

# **Movie Tickets Booking Application**

SRS Document

## Table of Contents

<b>Introduction.....</b>	<b>3</b>
System User.....	3
Assumptions.....	3
<b>System Architecture.....</b>	<b>4</b>
<b>Functional Requirements.....</b>	<b>5</b>
Object Diagram .....	6
Push Notifications.....	7
Use Cases.....	7
View Movie .....	8
Make a Booking.....	8
View Booking Status.....	9
<b>ABOUT HAVERSINE FORMULA .....</b>	<b>9</b>
<b>TESTING GUIDELINES.....</b>	<b>9</b>
<b>SUGGESTED READING.....</b>	<b>10</b>
TOOLS.....	10
USER INTERFACE.....	11

# Introduction

The Movie Booking System revolutionizes the way movies are enjoyed. Seamlessly blending technology and entertainment, it offers a user-friendly online platform for discovering, selecting, and booking movie tickets. Users can easily explore movie options, check show time, pick seats, and make secure payments, eliminating the hassle of traditional ticketing. The system's intuitive interface provides real-time seat availability and layout information. Secure online payment ensures a smooth transaction process, enhancing the overall user experience. Notably, the system extends its advantages to theater administrators through an efficient administrative panel, enabling effective management of movie listings, show time, and seating arrangements. This synergy between user convenience and theater management simplifies the movie-going experience for all, setting a new standard in the world of cinema.

## System User

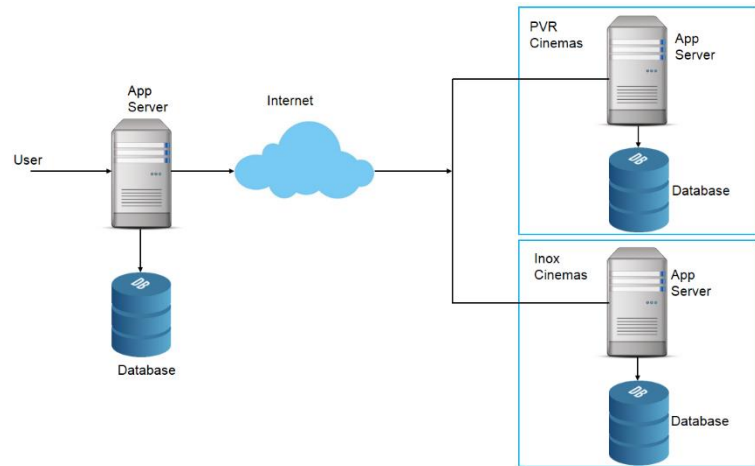
The system user, often known as an admin, plays a pivotal role in the Movie Booking System. They oversee movie listings, show times, and user accounts, ensuring a seamless experience. Admins manage seat availability, resolve issues, and maintain the system's integrity. Through their actions, they enhance user satisfaction and streamline the overall movie booking process.

## Assumptions

1. **Stable Internet Connectivity:** Users have reliable internet connectivity to browse movies, book tickets, and make online payments.
2. **Accurate Movie Data:** Movie details, show times, and availability are assumed to be accurate and updated in real time.
3. **Secure Payment Processing:** The system assumes integration with secure and trusted payment gateways to handle online transactions.
4. **User Authentication:** User registration and login assume a secure authentication process to protect user accounts and data.
5. **Booking Confirmation:** The system provides instant booking confirmations via email or within the user's account after successful payment.

These assumptions form the foundation for the functionality and user experience of the Movie Booking System.

# System Architecture



1. The Movie Booking System's architecture is structured to provide a seamless and efficient experience for users. At its core, the system comprises three main layers: The Presentation Layer, Application Layer, and Database Layer.
2. The Presentation Layer is the user interface where users interact with the system. It includes the web application accessible via browsers or mobile devices. Users can explore movies, view show times, select seats, and complete bookings through this interface.
3. The Application Layer hosts the system's logic and processes. It handles user authentication, validates bookings, checks seat availability, processes payments, and manages communication between the Presentation and Database Layers.
4. The Database Layer stores essential data, including movie details, show times, user profiles, and booking records. It ensures data integrity, supports efficient querying, and ensures the security of sensitive information.

# Functional Requirements

Functional Requirements for Internet-Based Movie Booking System with Authentication:

## 1. User Registration and Authentication:

- Users can create accounts with unique usernames and passwords.
- Users can log in securely using their credentials.
- Forgot password functionality allows users to reset passwords via email.

## 2. Browsing Movies:

- Users can view a list of available movies with details such as title, genre, and rating.
- Each movie entry includes a clickable link to view more information.

## 3. Viewing Movie Details:

- Users can access detailed information about a selected movie, including a synopsis, cast, duration, and trailers.

## 4. Showtimes and Seat Availability:

- Users can choose a movie and view available showtimes.
- For each showtime, users can see the available seats in a visual layout.

## 5. Seat Selection and Booking:

- Users can select seats from the layout for a chosen showtime.
- The system prevents seat conflicts and overbooking.
- Users can review their seat selection and confirm the booking.

## 6. Online Payment:

- Users can pay for their booked seats using various payment methods.
- Payment processing is secure and integrated with trusted payment gateways.

## 7. Booking Management:

- Users can view their booking history, including details like movie title, showtime, and seats.
- Users can cancel bookings within a specific time frame before the showtime.
- Cancelled seats are released for others to book.

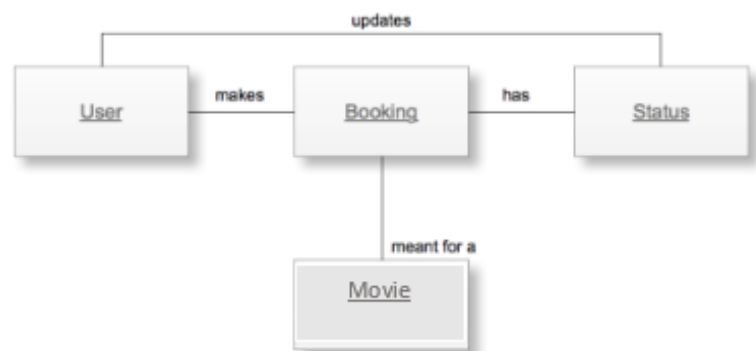
#### 8. Admin Panel:

- Admins can access an admin panel with elevated privileges.
- Admins can manage movie information, showtimes, and user accounts.
- Adding, updating, and deleting movie records is possible through the admin panel.

#### 9. User Feedback and Support:

- Users can provide feedback, report issues, or seek assistance via a contact form.

### Object Diagram



1. User makes a booking for a Movie.
2. Locations are pre-defined in terms of various localities in the city.
3. Movie details include address, timings, location name.  
The latitude and the longitude are obtained using Google Geocoding API located at <https://developers.google.com/maps/documentation/geocoding/>. The application displays cinema hall address and timing on touching one of the location from the list displayed on the users mobile.
4. A booking has a user id, date & time.
5. Status of a booking keeps getting updated.

6. Booking status appears, when the user access the application on their mobile, if the user has made a booking for a movie.

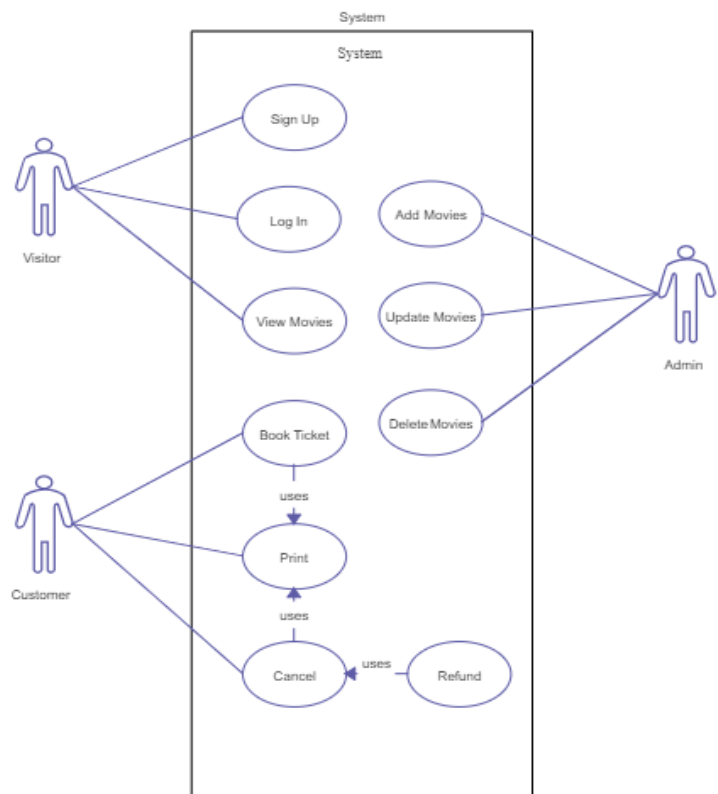
This object diagram illustrates the key objects and their relationships in the Movie Booking System. Users interact with movies, show times, and bookings, while administrators manage the system's data and functionality. The Payment Gateway object handles payment processing securely. The System object orchestrates the interactions and functions within the system.

## Push Notifications

Push Notifications are triggered by the back-end system to notify the user of the booking expiry. The back-end system explicitly pushes notification(s) and the same are retrieved by the Work light adapter's event source. A push also updates the user's booking status to 'Ready to Serve' or 'Expired' in the app. In absence of necessary prerequisites for notifications, the actual pushing of notification may be ignored.

## Use Cases

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



## View Movie

Use Case Element	Description
Browsing Movies	Users can explore a list of available movies, viewing titles, genres, and ratings for each movie.
Viewing Movie Details	Users can select a movie to access detailed information, such as its synopsis, duration, and release date.
Show times and Seat Selection	Users can choose a movie and see its available show times, enabling them to plan their movie experience.
Booking Tickets	Users can select a movie, choose a show time, and proceed to select seats for booking tickets.
Admin Movie Management	Admins can manage the "Movies" table, adding new movies, updating details, or removing movies.

## Make a Booking

Use Case Element	Description
Application	To make a booking
Primary Actor	Mobile User
Trigger	User invokes restaurant for the current location using appropriate gesture on touch screen of the smart phone.
Basic Flow	The basic flow of the movie application involves users exploring available movies, selecting a movie, viewing its details and showtimes, choosing a showtime, selecting seats, and booking tickets. Admins can manage movie information. Users can explore, select, and book movies seamlessly, enhancing the movie-watching experience.



## View Booking Status

Use Case Element	Description
Application	To make a booking
Primary Actor	Mobile User
Trigger	User invokes cinema for the current location using appropriate gesture on touch screen of the smart phone.
Basic Flow	Users' booking status transitions from seat selection to payment confirmation, concluding with a successful booking.

## ABOUT HAVERSINE FORMULA

The Haversine formula is an equation important in navigation, giving great-circle

distances between two points on a sphere from their longitudes and latitudes. The

basic details can be found at Wikipedia entry located at

[http://en.wikipedia.org/wiki/Haversine\\_formula](http://en.wikipedia.org/wiki/Haversine_formula) URL.

## TESTING GUIDELINES

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. Here are some essential guidelines:

1. Plan on unit testing mobile applications early and frequently during

development. This is simply because mobile testing is challenging, given the

wide range of possible devices and platforms for even a single application.

2. IBM Worklight Server provides an app preview service that enables you to

simulate mobile web artifacts in a desktop web browser. App preview also

enables the simulation of Worklight client APIs and it is available from the

Worklight Studio test server console. Use it as a quick and easy way to test your app during early stages of development.

3. Use native mobile SDK simulator to perform near real life like testing on your

desktop.

4. Finally use the real device to test your app. For example, when an Android device is connected to the computer via a USB cable, the Eclipse ADT plug-in automatically recognizes it and attempts to deploy applications onto it.

5. In order to test your mobile application with full back end enterprise system integration, the application needs to be installed in an integrated environment having all the components. Plan for setting up such environment at an early stage.

6. Testing push notifications, if applicable, will require necessary account set

up for the target device(s). Some key examples are Android Cloud to Device

Messaging Framework (now Google Cloud Messaging – GCM) or Apple

Push Notification service (APNs for short).

## SUGGESTED READING

The project is aimed at making the student understand concepts of (a) Design and

Development using IBM Worklight, IBM Rational tools, Web Sphere Application

Server and DB2 Database; and (b) User Interface Design for Mobile Devices.

## TOOLS

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.

Resource	URL
RAD-Tackling challenges of software Development with Rational Application Developer For WebSphere Software	<a href="http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html">http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html</a>
IBMEducation Assistant – IBM Worklight V5	<a href="http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.worklight/worklight/5.0/Back-End_Connectivity/Worklight_Backend_Connectivity/player.html">http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.worklight/worklight/5.0/Back-End_Connectivity/Worklight_Backend_Connectivity/player.html</a>
Getting started with IBM Worklight	<a href="https://www.ibm.com/developerworks/mobile/worklight/getting-started/">https://www.ibm.com/developerworks/mobile/worklight/getting-started/</a>

## USER INTERFACE

Designing a UI for mobile devices is not identical to desktop or web application UI design. It is not about simply adding controls, some text and color to each screen. Here gestures play a vital role, chrome has to be minimized. The following UI design guidelines, developed by Google for Android and by Apple for iOS, will give you insight in to how to create great UI for mobile applications.

Resource	URL
Google Android UI Guidelines	<a href="http://developer.android.com/guide/topics/ui/index.html">http://developer.android.com/guide/topics/ui/index.html</a>
Apple iOS Human Interface Guidelines	<a href="http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html">http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html</a>