; important design and implementation decisions; the technologies and tools you used; possible improvements and extensions. Shows an adequate understanding of course concepts and application development.	No meaningful discussion provided. Does not show an adequate understanding of course concepts and application development.
<b>User Guide</b>	Clear and complete explanation of how the application can be setup and initialized for use. Clear and complete explanation of how both members and administrators can use the application and what functionality is available to both groups.	Good explanation of how the application can be setup and initialized for use. Good explanation of how both members and administrators can use the application and what functionality is available to both groups.	Some explanation of how the application can be setup and initialized for use. Some explanation of how both members and administrators can use the application and what functionality is available to both groups.	Little explanation of how the application can be setup and initialized for use. Little explanation of how both members and administrators can use the application and what functionality is available to both groups.

## Late Policy

Late submissions will not be accepted for the project deliverables for any reason.

Groups that have no members show up for their scheduled demo times will receive 0 on the demo/project. Individual group members who do not show up for the demo will receive 0 for deliverable 2 of the project. Groups will NOT be given a second chance to demo if they miss their originally scheduled date.