**My\_MoviePlan**

**(Documentation)**

Project Name:- My MoviePlan

Project Developer :- Priya Naveen

Objectives of Project:-

* Create a dynamic and responsive web application for booking movie tickets online for different genres and languages.
* Sprint

One sprint taken

And its time period is 3 weeks 15 worling days

|  |  |
| --- | --- |
| ConFig | Value |
| Sprint Start | 15/05/2022 |
| Sprint End | 02/06/2022 |
| Daily Hours | 9 |
| Focus Factor | 95% |
| No. Weeks | 3 |
| Working Days | 15 |
| Work Hours | 135 |
| Available WH | 99 |

|  |  |  |
| --- | --- | --- |
| Name | Role | Contribution |
| Priya Naveen | FSD | 100% |
| Priya Naveen | QA | 100% |
| Priya Naveen | Deployment | 100% |

**Background of the problem statement:**

* NMS Cinemas is a chain of single screen theatres that screen movie shows of different genres and languages at very genuine prices.
* It was established in 2004 in Pune, India. Recently, the business analysts noticed a decline in sales since 2010.
* They found out that the online booking of movie tickets from apps, such as BookMyShow and Paytm were gaining more profit by eliminating middlemen from the equation.
* As a result, the team decided to hire a Full Stack developer to develop an online movie ticket booking web application with a rich and user-friendly interface.
* You are hired as the Full Stack Java developer and are asked to develop the web application.
* The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Story Name | Story Points | Hours For Fed | Hours For Bed | Hours For QA |
| As the Hr Manager I want | 13 | 30 | 2 | 22 |
| As a Manager | 13 | 70 |  |  |
| As a User | 8 |  | 2 | 20 |

**Features of the application:**

* Registration
* Login
* Payment gateway
* Searching
* Filtering
* Sorting
* Dynamic data
* Responsive and compatible with different devices

Algorithm:-

Step1:- Start

Step2:- Creating a Spring starter project on eccilpse.

Step3:- Create class MyMoviePlan.

Step3:- Create Controllers classes 1.Admin Controller.java 2.Booking controller.java

3.Customer Controller.java 4.Login Controller.java 5.Movies Controller.java

6.Screen Controller.java 7.SeatController.java 8. Show Controller.java

Step4:- create Pojo classes 1.Booking.java 2.Customer.java 3.Login.java

4.Movie.java 5.Screen.java 6.Seat.java 7.SeatStatus.java 8.Show.java

9.Theatre.java 10.Ticket.java 11.User.java

Step5:- create Interfaces Repo 1.AdminRepository.java 2.CategoryRepository.java

3.ProductRepository.java 4.PurchaseRepository.java 5.UserRepository

6.BookingRepository.java 7.CustomerRepository.java 8.IAdminRepository.java

Step6:- create Response clasess 1.CategoryResponse.java 2.ProductResponse .java

3.UserResponse.java

Step7:- create services 1.AdminService 2.Category Service 3.ProductServiece

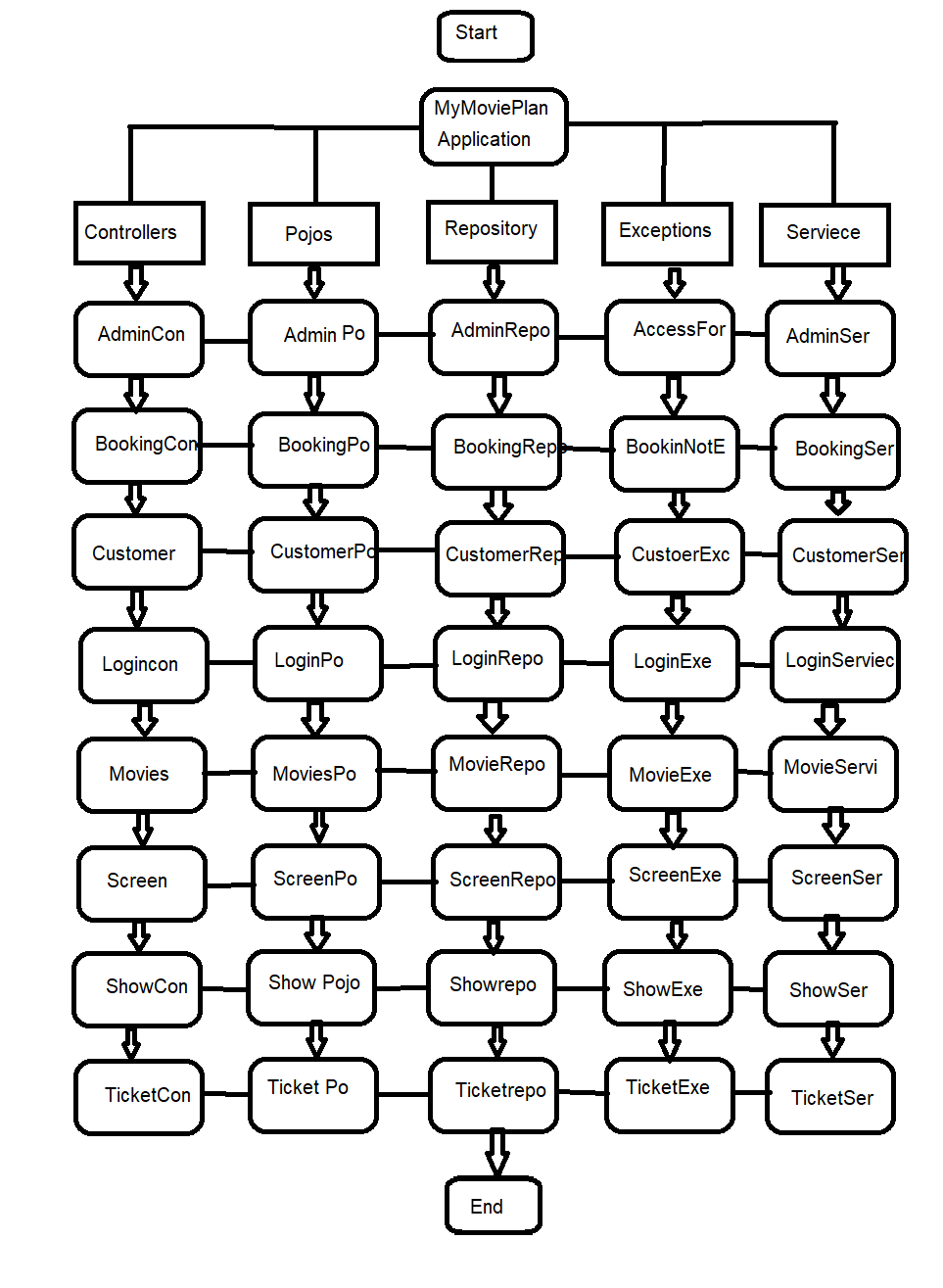
4.PurchaseServiece 5.UserServiece.

Step8:- Before building the project, we need to confirm that the servlet.jar has been added to the project.

Step9:- open Visual studio app, create app.

Step10:- write html , css, ts code here.

FlowChart:-





**Features :-**

**Admin Portal:**  
It deals with all the backend data generation and product information. The admin user should be able to:

* Add or remove different genres to or from the application to build a rich product line
* Edit movie details like name, ticket price, language, description, and show timings to keep it aligned to the current prices
* Enable or disable the movie shows from the application

**User Portal:**  
It deals with the user activities. The end-user should be able to:

* Sign-in to the application to maintain a record of activities
* Search for movie tickets based on the search keyword
* Apply filters and sort results based on different genres
* Add all the selected movie tickets to a cart and customize the purchase at the end
* Experience a seamless payment process
* Receive a booking summary page once the payment is complete

**Tools used for development**:

1. Eclipse IDE

2. MySQL Relation Database Management System

3. Apache Tom-Cat 10 server

4. Html, Css

5. Apache Maven

6. JDBC Java Database Connectivity

7.Visual studio

8.Database management: MySQL and Oracle.

9.Backend logic: Java programming, NodeJS

10.Frontend development: JSP, Angular, Bootstrap, HTML/CSS, and Javascript

11.Automation and testing technologies: Selenium, Jasmine, and TestNG

12.DevOps and production technologies: Git, GitHub, Jenkins, Docker, Kubernetes, and AWS

Unique selling Point:-

“Quality Matters here, The Next Quality ”