**Project On Online Event Booking Website**

**EventAdda**

Under the guidance of

**Mrs Kishori Khadilkar**

**Submitted By**

Shadad Yunus Khan

Gaurav Patil

Harshal Satpute

Shirish Gupta

Aditya Dileep Sawant

Faizan Ahmed

**Table Of Contents**

1. Synopsis
2. Requirements
3. Collection’s Details
4. Class Diagram
5. Sequence Diagram
6. Activity Diagram

**Synopsis**

**EventAdda** – Event booking web application

**Introduction**

EventAdda is a web based application which provides the interface for booking events such as movies, sports, concerts and plays. Application uses JavaScript framework Angular 6 for the front end while the data is fetched from MongoDb database using Java web services developed using spring and hibernate framework.

**Project Description**

Booking a particular event can be a tricky task. EventAdda makes it easier by using a simple search, book and pay paradigm. User is provided a number of trending and popular events in his/her locality. If that doesn’t fit the requirement, user has the ability to search a particular event by name or applying filters such as location, price, genre, date, time and much more. These events can be further categorized into movies, sports, concerts and plays.

Now that the user has settled on a particular event, he/she may select the available seats as needed. After selecting the seats, user is asked to login if not already logged in. On successful authentication, the user is redirected to payment option which is completed by a payment gateway provided by third party application. Booking and ticket details are confirmed to the user on successful transaction.

**Project Technology**

Since Angular 6 utilizes MVC architecture for development, it provides an easier and faster development methodology for the front end. Application is divided into separate individual components such as a particular event like concerts which can be independently developed. Use of templates for generating views for website reduces the code and provides faster performance.

Functionalities such searching for an event, populating the event list based on filters, booking event, authenticating user are fulfilled using Java web service. Java web service is developed using spring and hibernate framework which also utilizes MVC architecture. Web service provides a path using which the front end may access the functionalities and in return it receives the data in JSON format. This JSON data can contain a simple key value pair or an array of hundreds of element or combination of both. Web service uses Data Access Object layer for accessing database content.

Database is handled using MongoDb – a document based database system). Since MongoDb is a NoSQL database and focuses more on the performance rather than the storage size of the data while providing high scalability and availability option, it is beneficial for an application like EventAdda which can’t afford to be down for large periods.

**Collection Details**

**Customer**

-Cust\_ID(PK)

-Name

-Mob\_no

-DOB

-Gender

-Addresses[{street:"",city:"",state:""}]

-Email

**Booking**

-Booking\_id(PK)

-Book\_status

-Timestamp

-Payment\_status

-Event\_id

-Customer\_id

-No\_of\_Tickets

**Events**

-Event\_id

-Name

-Categories[{Name:"", Genre:""},{}]

-Venue\_id

-Timestamp

-Artist

-Language[{}]

**Ticket**

-Booking\_id

-seat\_loc

-Price

**Venue**

-Venue\_id

-name

-loc

-section

-contact\_number

**Requirements**

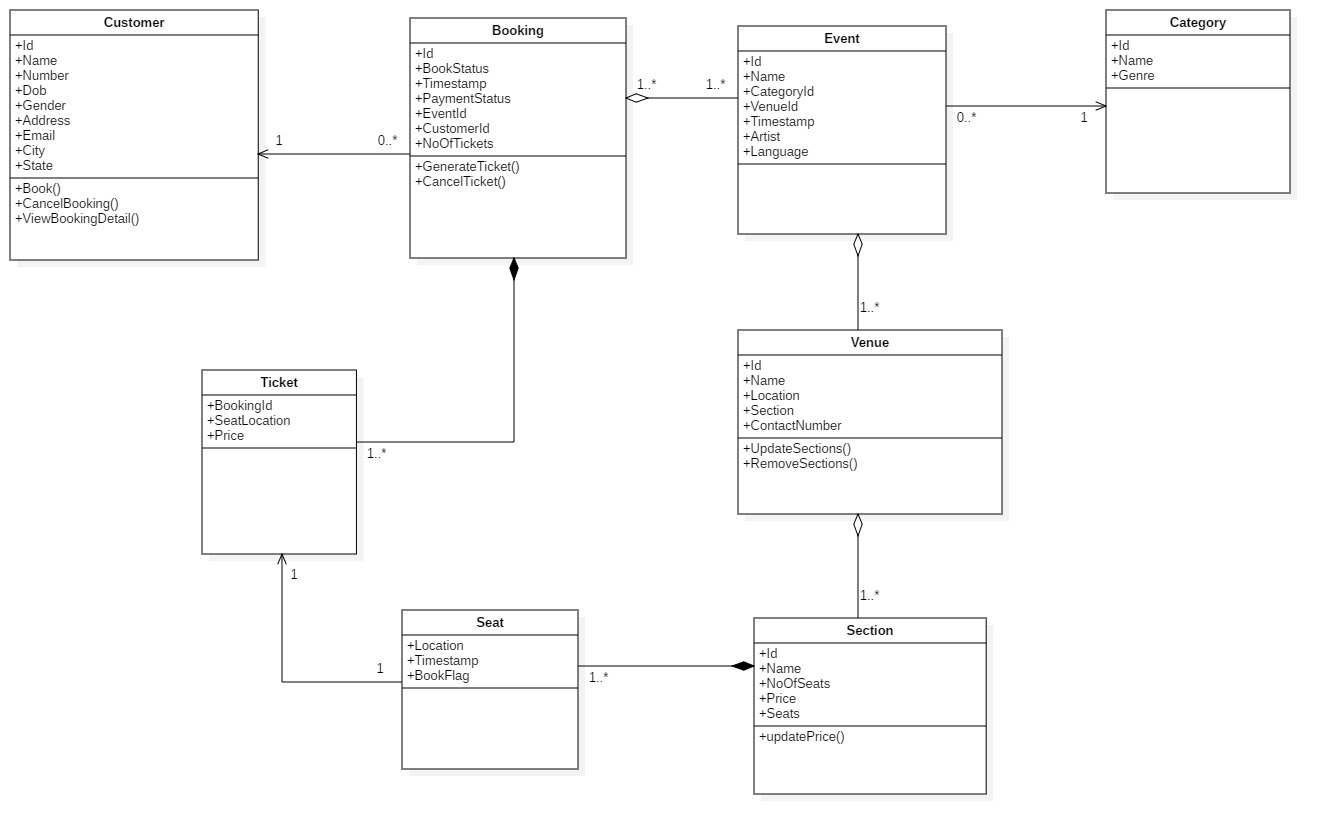
**Software Requirements:**

* Angular 4.0 or above
* Visual Code
* J2EE
* Restful Web Services
* MongoDB
* Eclipse
* Apache Tomcat

**Hardware Requirements:**

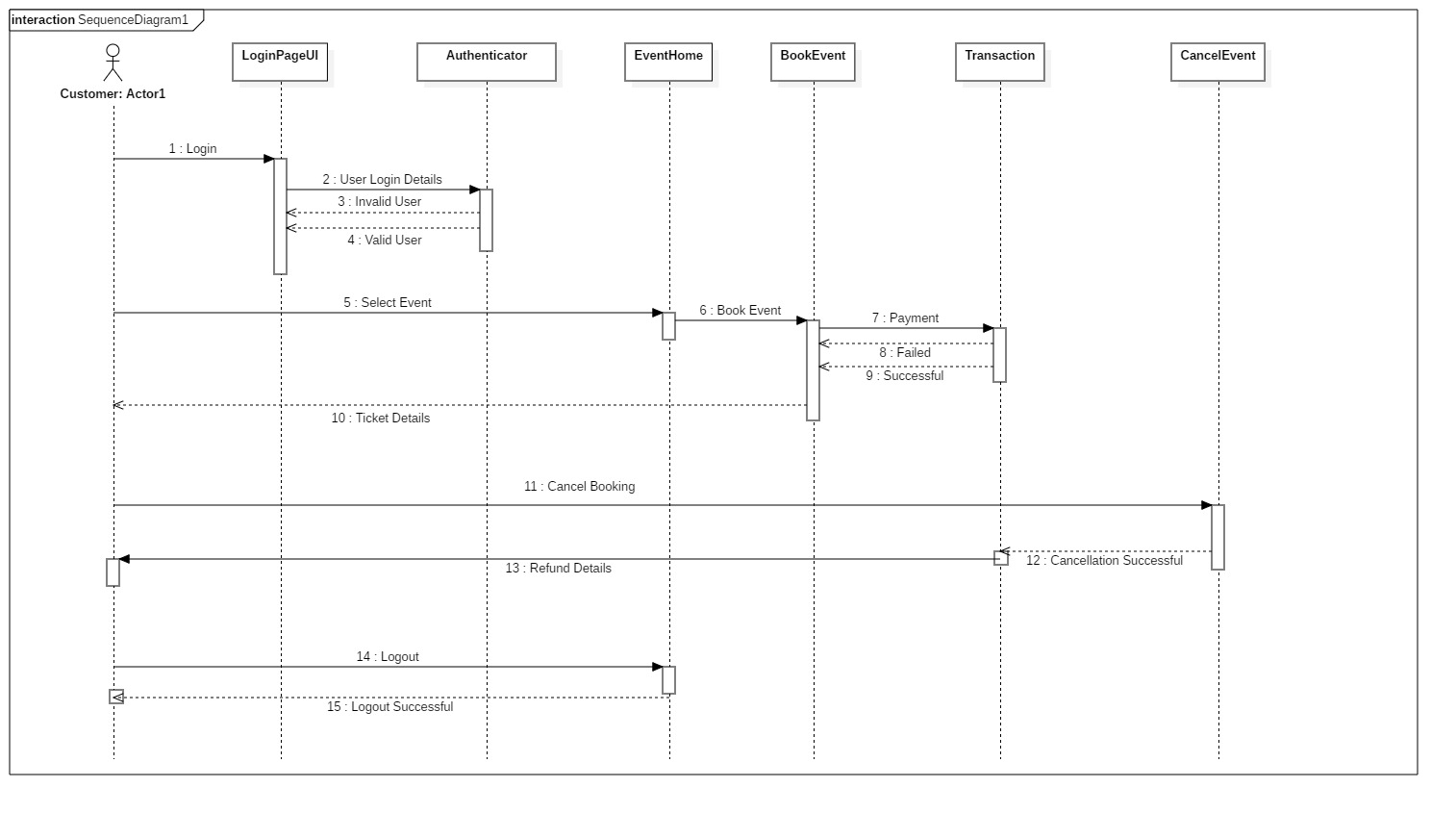
* RAM: 1GB
* Processor: Pentium

**Class Diagram**

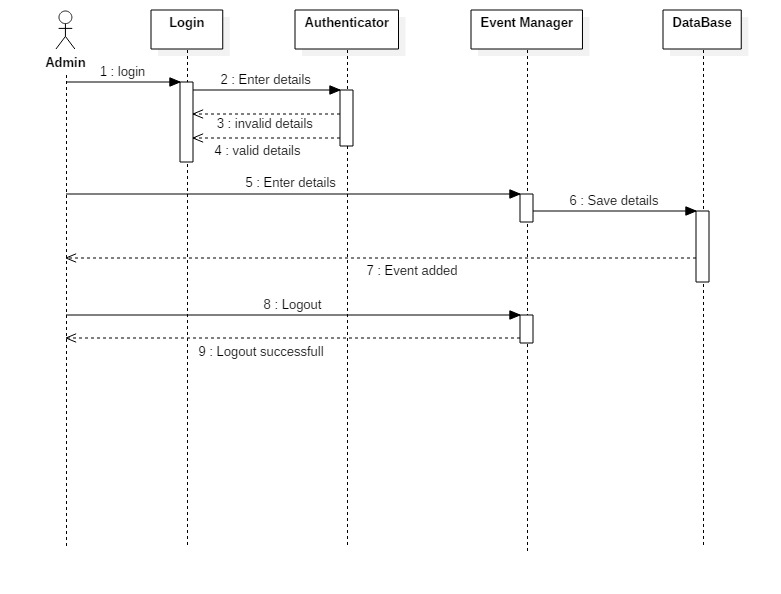


**Sequence Diagram**

1. Book Event Sequence Diagram

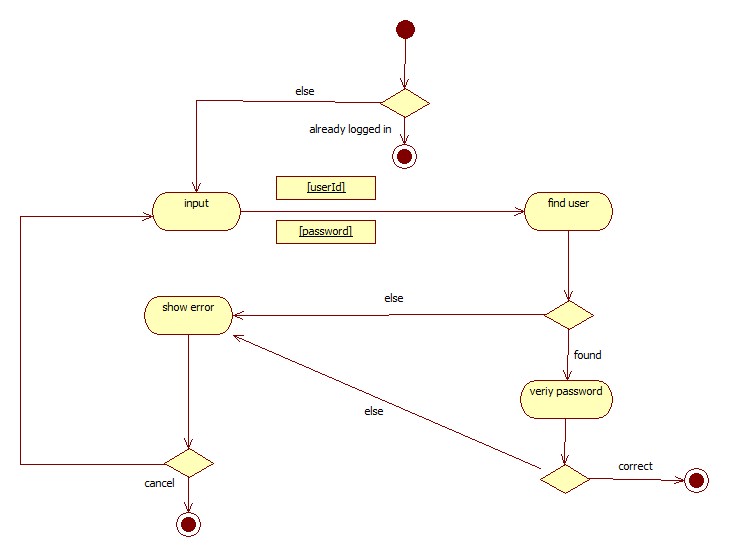


1. Add Event Sequence Diagram

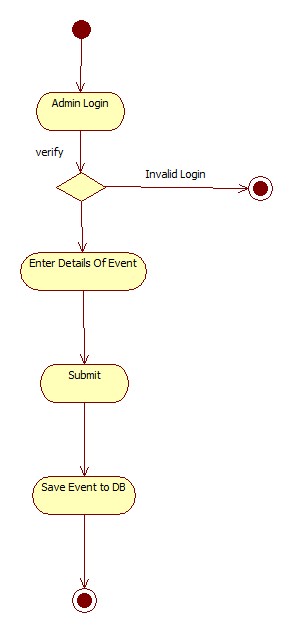


**Activity Diagram**

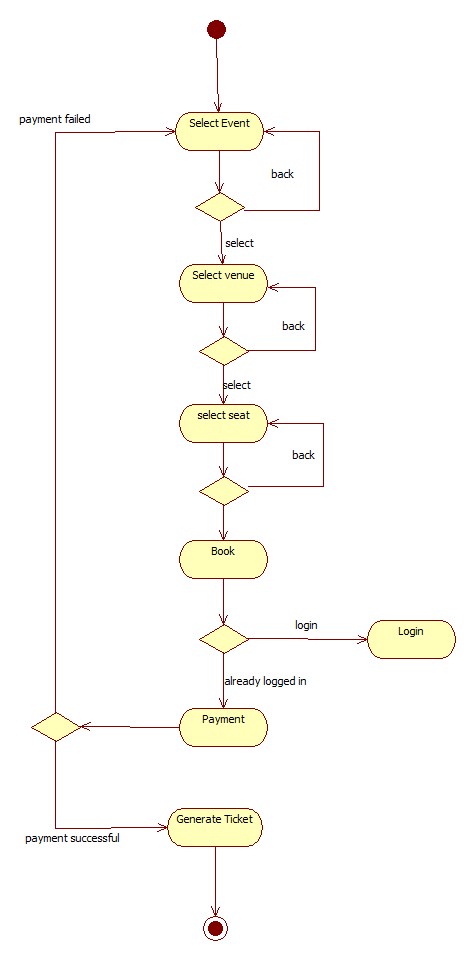
1. Login



1. Add Event Activity



1. Book Event



1. Cancel Booking

