**PROJECT DOCUMENTATION**

**ON**

**MOVIE BINGE**

****



**By:**

Vinamra Sareen



**CERTIFICATE**

**This is to certify that this report, titled MOVIE BINGE embodies the original work done by Vinamra Sareen in partial fulfillment of their course requirement at NIIT.**

**COORDINATOR**

**ACKNOWLEDGEMENT**

**I have benefitted a lot from the feedback and suggestions given to me by Mrs. Lopamudra Bera and other faculty members.**

**REQUIREMENTS OF PRODUCT**

**The software and hardware product requirements for Movie Binge:**

**Hardware specifications for running Movie Binge**

**There is no minimum requirement, you may use old hardware (like Pentium 2 or older), 128MB RAM will be enough and a 20GB HD will be big enough.**

**Software Details for running Movie Binge**

**Web Server (preferably Apache Tomcat)**

**H2 Database configuration**

**Technologies used:**

* **Bootstrap**
* **jQuery**
* **Font-Awesome**
* **HTML5**
* **CSS3**
* **JavaScript**
* **Google Fonts**
* **Spring MVC**
* **Hibernate**
* **Maven**

**And many jars i.e., dependencies.**

**PROJECT LIFE CYCLE MODEL**

**Initiation Phase**

**VISION DOCUMENT**

**PROJECT PLAN**

**Project Name: Movie Binge**

|  |  |
| --- | --- |
|  | Milestone for Movie Binge |
| Milestone 1 | * **Created repository for Movie Binge on GitHub** * **Initialize this repo** * **Added Custom Controller for landing page** |
| Milestone 2 | * **Configure Hibernate and H2 database with Spring MVC** |
| Milestone 3 | * **Admin Panel performing CRUD operations for movies section** |
| Milestone 4 | * **Landing Page** * **Login and Booking Validations** |
| Milestone 5 | * **Booking Module** |

**Time Duration for Project Completion**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Milestones | Planned Start Date | Planned End Date | Actual Start Date | Actual End Date |
| Milestone 1 | **10 April** | **19 April** | **14 April** | **20 April** |
| Milestone 2 | **20 April** | **21 April** | **20 April** | **21 April** |
| Milestone 3 | **23 April** | **31 April** | **25 April** | **1 May** |
| Milestone 4 | **1 May** | **8 May** | **2 May** | **10 May** |
| Milestone 5 | **10 May** | **15 May** | **11 May** | **15 May** |

**WEEKLY STATUS REPORT**

1. **April – First Week** 
   1. **April 14** **- Created Repository for Movie Binge on GitHub**
   2. **April 18 – Initialized the repo**
   3. **April 19 - Added Custom Controller [Index Controller] for landing page**
2. **April – Second Week**
   1. **April 21 - Configured Hibernate and H2 database with Spring**
3. **April – May [End on April and Start of May]**
   1. **May 1 - Admin panel performing CRUD operations for movies section**
4. **May - Second Week**
   1. **May 7 - Landing page - [Partially Completed]**
   2. **May 9 – Login and Booking Validations**
5. **May – Third Week** 
   1. **May 15 – Booking Module Completed**

**WEEKLY STATUS REPORT**

|  |
| --- |
| **ACTIVITIES COMPLETED THIS WEEK** |

* **Created Repository for Movie Binge on GitHub**
* **Initialized the repo**
* **Added Custom Controller [Index Controller] for landing page**

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES IN PROCESS** | **NEXT ACTION** | **DUE DATE** |
| Mapping URLs in Controller for other  requests | Database Configuration | 21 April |

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES TO BE STARTED NEXT WEEK** | | |
| Hibernate and H2 Database Configuration | POJO for Movie, Movie  Details, Users, Show  and Trailers | Data Access  Objects for the  Same models. |
| Implement Dao to create queries and get  data |  |  |

|  |
| --- |
| **ISSUES FOR IMMEDIATE ATTENTION** |
| Not Yet |

**WEEKLY STATUS REPORT**

|  |
| --- |
| **ACTIVITIES COMPLETED THIS WEEK** |

* **Configured Hibernate and H2 database with Spring**

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES IN PROCESS** | **NEXT ACTION** | **DUE DATE** |
| Creating queries to perform CRUD  operations | Create queries to  display data on  homepage | 1 May |

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES TO BE STARTED NEXT WEEK** | | |
| Admin Panel performing CRUD operations |  |  |

|  |
| --- |
| **ISSUES FOR IMMEDIATE ATTENTION** |
| Handling exception in case of database connection error or other exceptions |

**WEEKLY STATUS REPORT**

|  |
| --- |
| **ACTIVITIES COMPLETED THIS WEEK** |

* Admin Panel performing CRUD operations

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES IN PROCESS** | **NEXT ACTION** | **DUE DATE** |
| Landing Page | Booking Module | 9 May |

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES TO BE STARTED NEXT WEEK** | | |
| Landing Page [ Homepage] | Booking and Validations |  |

|  |
| --- |
| **ISSUES FOR IMMEDIATE ATTENTION** |
| * Adding Custom Validations and implementing messages wherever required. * Login Security and session handling |

**WEEKLY STATUS REPORT**

|  |
| --- |
| **ACTIVITIES COMPLETED THIS WEEK** |

* **Landing page - [Partially Completed]**
* **Login and Booking Validations**

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES IN PROCESS** | **NEXT ACTION** | **DUE DATE** |
| Service for seat booking | * Seat Booking Module * Get Seats based on date and time * Calculate Price in real time | 15 May |

|  |  |  |
| --- | --- | --- |
| **ACTIVITIES TO BE STARTED NEXT WEEK** | | |
|  |  |  |
|  |  |  |

|  |
| --- |
| **ISSUES FOR IMMEDIATE ATTENTION** |
| * Apply Transaction to book seats in case if more than one user tries to book the same seat then the first query proceeds. * In case on failure the transaction should be rolled back. * Errors should be handled carefully * Seats that are already booked must be shown by selection. |

**WEEKLY STATUS REPORT**

|  |
| --- |
| **ACTIVITIES COMPLETED THIS WEEK** |

* Booking Module Completed

**FUNCTIONAL REQUIREMENT DOCUMENT**

* **Customer can perform these functions:**
  + **Book movies**
  + **Seat Selection based on date and time**
  + **Cancel tickets**
  + **Browse to see all the movies currently playing**
  + **Browse to see all the upcoming movies**
  + **Verify the credentials at the time of login**
  + **Provide a mechanism to restrict access to administrative functions based on the fact whether the current user is admin or not.**
  + **Provide a mechanism to check for the current session, if session exist perform functions [sql queries].**
  + **Removes session after a long inactivity.**
* **Admin can perform these functions:**
  + **Add new movie**
  + **Update the movie details**
  + **View all movies**
  + **Delete movies**
  + **Add Show timings**
  + **Delete Show timings­­­**

**NON-FUNCTIONAL REQUIREMENT DOCUMENT**

**Usability**

* **Easy to navigate and tap buttons and navigation menu links.**

**Vision**

* **Appropriate messages should display wherever required. Most probable situations are successful logged in, signup.**

**Human Error**

* **Proper validations have been applied on forms.**

**Security**

* **According to industry best practices covered most of the security concerns.**

**Scalability**

* **Horizontal Scalability** 
  + **Movie Binge has been designed considering in the factor of database configuration. Hibernate allows easy adaption to different databases.**

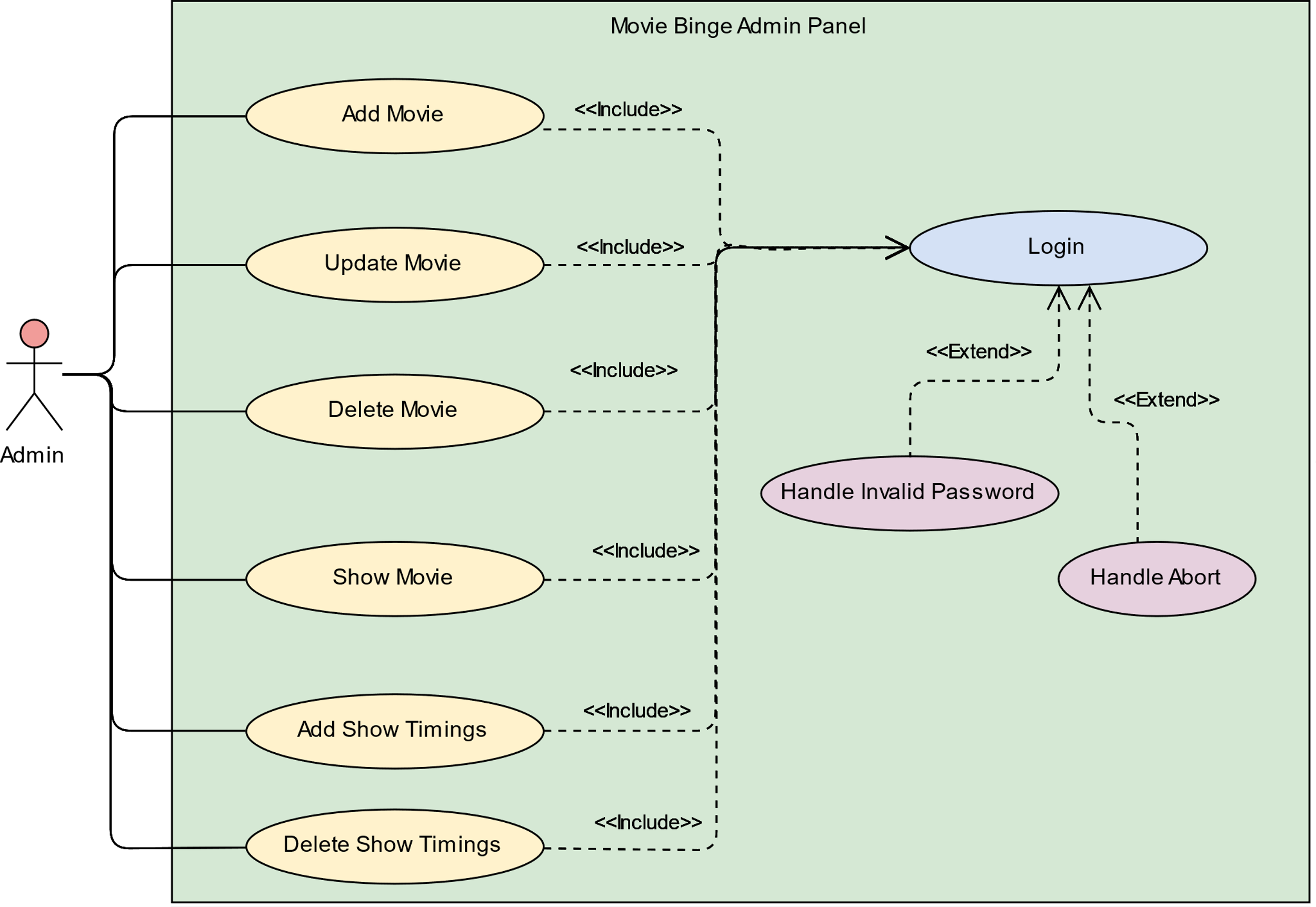
**Maintainability**

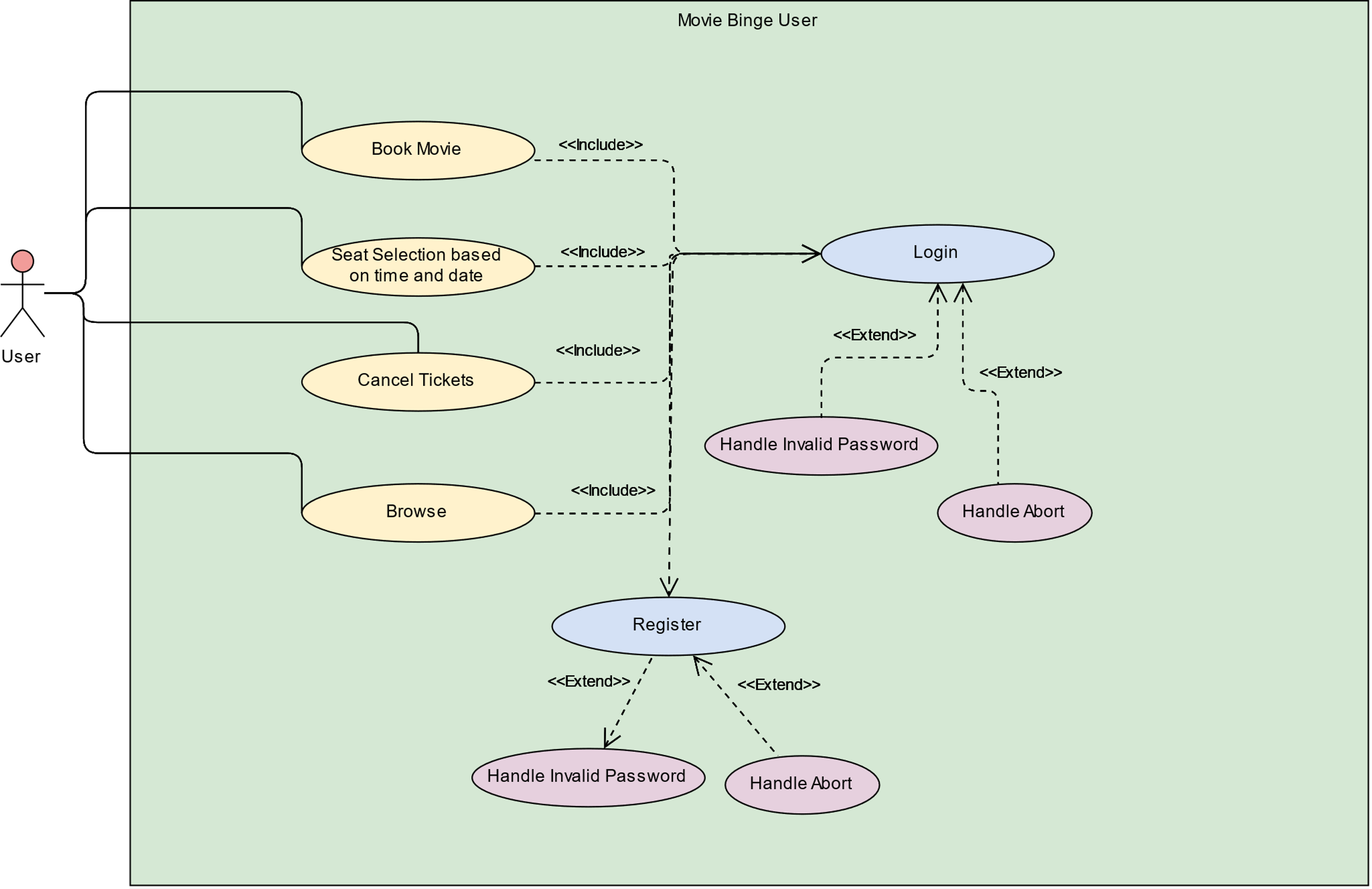
* **Anyone who can understand documentation can make changes to code according to requirements.**

**Reliability**

* **Covered possible cases and it catches exceptions in case of failure.**

**UML: USE CASE DIAGRAM FOR USER & ADMIN**

****

****

**USES CASES DETAIL**