



Show Time

SRS Document

Show Time - Online Ticket Booking	3
Introduction	3
Management Summary	3
Key Assumptions	3
High Level Architecture	4
Functional Requirements	4
Use Case Diagrams	5
Use Cases	5
<i>Upload Masters</i>	5
<i>Setup Cineplex</i>	6
<i>Maintain Movie Schedule</i>	8
<i>Booking Tickets</i>	8
<i>Issue Tickets</i>	9
<i>View Utilization</i>	10
Logical Object Model	10
Database Design Guidelines	11
Testing Approach	12
Suggested Technical Reading	12

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document should be used as a guideline by the students to design the Solution Architecture for the project. The document also describes the broad scope of the project and high level logical object model. But while developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation.

Show Time - Online Ticket Booking

Introduction

The purpose of this document is to define scope and requirements of an Showbiz initiative - "Show Time". It is a vision of providing unified access to customer anywhere anytime to their most coveted way of entertaining with friends and family.

This document should be used by the development team to architect the solution the project.

Management Summary

Idea of Show Time spawned with the advent of multiplexes in are country. With fast paced world, long distances to commute for work, it would be a welcome thought to bring convenience of scanning whats available where and accordingly people plan their outings. This initiative will address:

1. Setting up of Cineplexes
2. Hosting content for sneak a peek and current releases.
3. Online booking
4. Maintain Seat Availability Status
5. Utilization

The proposed solution will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

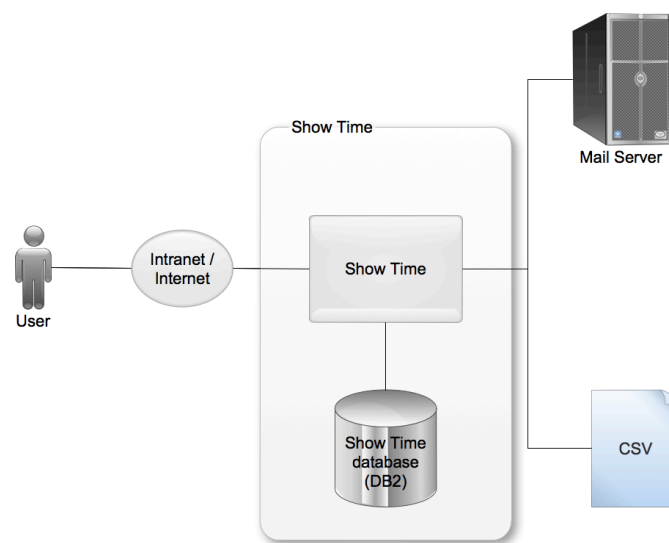
Key Assumptions

1. The system will not have payment option, the tickets are assumed to be collected from the ticket counter.
2. Customers are pre-registered is being assumed, no use case to register is mentioned in this document.
3. For the seat plan in the Audi, its recommended to use an HTML Table for Row/Column layout. You may use RowSpan, ColSpan to mimic the seating layout. Each cell will show either a seat or a aisle or empty area. A suitable color coding may depict it clearly. An on click function will as a toggle to block /unblock a seat. Such a function will be disabled on the seats have already been booked.

High Level Architecture

Show Time high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1. Administrator (Show Time)
2. Manager (Show Time)
3. Manager (Cineplex)
4. Ticket Counter (Cineplex)
5. Customer



Show Time Context Diagram

Show Time	Web application for providing access to information of shows running in various cineplexes, making online booking convenient for the customers, providing rights to cineplexes to enter their movie schedules and monitor the booking status
Show Time Database	Designed to store data for Cineplexes, locations, address, Auditoriums, customer registrations, bookings, availability, seating arrangement of Audi in Cineplexes.
CSV	Masters like location, Language, Showtime Users, Cineplex Users
Mail Server	All notifications are routed through the Mail Server

Functional Requirements

The high level functional requirements for the Show Time are outlined in the Use Case diagram described in this section.

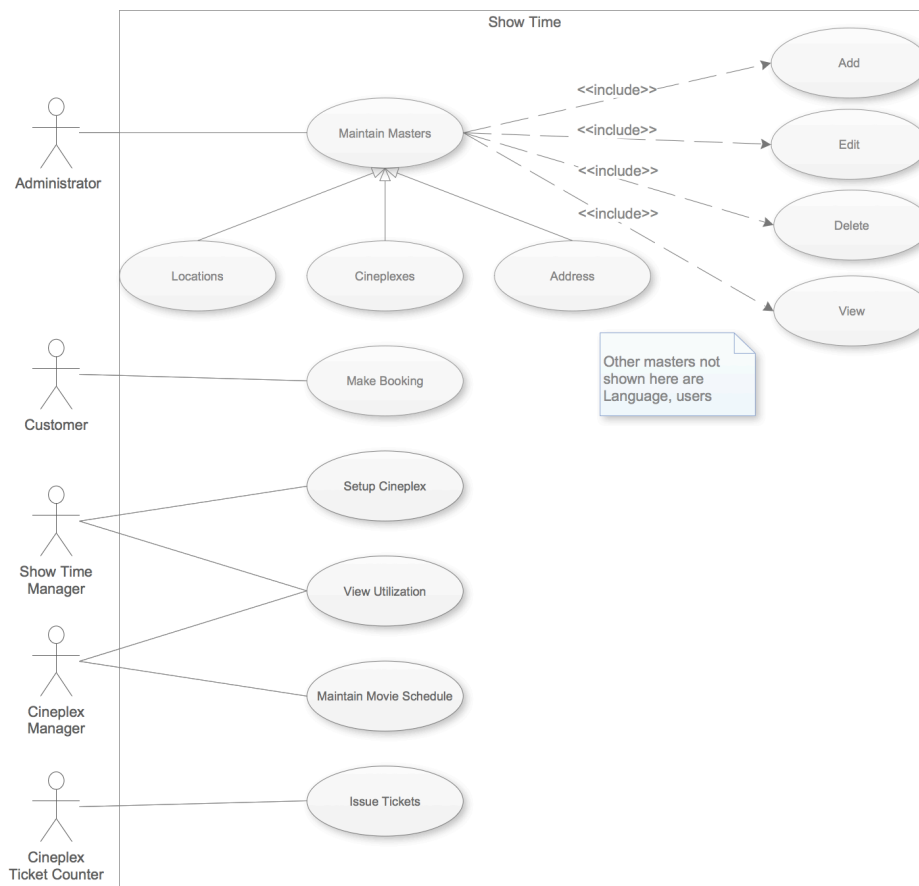
Show Time will provide a secure user-id/password based secured login mechanism to access its services. The details of this are not outlined here. The development team is expected to create these keeping in mind the general practices followed by

the web applications. Login will be a prerequisite to use Show Time. Internal users will be provided user id/password pair separately.

Once user logs in, s/he can view the Latest updates of releases. This menu options shall primarily come from the use case of the role player.

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system.



Use Case Diagram

Use Cases

Upload Masters

Use Case Element	Description
Number	UC.01

Use Case Element	Description
Application	<p>Masters in Show Time are uploaded using CSV Format</p> <p>Location Master contains Location id, Location Name (Delhi, Noida, Faridabad, Ghaziabad)</p> <p>Area Master contains Area id, Area Name, Location id</p> <p>Language Master contains Language id, Language</p> <p>Show Timing Slots Master contains: Slot type (Morning, Noon, Afternoon, Evening, Night) Start Time, End Time</p> <p>User Master contains User id, User Name, Role (Manager / Admin / Counter), User Type(Cineplex, ShowTime)</p>
Use Case Name	Upload Masters
Primary Actor	Content Manager
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Upload Masters menu item on the landing page
Basic Flow	<p>System prompts for the file name to be uploaded. Standard file upload dialog is presented to select a file from the local system.</p> <ul style="list-style-type: none"> The selected file data is uploaded in the related tables; if an existing record is encountered, the old details are replaced with the new details.
Alternate Flow	<ul style="list-style-type: none"> In event of incorrect CSV format, system gives an error and NO data is uploaded. Operation is cancelled
Output	System displays the number of records uploaded. It also highlights the number of records updated (i.e. already existing ones being replaced)

Setup Cineplex

Use Case Element	Description
Number	UC.02
Application	The Showtime manager adds new Cineplexes operating in the Locations of their business. The Cineplex users then can upload their content for customers to view and make bookings
Use Case Name	Setup Cineplex
Primary Actor	Show Time Manager
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Setup Cineplex link on the landing page

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none">• The system displays a form with following fields for the user to enter.• Name of Cineplex, Address 1, Address 2, Area[Picker], Location [Picker]• Number of Audi, Number of Seats per Audi, Manager [Picker from the user list where role type is Manager], Counter [Picker from the user list where role type is Counter] <p>On Submit the system creates a profile of the Cineplex information being entered.</p> <p>Once added, this cineplex will start to be displayed in the list of Cineplexes available in a Area or location.</p>
Alternate Flow	On Cancel, there is no impact on database
Output	A mail notification to the Manager selected in the Cineplex profile to Setup up the movie shows.

Maintain Movie Schedule

Use Case Element	Description
Number	UC.03
Application	Cineplex Manager enters the schedule for the movie
Use Case Name	Maintain Movie Schedule
Primary Actor	Cineplex Manager
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the Maintain Movie Schedule link on the landing page.
Basic Flow	<ul style="list-style-type: none"> The system displays the list of movies, Audi Number currently playing in the Cineplex with radio buttons to multi select movies for editing. The selected movie names, show timings with check boxes are displayed as checked by default for the shows that are on. The user may remove a movie completely by clicking on link to delete in front of the movie name or may choose to remove some of the show timings by unchecking the showtime entry. The user may check a show time to add a new time slot for beaming the movie. Effective Date is selected using a Calendar control. (Normally the movie schedule changes after a week, for the purpose of this project, it can be changed any day and the change will take effect on that date itself on save). System will validate if the tickets are already issued for a movie, if yes, then it will alert the user to inform the customers. Add movie button on top of the page will allow the user to enter the movie name, select type(U/A), language, Cast, attach a poster image, select time slots and specify Audi Number. On submit, the information gets saved and updated as per the maintenance activity.
Alternate Flow	None
Output	Email to customer in case of change of schedule or movie being removed from the Cineplex

Booking Tickets

Use Case Element	Description
Number	UC.04
Application	The customers search for cineplex in the area or location, select a movie to watch and block the tickets.
Use Case Name	Booking Tickets
Primary Actor	Customer
Secondary Actor	None

Use Case Element	Description
Pre-condition	User is an authorized role in the system
Trigger	The user clicks on the Book Tickets link on the landing page.
Basic Flow	<p>The system displays list of locations to select from. The user selects a location. This areas under that location are listed along with the Cineplexes in that Area.</p> <p>User selects a Cineplex to view the current movies.</p> <p>The list of movies with poster image, movie name, type, cast are displayed along with the schedule.</p> <p>The user clicks on one of the schedule for the movie being played, the system displays the dates in that week for booking, Number of seats</p> <p>The system displays a seat plan window for the number of seats of the Cineplex's audi. The seat plan functionality will be as per the note given in assumptions.</p> <p>The user selects the seats and clicks on the book tickets.</p> <p>A reference number is generated for the user's booking and an email is sent along with the reference number to bring at the counter for ticket issue and payment.</p> <p>The system reduces the number of seats of an audi by the seat count booked by the user.</p> <p>Note: System will not allow any user to book beyond the seating capacity, block seats that are already blocked,</p>
Alternate Flow	Pressing Cancel abandons operation, no database gets affected
Output	Email to customer with booking reference

Issue Tickets

Use Case Element	Description
Number	UC.05
Application	The counter at cineplex will issue tickets booked via the system.
Use Case Name	Issue Tickets
Primary Actor	Ticket Counter Staff
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the Issue Ticket link on the landing page.

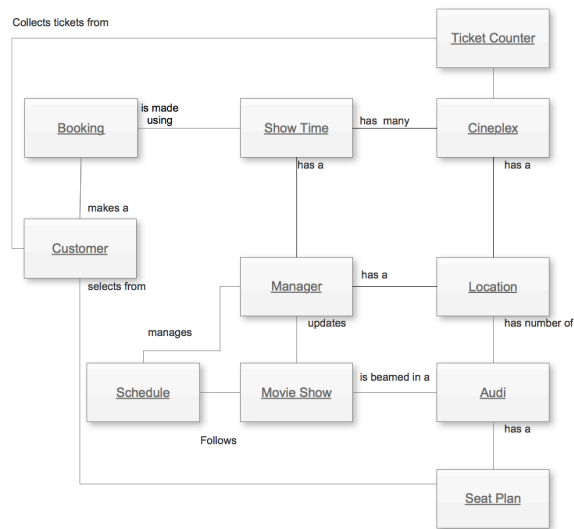
Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> User enters the reference number provided by the customer. If the reference number has not been issued earlier, the system will pull up the booking data and display movie name, show time, no. seats. User confirms the issue of ticket (no ticket generation required in the project) The system adds the utilization count for an Audi by the number of seats in the ticket issued. The booking is marked as closed, no further ticket can be issued for this booking reference.
Alternate Flow	Pressing Cancel abandons operation, no database gets affected
Output	None

View Utilization

Use Case Element	Description
Number	UC.06
Application	The Managers of Cineplex and Showtime keep track of the booking and utilization status
Use Case Name	View Utilization
Primary Actor	Manager (Show Time)
Secondary Actor	Manager (Cineplex)
Pre-condition	None
Trigger	The user clicks on the View Utilization link on the landing page.
Basic Flow	<ul style="list-style-type: none"> Show Time manager will get to see list of Cineplexes as group heading, Audi Counts as rows and booking count, utilization count as data for each of the Audi number in the Cineplex. Cineplex Manager will get to see only data mentioned above for his/her own Cineplex.
Alternate Flow	None
Output	None

Logical Object Model

A high level logical object model of the system is shown below. During technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.



Logical Object Model

1. Show Time is a host website for provider of information on various cineplexes operating at various locations.
2. Show Time maintains masters for locations and languages of films
3. Manager of Show Time has the responsibility of adding cineplex to the Show Time website and configured for the number of Audis and their seating plan.
4. Manager at the Cineplex is responsible for posting new releases and maintaining schedule for the shows in a week starting every Friday.
5. Customer visits the Show time website to search for the new releases and current shows in various cineplexes.
6. Customer makes a booking via Show time.
7. Seat plan feature is used while making booking online.
8. Each Cineplex has a ticket counter. Customer collects tickets from the Cineplex Ticket counter. The system keeps track of audi utilization.

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimized physical database table designs.

Typically persistent classes will map to table(s) with their attributes as columns of the table. In some cases a high level object may map in to a master-child table. Invoice is one such example where it maps in to "invoice_header" and "invoice_line_item" table.

Associations between two persistent objects are realized as foreign keys to the associated objects. A foreign key is a column in one table that contains the primary key value of the associated object.

Similarly, a standard technique in relational modeling is to use an intersection entity to represent many-to-many associations. Following is a broad checklist for physical database database design:

1. Database must be properly normalized except those instances where de-normalization help improves performance. This option must be used with special care.

2. All persistent classes that use the database for persistency must map to database structures.
3. Many-to-many relationships must have an intersecting table.
4. Primary keys should be defined for each table, unless there is a performance reason not to define a primary key.
5. Indexes should be defined to optimize access.
6. Data and referential integrity constraints should be defined.

Testing Approach

Quality of the software can be achieved with basic hygiene and consistency followed during design and development of User Interface(UI), Navigation, Validations as per the business process requirement.

To ensure the project delivers acceptable quality to the customer, its important to create a checklist of the conventions to be followed across. Common checks as below are for your reference during design and development:

Common Checks	Validation Type
Page Title is valid for the feature being provided on the page	UI
Order of the Data Entry Fields is logical as per the functionality being provided by the feature	UI
Order of the Display only Fields makes viewing and understanding easy for the user	UI
Spellings and Correctness of Label for the Data Entry and Display fields	UI
The labels are not wrapping onto another row thereby adding a blank row on the page	UI
The fields with drop down are displayed in single row instead of drop down coming on the next row	UI
Data Entry field basic validations are working i.e Text field /Numbers / Dates allow data for their type only	Functional
The dates are following a standard format dd/mm/yy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI
The action buttons are performing the desired action e.g. "submit" is creating a new record if there are no errors and recording all the input fields, whereas 'cancel' is not creating a new record in the database	Functional
The links provided on the forms are opening correctly.	Functional
The data feed mechanism for Read and Write files is generating a log with count of entries.	Navigation

Suggested Technical Reading

The project is aimed at making the student understand concepts of Design and Development using IBM Rational tools, Web Sphere Application Server and DB2 Database. The following reading reference is easy to understand and should be read to get a clear understanding of capabilities of the tools and how you would leverage them to execute a project.

Technical Reference	URL to access
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html

Technical Reference	URL to access
IBM Education Assistant - Rational Application Developer 7.5	http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/