

**<Online Movie Ticketing System>
Test Specification**

Date of Issue: [27/02/2008]

Revision Status: **Approved**

Document Control Information

Document Author: Li Shihui, Natalie
Document Owner: Sigma Five Pte Ltd

Document Approver(s): (All Approvers are required. Records of each approver must be maintained)

Approver Name	Role
Desmond Ang	IT Manager, Cinematic Entertainment

Document Reviewers: (Records of each required reviewer must be maintained)

Reviewer Name	Role
Desmond Ang	IT Manager, Cinematic Entertainment

NOTE: All Reviewers in the list are considered Required unless explicitly listed as Optional.

Summary of Changes:

The Document Author is authorized to make the following types of changes to the document without requiring that the document be re-approved:

- Editorial, formatting, and spelling
- Clarification
- Document structure

To request a change to this document, contact the Document Author or Owner. Changes to this document are summarized in the following table in chronological order.

Revision	Date	Created by	Short Description of Changes
1.0	04/02/2008	Li Shihui, Natalie	Draft
2.0	27/01/2008	Li Shihui, Natalie	Added finalized test cases

Document Source:

The latest version of this controlled document is stored in <http://www.freewebs.com/sigmafive>

Table of Contents

1	Introduction	5
1.1	Purpose	5
1.2	Background	5
1.3	Scope	5
1.4	Project Identification	6
2	Requirements for Test	6
3	Test Strategy	25
3.1	Test Levels	25
3.2	Test Approach	29
3.2.1	Design Validation Testing	29
3.2.2	Data Validation Testing	29
3.2.3	Content Testing	30
3.2.4	Interoperability Testing	30
3.2.5	Beta Testing	31
3.2.6	Environment / System Testing	31
3.2.7	User Interface Testing	32
3.2.8	Performance Testing	32
3.2.9	Scalability Testing	34
3.2.10	Stress Testing	35
3.2.11	Regression Testing	36
3.2.12	Robustness Testing	36
3.2.13	Error Handling Testing	37
3.2.14	Security Testing	37
3.2.15	Usability Testing	38
3.2.16	User Scenarios Testing	38
3.2.17	Backup Testing	39
3.2.18	Failover / Recovery Testing	39
3.2.19	Upgrading Testing	41
3.3	Tools	42
4	Resources	42
4.1	Workers	42
4.2	System	44
5	Project Milestones	44
6	Deliverables	44
6.1	Project Tasks	44
6.2	Test Cases	45
6.3	Test Results	45

Table of Contents

Appendix A: Test Case for Add Movie

Appendix B: Test Case for Registration

Appendix C: Test Case for Booking

Appendix D: Test Results

Appendix E: Project Tasks

Test Specification

1 Introduction

1.1 Purpose

This Test Plan document for the Online Movie Ticketing System supports the following objectives:

- Identify software components that should be tested
- List the recommended requirements for test
- Describe the testing strategies to be employed
- Identify the required resources and provide an estimate of the test efforts
- List the deliverable elements of the test project

1.2 Background

The Online Movie Ticketing System allows users of the client organization to book and purchase movie tickets online. Users will be able to register themselves through the system and have access to the members-only functions. Latest movies and screening times can be updated through this system.

The Online Movie Ticketing System is a tool integrated into the internet. It consists of four major components, a Server Module, a Database Module, User Module, Ticketing Module.

The Server Module is a daemon connecting the database module to both users and ticketing module. It can receive all requests, format the pages to be displayed, validate and execute all requests.

The Database Module is MYSQL Server 5.0. It is developed as part of the OMTS system without any interface.

The User Module includes the Administrator, Registered Users and Normal Users. Registered and Normal Users can use the system to view information about particular movies available. Registered Users can update and maintain their profiles. Normal Users can register themselves through the system. Administrator can add, update and delete movie details and session times in the database.

1.3 Scope

The test scope defines what is and is not included in the testing effort for the OMTS project. Identifying the scope of testing limits the effort to a manageable size and assures testing is concentrated where it is

required.

The scope of testing for OMTS project includes:

Testing of the Online Movie Ticketing System

1.4 Project Identification

The table below identifies the documentation and availability, used for developing the test plan:

Document	Created or Available	Received or Reviewed	Author	Notes
Requirements Specification	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No		
Design Specifications	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No		
Prototype	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No		
Project Plan	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No		
Iteration Plan	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No		

2 Requirements for Test

The table below identifies the use cases, functional requirements and non functional requirements as targets for testing:

Functionality	Interface	Database	Calculus
1.1 User shall be able to load the Log in Module in the internet browser.	Online Movie Ticketing System login page	None	None
1.2 The Log in Module shall support user to log in to the system.			

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
1.2.1 The login panel shall contain a field for User ID, a field for Password, a button labeled Login and a link labeled Forget Password. The password field shall be masked with symbols and does not display what the user types	Ensure that password is visually encrypted.	None	None
1.2.2 When the Login button is clicked, the Log in Module will send the request to the Server Module to login the user.	Upon clicking the <i>login</i> button, the Log in Module validates the submitted data and sends the login request to the Server Module.	Access <i>users</i> table and verify submitted username and password.	None
1.2.3 If logging of user is successful Log in Module shall display a successful login page.	Ensure that successful login page is loaded.	None	None
1.2.3.1 Registered users who successfully logged in will be brought to the Registered User Module. (Refer to 2.1)	Upon successful login for Registered User, ensure that the Main page will be shown with the user's name and login panel will be invisible.	Access <i>Registered Users</i> table to retrieve user's name.	None
1.2.3.2 Administrators who successfully logged in will be brought to the Administrator Module. (Refer to 3.1)	Upon successful login for Administrator, ensure that the Assign User Role page will be shown with the administrator's name and the login panel will be invisible.	Access <i>Administrators</i> table to retrieve administrator's name.	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
1.2.3.3 Ticketing officers who successfully logged in will be brought to the Ticket Collection Module. (Refer to 4.1)	Upon successful login for Ticketing Officer, ensure that the Change Ticket Status page will be shown with the officer's name and the login panel will be invisible.	Access <i>Ticketing Officer's</i> table to retrieve officer's name.	None
1.2.4 Unsuccessful login will display an error message. (Refer to 1.2)	Upon entering the incorrect details, ensure that an 'incorrect credentials' message will be shown and all text fields will be cleared.	None	None
1.2.5 When the Forget Password link is clicked, the Log in Module will redirect the user to the Forget Password page	Upon clicking the Forget Password link, the Log in Module will redirect the user to the Forget Password page	None	None
1.2.5.1 The forget password panel shall contain a field for User ID, a field for Email address and a button labeled Send.	Forget Password page	None	None
1.2.5.2 When the Send button is clicked, the Log in Module will send the request to the Server Module to send password to user	Upon clicking the Send button, the Log in Module validates the submitted data and sends the send request to the Server Module	Access <i>users</i> table to validate User ID and retrieve password.	Validate that email address should be in this format – xxx@xxx.xxx

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
1.2.6 If validation of email address and User ID is successful, Server Module shall send retrieved password to user's email address. (Refer to 1.2.5.2)	Upon successful validation of email address and User ID, a sent notification is shown.	None	None
1.2.7 If validation of either User ID or email address is unsuccessful, an error message is shown.	Upon entering the incorrect details, an 'incorrect credentials' message will be shown and all text fields will be cleared.	None	None

Functionality	Interface	Database	Calculus
2.1 Upon successful log in, Registered users shall be able to continue navigating the website and view currently showing and upcoming movie listings. (Refer to 1.2.3.1)	Upon successful login, ensure that <i>Main</i> page is shown with Registered User's name and <i>Login</i> panel is invisible.	Access <i>Registered User's</i> table to retrieve user's name.	None
2.2 Registered User's navigation panel contains links labeling <i>Home</i> , <i>Now Showing</i> , <i>Coming Soon</i> , <i>Book Ticket</i> , <i>View/Edit Profile</i> , <i>Virtual Banking</i> , <i>Log Out</i> and <i>Contact Us</i> .	Ensure that all the links link to the correct page.	None	None
2.2.1 When the <i>Home</i> link is clicked, main page will be displayed.	Upon clicking the <i>Home</i> link, ensure that the <i>Main</i> page for Registered User is displayed.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
2.2.2 When the <i>Now Showing</i> link is clicked, <i>Now Showing</i> page will be displayed.	Upon clicking the <i>Now Showing</i> link, ensure that the <i>Now Showing</i> page is displayed.	None	None
2.2.2.1 The <i>Now Showing</i> panel contains a button labeled <i>Book</i> under each now showing movies and details of all the <i>Now Showing</i> movies.	Upon clicking the <i>Book</i> button, ensure that Registered users will be redirected to the <i>Movie Selection</i> page. (Refer to 7.1)	None	None
2.2.3 When the <i>Coming Soon</i> link is clicked, <i>Coming Soon</i> page will be displayed.	Upon clicking the <i>Coming Soon</i> link, ensure that the <i>Coming Soon</i> page will be display.	None	None
2.2.3.1 The <i>Coming Soon</i> panel displays the details (title, genre, rating, runtime, synopsis) of all the upcoming movies.	Ensure that all details are displayed correctly	Access <i>Coming Soon Movie</i> table to retrieve all the information to be displayed.	None
2.2.4 When the <i>Book Ticket</i> link is clicked, <i>Movie Selection</i> page will be displayed. (Refer to 7.1)	Upon clicking the <i>Book Ticket</i> link, ensure that Registered users will be redirected to the <i>Movie Selection</i> page.	None	None
2.2.5 When the <i>Virtual Banking</i> link is clicked, <i>Virtual Banking</i> page will be displayed. (Refer to 6.1)	Upon clicking the <i>Virtual Banking</i> link, ensure that Registered users will be redirected to the <i>Virtual Banking</i> page.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
2.2.6 When the <i>Log Out</i> link is clicked, users shall be redirected to the <i>Main</i> page.	Ensure that users are redirected to the <i>Main</i> page and sessions will be cleared and <i>Login</i> panel will be visible.	None	None
2.2.7 When the <i>View/Edit Profile</i> link is clicked, <i>View/Edit Profile</i> page will be displayed.	Upon clicking the <i>View/Edit Profile</i> link, ensure that Registered users will be redirected to the <i>Edit Profile</i> page.	None	None
2.2.7.1 Registered users <i>View/Edit Profile</i> panel shall contain fields for <i>User ID, Name, Address, Contact number, Email address, Old password, New password, Confirm password</i> , a button labeled <i>Cancel</i> , a button labeled <i>Update</i> and a button labeled <i>Edit</i> .	Ensure that all fields are disabled.	Access <i>Registered User's</i> table to retrieve user's information.	None
2.2.7.2 When the <i>Edit</i> button is clicked, fields will be enabled for editing.	Upon clicking the <i>Edit</i> button, ensure that all fields are enabled for editing except for the <i>User ID</i> field.	None	None
2.2.7.3 When the <i>Update</i> button is clicked, the Registered User module will send request to the Server Module to update the user's profile.	Upon clicking the <i>Update</i> button, ensure that the submitted data is validated and updated in the database.	Access the Registered User's table to verify old password.	Validate that password length should be 8 characters long. Email address should be in this format – xxx@xxx.xxx

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
2.2.8 If validation is unsuccessful, an error message is shown.	Ensure that 'incorrect credentials' message will be shown and fields cleared when the validation is unsuccessful.	None	None
2.2.9 If validation is successful, Server Module will update Database and a successful update notification message is shown.	Upon successful update, ensure that the submitted data is updated in the database and a successful update notification message is shown.	Access the <i>Registered User's</i> table and update profile.	None

Functionality	Interface	Database	Calculus
3.1 Upon successful login, system shall display administrator <i>Assign User Role</i> page. (Refer to 1.2.3.2)	Ensure that administrator's <i>Assign User Role</i> page is shown with Administrator's name and <i>Login</i> panel is invisible.	Access <i>Administrator's</i> table to retrieve user's name.	None
3.2 Administrator navigation panel contains links labeling <i>Assign User Role, Add Movie, Modify Records, Arrange Session, Promotion Details, Change Password</i> and <i>Log Out</i>	Ensure that all the links link to the correct page.	None	None
3.2.1 When <i>Assign User Role</i> link is clicked, <i>Assign User Role</i> page will be displayed.	Upon clicking the <i>Assign User Role</i> link, ensure that <i>Assign User Role</i> page will be displayed.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
3.2.1.1 The <i>Assign User Role</i> panel contains a dropdown list for <i>User Role</i> , fields for <i>User ID</i> , <i>First Name</i> , <i>Last Name</i> , <i>Password</i> , <i>Contact Number</i> , <i>Email</i> and buttons labeled <i>Add</i> and <i>Reset</i> .	Ensure that the data contained in the dropdown list is correct.	None	None
3.2.1.2 When the <i>Add</i> button is clicked, Administrator Module will send request to the Server Module to update the user's role.	Upon clicking the <i>Add</i> button, ensure that submitted data validated and added into the database.	None	Validate that password length should be 8 characters long. Email address should be in this format – xxx@xxx.xxx
3.2.1.3 If validation is successful, Server Module will add user's role to the Database and a successful update notification message is shown.	Upon successful update, ensure that the submitted data is added in the database and a successful update notification message is shown.	Access the <i>User's</i> table and update profile.	None
3.2.1.4 If validation is unsuccessful, an 'incorrect credential' is shown and all fields will be cleared.	Ensure that all fields are cleared and error message is shown.	None	None
3.2.1.5 When the <i>Reset</i> button is clicked, all fields will be cleared.	Upon clicking the <i>Reset</i> button, ensure all fields in the <i>Assign User Role</i> panel will be cleared.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
3.2.2 When <i>Add Movie</i> link is clicked, <i>Add Movie</i> page will be displayed.	Upon clicking the <i>Add Movie</i> link, Administrator Module will display the <i>Add Movie</i> page.	None	None
3.2.2.1 The <i>Add Movie</i> panel shall contain fields for <i>Movie ID, Movie Title, Runtime, Cost, Start and End Date, synopsis, dropdown list for Rating, Genre, Status</i> , a button labeled <i>Browse</i> , a button labeled <i>Add Movie</i> and a button labeled <i>Clear</i> .	Ensure that all details in the dropdown lists are correct.	None	None
3.2.2.2 When the <i>Browse</i> button is clicked, Administrator Module will allow Administrator to search for movie <i>Image</i> .	Upon clicking the <i>Browse</i> button, ensure Administrator will be able to search for movie image to be uploaded to the image box.	None	None
3.2.2.3 When the <i>Add</i> button is clicked, Administrator Module will send request to the Server Module to add the new movie details.	Upon clicking the <i>Add</i> button, ensure that submitted data is validated and request is added into the movie database.	None	Date should be in dd/mm/yy format.
3.2.2.4 If validation is unsuccessful, an error message will be shown and fields will be cleared.	Ensure that error message is shown and fields cleared upon unsuccessful validation.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
3.2.2.5 If validation is successful, Server Module will add movie into the database and successful add notification message is shown.	Ensure that submitted data is added into the database and successful add notification message is shown.	Access <i>Movie</i> 's table and add movie details.	None
3.2.2.6 When the <i>Reset</i> button is clicked, all fields will be cleared.	Upon clicking the <i>Reset</i> button, ensure that all fields in the <i>Add Movie</i> panel will be cleared.	None	None
3.2.3 When the <i>Modify Records</i> link is clicked, <i>Modify Records</i> page will be displayed.	Upon clicking the <i>Modify Records</i> link, ensure that <i>Modify Records</i> page will be shown.	None	None
3.2.4 When the <i>Arrange Session</i> link is clicked, <i>Arrange Session</i> page will be displayed.	Upon clicking the <i>Arrange Session</i> link, Administrator Module will display the <i>Arrange Session</i> page.	None	None
3.2.4.1 The <i>Arrange Session</i> panel shall contain cinema number, dropdown lists for <i>Sessions and Movies</i> , links labeled <i>Update</i> for individual cinema.	Ensure that every column's title is appropriate to the data contained within that column.	None	None
3.2.4.2 When the <i>Update</i> link is clicked, Administrator Module will send request to the Server Module to update the session times.	Ensure that Sessions submitted is updated upon clicking the <i>Update</i> link.	Access the <i>Cinema</i> table to update the <i>Session time</i> .	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
3.2.5 When the <i>Promotion Details</i> link is clicked, <i>Promotion Details</i> page will be displayed.	Upon clicking the <i>Promotion Details</i> link, ensure that <i>Promotion Details</i> page is displayed.	None	None
3.2.5.1 The <i>Promotion Details</i> panel shall contain fields for <i>Enter New Promotion Code</i> , <i>Promotion Details</i> , <i>Promotion Details</i> and buttons labeled <i>Update</i> and <i>Clear</i> .	Ensure that details are entered correctly.	None	None
3.2.5.2 When the <i>Update</i> button is clicked, Administrator Module shall send update request to Server Module to update the promotion details.	Upon clicking the <i>Update</i> button, ensure that submitted data is validated and details are updated into the promotion detail database.	Access <i>Promotion Detail</i> database to update details.	None
3.2.5.3 When the <i>Clear</i> button is clicked, all fields in the panel shall be cleared.	Upon clicking the <i>Clear</i> button, ensure that all fields in the panel should be cleared.	None	None
3.2.6 When the <i>Change Password</i> link is clicked, <i>Change Password</i> page shall be displayed.	Upon clicking the <i>Change Password</i> link, ensure that the <i>Change Password</i> page is displayed.	None	None
3.2.6.1 The <i>Change Password</i> panel shall contain fields for <i>Enter old Password</i> , <i>Enter New Password</i> , <i>Confirm New Password</i> and buttons labeled <i>Change</i> and <i>Reset</i> .	Ensure that the password is visually encrypted.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
3.2.6.2 When the <i>Change</i> button is clicked, Administrator Module shall send change request to Server Module to change the password.	Upon clicking the <i>Change</i> button, ensure that submitted data is validated and password is changed.	None	Validate that password length should be 8 characters long.
3.2.6.3 If validation is unsuccessful, an error message shall be shown and fields shall be cleared.	Ensure that error message is shown and fields cleared upon unsuccessful validation.	None	None
3.2.6.4 If validation is successful, a successful changed message shall be shown and fields shall be cleared.	Ensure that submitted data is changed in to database and successful changed message is shown.	Access <i>User</i> table to change user's password.	None
2.2.6 When the <i>Log Out</i> link is clicked, users shall be redirected to the <i>Main</i> page.	Ensure that users are redirected to the <i>Main</i> page and sessions will be cleared and <i>Login</i> panel will be visible.	None	None

Functionality	Interface	Database	Calculus
4.1 Upon successful login, ticketing officer shall be able to view the <i>Update Ticket Status</i> page. (Refer to 1.2.3.3)	Ensure that <i>Update Ticket Status</i> page is displayed.	None	None
4.2 The <i>Update Ticket Status</i> panel shall contain field for <i>Confirmation ID</i> , dropdown list for <i>Ticket Status</i> and button labeled <i>Update</i> .	Ensure that data contained in the dropdown list is correct.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
4.2.1 Ticketing officer shall be able to issue tickets to users and update status of tickets as paid, booked, collected or cancelled.	Ensure that status selected is correct.	None	None
4.2.2 When the <i>Update</i> button is clicked, Ticket Collection Module shall send update request to the Server Module to update database.	Ensure that submitted data is updated into the database.	Access database to update status.	None

Functionality	Interface	Database	Calculus
5.1 Users who visit the Online Movie Ticketing System but have not register, are able to navigate through the website.	Ensure that Normal users are not allowed to use functions for the Registered Users.	None	None
5.2 Registered User's navigation panel contains links labeling <i>Home, Now Showing, Coming Soon, Book Ticket, View/Edit Profile, Virtual Banking and Contact Us.</i>	Ensure that all links link to the correct page.	None	None
5.2.1 When the <i>Home</i> link is clicked, main page will be displayed.	Upon clicking the <i>Home</i> link, ensure that the <i>Main</i> page is displayed.	None	None
5.2.2 When the <i>Now Showing</i> link is clicked, Now Showing page will be displayed.	Upon clicking the <i>Now Showing</i> link, ensure that the <i>Now Showing</i> page is displayed.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
5.2.2.1 The Now Showing panel contains a button labeled <i>Book</i> under each now showing movies and details of all the Now Showing movies.	Upon clicking the <i>Book</i> button, ensure that users will be redirected to the <i>Movie Selection</i> page. (Refer to 7.1)	None	None
5.2.3 When the <i>Coming Soon</i> link is clicked, Coming Soon page will be displayed.	Upon clicking the <i>Coming Soon</i> link, ensure that the <i>Coming Soon</i> page will be display.	None	None
5.2.3.1 The <i>Coming Soon</i> panel displays the details (title, genre, rating, runtime, and synopsis) of all the upcoming movies.	Ensure that all details are displayed correctly	Access <i>Movie</i> table to retrieve all the information to be displayed.	None
5.2.4 When the <i>Book Ticket</i> link is clicked, <i>Movie Selection</i> page will be displayed. (Refer to 7.1)	Upon clicking the <i>Book Ticket</i> link, ensure that users will be redirected to the <i>Movie Selection</i> page.	None	None
5.3 Users shall be able to view movies and their screening times by entering the <i>Movie Selection</i> page. (Refer to 7.1)	Ensure that users are redirected to the <i>Movie Selection</i> page when they want to book tickets.	None	None
5.4 Users shall be able to register themselves as registered users, by clicking on the register now button.	Ensure that users are redirected to the <i>Registration</i> page when they click on the <i>Register Now</i> button.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
<p>5.4.1 The <i>Registration</i> panel contains of fields for <i>First Name, Last Name, User ID, NRiC, Password, Confirm Password, Contact Number, Email, Confirm Email, Address</i>, a button labeled <i>Submit</i> and a button labeled <i>Reset</i>.</p>	<p>Ensure that password and email entered in the <i>Confirm Password</i> and <i>Email Address</i> field should match the password and email in the <i>Password</i> and <i>Email Address</i> field.</p>	<p>Access user's table to validate if <i>User ID</i> is available.</p>	<p>Validate that password length should be 8 characters long. Email should be in this format – xxx@xxx.xxx</p>
<p>5.4.2 When <i>Submit</i> button is clicked, Normal Users Module will send a request to the Server Module to register user.</p>	<p>Upon clicking the <i>Submit</i> button, the Normal User Module will validate the submitted data and sends a register request to the Server Module to register the user.</p>	<p>None</p>	<p>None</p>
<p>5.4.3 When <i>Reset</i> button is clicked, all fields in the <i>Registration</i> panel shall be cleared.</p>	<p>Upon clicking the <i>Reset</i> button, ensure that all fields in the <i>Registration</i> panel are cleared.</p>	<p>None</p>	<p>None</p>
<p>5.5 If submission is successful, a successful submission message shall be shown and text fields shall be cleared.</p>	<p>Ensure that 'Your registration is successful' message is shown below the table and text fields are cleared.</p>	<p>None</p>	<p>None</p>
<p>5.6 If submission is unsuccessful, error message shall be shown.</p>	<p>Ensure that an error message is shown below the table.</p>	<p>None</p>	<p>None</p>

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
6.1 Registered users shall be able to use the Virtual Bank Module.	Virtual Banking page will be displayed.	None	None
6.2 The <i>Virtual Banking</i> panel shall contain fields for <i>Reference ID, Name, Email Address, Balance</i> , a dropdown list for <i>Deposit</i> and a button labeled <i>Confirm</i> .	Ensure that all fields are disabled.	Access Registered User's table to retrieve user's information.	None
6.2.1 Registered users shall be able to deposit money which can be used to pay for the tickets when they want to use it.	Ensure that users are able to select amount they want to deposit from the <i>Deposit</i> dropdown list.	None	None
6.2.2 When <i>Confirm</i> button is clicked, balance shall be update and information message shall be displayed.	Ensure that balance is updated and information message is displayed below the table.	None	None
6.3 If unsuccessful, an error message shall be displayed.	Ensure that an error message is displayed below the table.	None	None
6.4 If successful, message information shall be displayed and balance shall be updated.	Ensure that message information is shown below the table and the balance is updated.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
7.1 Both registered users and normal users shall be redirected to the <i>Movie Selection</i> page when they choose to book tickets.	Ensure that <i>Movie Selection</i> page are displayed when users choose to book tickets.	None	None
7.2 The <i>Movie Selection</i> panel shall contain dropdown lists for <i>Select Cinema</i> or <i>Select Movie</i> , <i>Select Day</i> , <i>Select Time</i> and a button labeled <i>Next</i> .	Ensure that all dropdown lists contain the correct data.	None	None
7.2.1 When the <i>Next</i> button is clicked, users shall be redirected to the <i>Seat Selection</i> page.	Ensure that users are redirected to the <i>Seat Selection</i> page upon clicking the <i>Next</i> button.	None	None
7.2.2 The <i>Seat Selection</i> panel shall contain buttons labeled with seats number and <i>Confirm</i> .	Ensure that not more than 5 seats can be selected.	None	None
7.2.2.1 When the <i>Confirm</i> button is clicked, users shall be redirected to the <i>Ticket Order Details</i> page.	Ensure that users are redirected to the <i>Ticket Order Details</i> page.	None	None
7.2.2.4 When users click on the <i>Confirm</i> button, Book Ticket Module shall redirect users to <i>Ticket Order Details</i> page. (Refer to 8.1)	Ensure that users are redirected to the <i>Ticket Order Details</i> page upon clicking the <i>Confirm</i> button.	None	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
8.1 Upon confirmation in the <i>Movie and Seat Selection</i> page, users shall be redirected to <i>Ticket Order Details</i> page.	Ensure that <i>Ticket Order Details</i> page is displayed.	None	None
8.1.1 For Normal User, <i>Ticket Order Details</i> panel shall contain the details of tickets ordered and fields for <i>Name, NRIC/Password, Contact Number, Email</i> , radio buttons for payment labeled <i>Credit Card, Cash</i> and buttons labeled <i>Confirm</i> and <i>Cancel</i> .	Ensure the correct details of tickets ordered are shown correctly.	None	None
8.1.1.1 Normal User shall enter their particulars and select payment mode, credit card or cash.	Ensure that upon selecting credit card radio button, panel with containing fields for credit card number and expiry date appears.	None	None
8.1.1.2 Upon clicking the <i>Confirm</i> button, user shall be redirected to the payment page. (Refer to	Ensure that upon selecting the <i>Confirm</i> button, <i>Payment</i> page is shown	None	None
8.1.2 For Registered User, <i>Ticket Order Details</i> panel shall contain details of tickets ordered and fields for <i>Name, NRIC/Password, Contact Number, Email</i> , radio buttons for payment labeled <i>Credit Card, Cash</i> and buttons labeled <i>Confirm</i> and <i>Cancel</i> .	Ensure the correct details of tickets ordered are shown correctly. Ensure that user's particulars are shown correctly and all fields are disabled.	Access <i>User</i> table to retrieve user's particulars.	None

<Online Movie Ticketing System> : Test Specification

Functionality	Interface	Database	Calculus
8.1.2.1 Registered User shall select payment mode, credit card, cash or virtual bank.	Ensure that upon selecting credit card radio button, panel with containing fields for credit card number and expiry date appears.	None	None
8.1.2.2 Registered users shall be able to key in and verify promotion code.	Ensure that the correct code is entered.	Access <i>Promotion</i> table to verify code.	None
8.1.2.3 Upon clicking the <i>Confirm</i> button, user shall be redirected to the <i>Confirmation</i> page. (Refer to 8.2)	Ensure that upon selecting the <i>Confirm</i> button, <i>Confirmation</i> page is shown.	None	None
8.2 Both Registered and Normal users shall be able to print the confirmation page and also exit to the main page of the system.	Ensure that details shown in the <i>Confirmation</i> page is correct and users are able to print the details and exit to the <i>Main</i> page.	None	None

Functionality	Test to execute
9.1 Server Module shall be between various modules and the Database.	No test
9.2 Server Module shall validate and execute all requests from the other modules.	For all functionalities specified for validation and verification, Server Module will execute all requests. The Server Module returns the requested action, interacts with the database and returns all database action details, then get results and return results.

3 Test Strategy

3.1 Test Levels

The Online Movie Ticketing System adopts an evolutionary approach of testing components at different levels as they are assembled into larger subsystems. Five different levels of testing, Static, Unit, Integration, System and Acceptance, are applicable in this project.

3.1.1 Static Testing

Static testing, also known as the dry run testing, is the review, inspection and validation of development requirements. It is done in the early stage of the project before the actual program and application is used. Static testing is the most effective and cost effective way of testing. It helps to prevent problems from occurring by reviewing project work products across every stage in the development cycle.

Static testing will be conducted throughout the project

The list below identifies the type of test considered during Static Testing for Online Movie Ticketing System:

- Review of Software Requirements Specification to ensure that all the documented requirements are complete and can be tested
- Review of Software Design Specification to ensure that the design of the system meets the requirements
- User interface
- Verify code against design requirements
- Verify error handling processing
- Review of test cases against design specification
- Selective inspection of test cases and verify that test cases are completed

3.1.2 Unit Testing

Unit testing is the first dynamic level of test performed on the smallest parts of an application. It isolates each part of the program ensuring that individual modules are working properly. Unit testing does not only test the functionality of the code but also ensure that the code is able to respond appropriately.

<Online Movie Ticketing System> : Test Specification

The list below identifies the type of test considered during Unit Testing for Online Movie Ticketing System:

- Data validation
- Scalability
- Regression
- Error handling
- Security

The table below identifies the individual units for the Online Movie Ticketing System to be tested during different iterations:

Iteration	Functions	Units to be tested
1	Administrative	<ul style="list-style-type: none">• Coding of Administrative subsystem• Coding of Movie Database
2	Ticket Sales	<ul style="list-style-type: none">• Coding of Booking subsystem• Coding of Payment subsystem• Coding of Collection subsystem
3	Registration	<ul style="list-style-type: none">• Coding of Registration and Profile Subsystem• Coding of Virtual Bank and Promotion subsystem

3.1.3 Integration Testing

Integration testing is a logical extension of unit testing where individual unit tested software modules are combined into a component and tested as a group. It will identify all problems that occur when units are combined and tested.

We will be using Bottom-up integration testing for this Online Movie Ticketing System project as we begin with unit testing before moving on to integration testing.

The list below identifies the type of test considered during Static Testing for online Movie Ticketing System:

- Data validation
- Scalability
- Regression
- Error handling
- Security
- Usability

Integrated functions that will be tested in this Online Movie Ticket System are identified below:

- Integration of Ticket Sales and Administrative functions
- Integration of entire system

3.1.4 System Testing

System testing ensures that the each functions of the system works as expected and errors are noted and analyzed. It is performed against the Software Requirements Specification focusing on both design and behavior of the system.

The list below identifies the types of test considered during System Testing:

- User Interface Testing
- Data validation
- Content testing
- Interoperability
- Environment/System
- Performance and Capacity
- Stress and Volume
- Regression
- Robustness

- Error handling
- Security
- Usability
- Users scenarios
- Operational issues

3.1.5 Acceptance Testing

Acceptance testing is a black box test conducted by end user or client without having them to understand how the internal system behaves. The objective of this test verifies the entire functionality of the system from the end user perspectives. Acceptance testing represents the interests of the end users and give them confidence that the application is functioning correctly and has all the required functions.

User Acceptance Testing, done based on the Software Requirement Specification, will be performed in a tightly controlled production-like environment, focusing on positive testing. It is the last stage of testing run by customers to demonstrate the functionality.

The list below identifies the types of tests considered during Acceptance Testing for the Online Movie Ticketing System:

- Error handling
- Beta testing
- Operational issues
- Design validation
- Data validation
- Usability
- Regression
- Security
- User interface

3.2 Test Approach

3.2.1 Design Validation Testing

Test Objective	Ensure that the design of the system meet the requirements (if any) and is to the stakeholders approval
Technique	<ul style="list-style-type: none"> • Formal/Informal meeting
Completion Criteria	<ul style="list-style-type: none"> • Stakeholders approval needed
Special Considerations	<ul style="list-style-type: none"> • Testing might not be required

3.2.2 Data Validation Testing

Test Objective	<ul style="list-style-type: none"> • Ensure methods and functions access the database processed data properly, without corrupting the data
Technique	<ul style="list-style-type: none"> • Invoke each database access method and process, seeding each with valid and invalid data (or requests for data) • Inspect the database to ensure the data has been populated as intended, all database events executed properly and review returned data to ensure that correct data was retrieved
Completion Criteria	<ul style="list-style-type: none"> • Data in the database are not corrupted
Special Considerations	<ul style="list-style-type: none"> • Testing may require a Database Management development environment or drivers to enter or modify data directly in the databases • Processes should be invoked manually • Small or minimized databases (limited number of records) should be used to increase the visibility of any non-acceptable events

3.2.3 Content Testing

Test Objective	<ul style="list-style-type: none">• Ensure every function returns the correct result when called upon• Verify that the database/application validates the data to be stored in the system are of correct format
Technique	<ul style="list-style-type: none">• Use the test procedure from the User Scenario Testing to ensure content retrieved from the database is correct• Programming Lead to write codes to retrieve the data stored to manually check the data format
Completion Criteria:	<ul style="list-style-type: none">• Content displayed on the system should tally with the data stored in the database
Special Considerations	<ul style="list-style-type: none">• Need to be done by Database Lead and Programming Lead as it involves coding

3.2.4 Interoperability Testing

Test Objective	<ul style="list-style-type: none">• Ensure the application can be used on any system.
Technique	<ul style="list-style-type: none">• Use the test procedures in User Scenario Testing to test on the following :<ul style="list-style-type: none">○ machines with different operating systems○ machines with similar operating system but different browsers
Completion Criteria	<ul style="list-style-type: none">• Results/Display for the application on all sources are the same
Special Considerations	<ul style="list-style-type: none">• Systems/Browsers might not be available (have to purchase)

3.2.5 Beta Testing

Test Objective	<ul style="list-style-type: none">• Verify and validate that the system meets business functional requirements• Ensure that application faults, failures and defects have been eliminated
Technique	<ul style="list-style-type: none">• Use the test procedures in the User Scenario Testing
Completion Criteria	<ul style="list-style-type: none">• All functional requirements are working• No error or failed functions encountered
Special Considerations	<ul style="list-style-type: none">• Need to simulate the environment that client wants the application to be deployed (large scaled)• Need to test on user's actual environment

3.2.6 Environment / System Testing

Test Objective	<ul style="list-style-type: none">• Verify and validate that the system meets business functional requirements• Detect application faults, failures and defects
Technique	<ul style="list-style-type: none">• Use the test procedures in the User Scenario Testing
Completion Criteria	<ul style="list-style-type: none">• All functional requirements are working• No error or failed functions encountered
Special Considerations	<ul style="list-style-type: none">• Need to simulate the environment that client wants the application to be deployed (small scaled)

3.2.7 User Interface Testing

Test Objective	<ul style="list-style-type: none">• Verify that navigation through the application properly reflects the business functions and requirements, including windows to windows, fields to fields and use of access methods (tab keys, mouse movements)• Verify that window objects and characteristics, such as menus, types, size, position, state, and focus conform to standards and user's requirements
Technique	<ul style="list-style-type: none">• Create/Modify tests for each window to verify proper navigation and object states for each application window and objects
Completion Criteria	<ul style="list-style-type: none">• Each window successfully verified to remain consistent with benchmark version or within acceptable standard
Special Considerations	<ul style="list-style-type: none">• Not all properties for custom and third party objects can be accessed

3.2.8 Performance Testing

Test Objective	<ul style="list-style-type: none">• Measure and evaluate performance behaviors for designated transactions or business functions under the following conditions:<ul style="list-style-type: none">○ normal anticipated workload○ worse case anticipated workload• Tune performance behaviors to meet standard benchmark or user's requirements
-----------------------	--

<Online Movie Ticketing System> : Test Specification

Technique	<ul style="list-style-type: none">• Use Test Procedures developed for User Scenarios Testing• Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs• Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients, virtual or actual (see special considerations below)
Completion Criteria	<ul style="list-style-type: none">• Single transaction (single user)<ul style="list-style-type: none">○ Successful completion of the test scripts without any failures and within the expected / required time allocation (per transaction)• Multiple transactions (multiple users)<ul style="list-style-type: none">○ Successful completion of the test scripts without any failures and within acceptable time allocation

Special Considerations	<ul style="list-style-type: none"> • Comprehensive performance testing includes having a “background” workload on the server • There are several methods that can be used to perform this, including: <ul style="list-style-type: none"> ○ “Drive transactions” directly to the server, usually in the form of SQL calls ○ Create “virtual” user load to simulate many (usually several hundred) clients. This technique can also be used to load the network with “traffic” ○ Use multiple physical clients, each running test scripts to place a load on the system • Performance testing should be performed on a dedicated machine or at a dedicated time as this permits full control and accurate measurement • The databases used for Performance testing should be either actual size, or scaled equally
-------------------------------	--

3.2.9 Scalability Testing

Test Objective	<ul style="list-style-type: none"> • Prove that both the functionality and the performance of a system will scale up to meet specified requirements
Technique	<ul style="list-style-type: none"> • Increase the minimum hardware resources that the machine currently sits on, or • Find a machine with a higher specification
Completion Criteria	<ul style="list-style-type: none"> • Stress Testing fails
Special Considerations	<ul style="list-style-type: none"> • Is budget allocated for scalability? • Scale up now or later?

3.2.10 Stress Testing

<p>Test Objective</p>	<ul style="list-style-type: none"> • Verify that main required functions perform properly without error under the following stress conditions: <ul style="list-style-type: none"> ○ Minimum hardware resource available (memory, CPU speed, hard disk size) ○ Maximum (actual or physically capable) number of clients connected (or simulated) ○ Multiple users performing the same transactions against the same data / accounts ○ Worse case anticipated transaction volume / mix (see performance testing above) <p>NOTE:</p> <p>The goal of Stress test might also be stated as identify and document the conditions under which the system FAILS to continue functioning properly.</p> <p>Stress testing of the client is described under section 3.1.10, Configuration testing.</p>
<p>Technique</p>	<ul style="list-style-type: none"> • Use tests developed for Performance Testing or Capacity Testing • Test limited resources, tests will be run on a single machine, with the minimum hardware resources available • For the remaining stress tests, multiple clients will be used, either running the same tests or complementary tests to produce the worst case transaction volume/mix
<p>Completion Criteria</p>	<ul style="list-style-type: none"> • All planned tests are executed and specified system limits are reached / exceeded without the software or software failing

<Online Movie Ticketing System> : Test Specification

Special Considerations	<ul style="list-style-type: none"> • Stressing the network may require network tools to load the network with messages/packets • Synchronization of simultaneous clients accessing the same records / data accounts
-------------------------------	---

3.2.11 Regression Testing

Test Objective	<ul style="list-style-type: none"> • Ensure that applied changes to the application have not adversely affected previously tested functionality
Technique	<ul style="list-style-type: none"> • Use the test procedures created for Data Validation Testing
Completion Criteria	<ul style="list-style-type: none"> • Results / deliverables from running the test have to be similar to the results produced in Data Validation Testing
Special Considerations	

3.2.12 Robustness Testing

Test Objective	<ul style="list-style-type: none"> • Develop test environments to assess the degree to which a system operates correctly in the presence of exceptional inputs or stressful environmental conditions
Technique	<ul style="list-style-type: none"> • Based on Performance Testing and Stress Testing
Completion Criteria	<ul style="list-style-type: none"> • Performance Testing and Stress Testing return PASS results
Special Considerations	

3.2.13 Error Handling Testing

Test Objective	<ul style="list-style-type: none"> • Ensure that the system is able to handle errors correctly
Technique	<ul style="list-style-type: none"> • Create a set of test procedures to test every field and function with the wrong set of format of data
Completion Criteria	<ul style="list-style-type: none"> • Error messages/notifications are shown
Special Considerations	<ul style="list-style-type: none"> • Not all fields have to be checked

3.2.14 Security Testing

Test Objective	<ul style="list-style-type: none"> • Application-level Security <ul style="list-style-type: none"> ○ Verify that an actor can access only those functions/data for which permission is provided for their user type • System-level Security <ul style="list-style-type: none"> ○ Verify that only those actors with access to the system and application(s) are permitted to access them
Technique	<ul style="list-style-type: none"> • Application-level <ul style="list-style-type: none"> ○ Identify and list each actor type and the functions/data each type has permissions for • Create tests for each actor type and verify permission by creating transactions specific to each user actor • Modify user type and re-run tests for same users and verify that additional functions/data in each case are correctly available or denied • System-level Access (see special considerations below)

<Online Movie Ticketing System> : Test Specification

Completion Criteria	<ul style="list-style-type: none">For each known actor type, the appropriate functions/data are available and all transactions function as expected and run in prior to the function tests
Special Considerations	<ul style="list-style-type: none">Access to the system must be reviewed/discussed with the appropriate network or systems administrator

3.2.15 Usability Testing

Test Objective	<ul style="list-style-type: none">Verify that functions being re-used returns the expected resultsEvaluate if the re-used functions need to be modified to produce the required results
Technique	<ul style="list-style-type: none">Create a test to ensure that the functions being reused returns the expected results
Completion Criteria	<ul style="list-style-type: none">Re-used functions produce the expected results
Special Considerations	

3.2.16 User Scenarios Testing

Test Objective	<ul style="list-style-type: none">Ensure that all or most requirements are available in the systemFind defects in the system
Technique	<ul style="list-style-type: none">Create all the possible test cases (scenarios) that users will use the systemEngage internal or external users to test the system by following the test case
Completion Criteria	<ul style="list-style-type: none">Users are able to fulfill all the steps stated in the test case

Special Considerations	<ul style="list-style-type: none"> • Technical ability of users
-------------------------------	--

3.2.17 Backup Testing

Test Objective	<ul style="list-style-type: none"> • Ensure backup of system/user's data are correctly and safely done
Technique	<ul style="list-style-type: none"> • Create a set of procedures to backup data. • Database Lead needs to create/vet the backup procedure and the backup data
Completion Criteria	<ul style="list-style-type: none"> • No error in backing up data
Special Considerations	<ul style="list-style-type: none"> • Medium to store the data • Backup can only be done on a specific period (need notification to users)

3.2.18 Failover / Recovery Testing

Test Objective	<ul style="list-style-type: none"> • Verify that recovery processes (manual or automated) properly restore the database, applications and system to a desired and known state. • The following types of conditions are to be included in the testing: <ul style="list-style-type: none"> ○ Power interruption to the client ○ Power interruption to the server ○ Communication interruption via network server(s) ○ Incomplete cycles (data filter processes interrupted, data synchronization processes interrupted) ○ Invalid database pointer/keys ○ Invalid/corrupted data element in
-----------------------	--

<Online Movie Ticketing System> : Test Specification

	<p align="center">database</p>
<p align="center">Technique</p>	<ul style="list-style-type: none"> • Tests created for Function and Business Cycle testing should be used to create a series of transactions. • Once the desired starting test point is reached, the following actions should be performed (or simulated) individually: <ul style="list-style-type: none"> ○ Power interruption to the client <ul style="list-style-type: none"> • Power the PC down ○ Power interruption to the server <ul style="list-style-type: none"> • Simulate or initiate power down procedures for the server ○ Interruption via network servers <ul style="list-style-type: none"> • Simulate or initiate communication losses with the network (physically disconnect communication wires or power down network server(s) /router(s)) • Testing for incomplete cycles utilizes the same technique as described above except that the database processes themselves should be aborted or prematurely terminated • Testing for the following conditions requires a known database state to be achieved • Several database fields, pointers and keys should be corrupted manually and directly within the database (via database tools) • Additional transactions should be executed using the tests from Application Function and Business Cycle Testing and full cycles executed
<p align="center">Completion Criteria</p>	<ul style="list-style-type: none"> • In all cases above, the application, database and system should, upon completion of recovery procedures, return to a known and desirable state • These states includes data corruption

<Online Movie Ticketing System> : Test Specification

	limited to the known corrupted fields, pointers / keys, and reports indicating the processes or transactions that were not completed due to interruptions
Special Considerations	<ul style="list-style-type: none"> • Recovery testing is highly intrusive, procedures to disconnect cabling (simulating power or communication loss) may not be desirable or feasible • Alternative methods, such as diagnostic software tools may be required • Resources from the System (or Computer Operations), Database, and Networking groups are required • These tests should be run after hours or on an isolated machine(s)

3.2.19 Upgrading Testing

Test Objective	<ul style="list-style-type: none"> • Ensure that the upgrade of the current version to a newer version/release/update is error free • Measure the impact of upgrading on the system
Technique	<ul style="list-style-type: none"> • Use a single machine with the minimum resources to see the impact after the upgrade • Test on multiple machines with different resources
Completion Criteria	<ul style="list-style-type: none"> • System works as per normal after upgrade
Special Considerations	<ul style="list-style-type: none"> • Scalability Testing done • System needs to be restarted (or restarted a few times) after the upgrade • Current system resources (hardware or software) are unable to support the upgraded version

3.3 Tools

The following tools identified in the table below will be employed for this project:

Types	Tool(s)
Project Management	Microsoft Project

4 Resources

The resources for the Online Movie Ticketing System test effort is the Testing Lead, Database Lead, Programming Lead and QA & Change Management Lead.

4.1 Workers

This table shows the staffing assumptions for the project.

Worker	Number of workers allocated	Specific Responsibilities/Comments
Quality Assurance and Change Management Lead	1	Identifies, prioritizes, and implements test cases and execute tests Responsibilities: <ul style="list-style-type: none"> • Generate test plan • Generate test model • Execute tests • Recover from errors • Evaluate effectiveness of test effort
Worker	Number of workers allocated	Specific Responsibilities/Comments

<Online Movie Ticketing System> : Test Specification

Testing Lead	1	<p>Provides management oversight, implement test cases and execute tests</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> • Provide technical direction • Acquire appropriate resources • Management reporting • Generate test plan • Execute tests • Log results
Programming Lead	1	<p>Identifies and defines the operations, attributes, and associations of the test classes and ensures test environment are managed and maintained</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> • Identifies and defines the test classes • Administer test management system • Install / manage worker access to test systems • Identifies and defines the test packages
Database Lead	1	<p>Ensures test data (database) environment and assets are managed, and maintained. Implements and unit tests the test classes and test packages</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> • Administer test data (database) • Creates the test classes and packages implemented in the test model.

4.2 System

All system tests will be executed on 5 different laptops. The Server Module will be mainly tested on a IBM eServer.

The server tests station must have the following softwares installed and properly configured:

- MYSQL Version 5
- Microsoft Windows Server 2003 R2

In order to make all Server Module tests more reliable, multiple tests will be executed at different traffic times:

- By day
- By night
- By morning
- By the peak hour

5 Project Milestones

Milestone Task	Effort	Start Date	End Date
Plan Test	17h	4 th Feb 2008	6 th Feb 2008
Design Test	21h	7 th Feb 2008	13 th Feb 2008
Implement Test	15h	14 th Feb 2008	18 th Feb 2008
Execute Test	38h	20 th Feb 2008	25 th Feb 2008
Evaluate Test	13h	25 th Feb 2008	26 th Feb 2008

6 Deliverables

6.1 Project Tasks

Refer to Appendix E

6.2 Test Cases

Refer to Appendix A

Refer to Appendix B

Refer to Appendix C

6.3 Test Results

Refer to Appendix D

Appendix A
Test Case for Add Movie

Appendix B
Test Case for Registration

Appendix C
Test Case for Booking

Appendix D
Test Results

Appendix E
Project Tasks

Below are the test related tasks:

- Plan Test
 - Test Objectives
 - Identify Requirements for Test
 - Risk
 - Develop Test Strategy
 - Identify Test Resources
 - Create Test Schedule
 - Generate Test Plan
- Prepare Test
 - Workload Analysis
 - Identify and Develop Test Cases
 - Identify and Structure Test Procedures
 - Confirm Test Tools and Techniques
- Execute Test
 - Execute Test Procedures
 - Evaluate Execution of Test
 - Verify the results
 - Log Defects
- Evaluate Test
 - Evaluate Test Results
 - Analyze Defects
 - Determine if Test Completion Criteria and Success Criteria have been achieved