Ticket Booking Application

Gamage Danushka - AE9196

YTSP0300-3004- Application Framework

25th April 2024

Master in Full Stack Software Development

Instructor(s): Pasi Manninen

A black background with purple text

Description automatically generated

Contents

[1 Introduction 2](#_Toc164954298)

[2 Planning Documeation 2](#_Toc164954299)

[2.1 Purpose 2](#_Toc164954300)

[2.2 Scope 2](#_Toc164954301)

[2.3 Objectives 2](#_Toc164954302)

[2.4 Requirements 3](#_Toc164954303)

[2.4.1 Functional Requirements 3](#_Toc164954304)

[2.4.2 Non-functional Requirements 3](#_Toc164954305)

[2.5 Flow Chart 3](#_Toc164954306)

[2.6 ‌Functionality 4](#_Toc164954307)

[2.6.1 Admin User Functionality 4](#_Toc164954308)

[2.6.2 Normal user Functionality 4](#_Toc164954309)

[2.7 Architecture And Dependencies 5](#_Toc164954310)

[2.7.1 Design Architecture 5](#_Toc164954311)

[2.7.2 Dependencies 5](#_Toc164954312)

[2.8 Development Plan 5](#_Toc164954313)

[2.8.1 Key milestones 5](#_Toc164954314)

[2.8.2 Timeline 5](#_Toc164954315)

[2.9 Test Strategy 6](#_Toc164954316)

[3 Design Solution 7](#_Toc164954317)

[3.1 Solution Architecture 7](#_Toc164954318)

[3.2 Front-end Design Architecture 7](#_Toc164954319)

[3.3 Structure of the Application 7](#_Toc164954320)

[4 Time Management 8](#_Toc164954321)

[5 Developed Solution structure 8](#_Toc164954322)

[5.1 Login.js 9](#_Toc164954323)

[5.2 Admin.js 10](#_Toc164954324)

[5.3 Movies.js 11](#_Toc164954325)

[5.4 Usermanagement.js 12](#_Toc164954326)

[6 Additional Information 12](#_Toc164954327)

[7 Conclusion 13](#_Toc164954328)

# Introduction

In this documentation, Basically covered the Frontend architecture of the ticket booking application and in this part, I will focus on the front end level. In the front end level architecture design on the ReactJ. In the design application focu on the Ticket Booking application and it will ensure the basic functionality of the booking application has. Basically, React and Redux are used for frontend development and state management.

# Planning Documeation

## Purpose

The purpose of this document is to provide a comprehensive plan for developing a ticket booking application backend that allows users to register in the system, browse tickets, view additional information, and book tickets via the web platform.

## Scope

The scope of this application includes:

* Session planning
* Admin privilleges handling
* User registration and authentication
* Ticket browsing and searching
* Showtime and additional information search
* Ticket booking and edit history

## Objectives

* To provide a user-friendly interface for booking tickets.
* To integrate with user registration systems and show available ticket count.
* Handle the privileges from the application level.

## Requirements

### Functional Requirements

* Administrative privileges
* User registration and login
* Ticket listing with details
* Ticket scheduling
* Payment processing
* Booking confirmation and ticket generation

### Non-functional Requirements

* Security: Secure user data and payment information.
* Performance: Efficient loading and response times.
* Scalability: Able to handle many concurrent users.
* Usability: Intuitive and easy-to-use interface

## Flow Chart

A diagram of a company

Description automatically generated

## ‌Functionality

In the application has two specific user types;

1. Admin User
2. Normal User

### Admin User Functionality

In the application level, user must always register with as a normal user. But, after that existing can change the user as an administrator. In this case, admin user can do few specific things when comparing other users.

* Add/Delete the ticket information
* Change the booking information
* Delete and update the user information
* Also possible to work as a normal user

### Normal user Functionality

Normal users mean the users who visiting the application to check the ticket and booking the slots. In that case, they can simply work with the following functionality.

* Login/ sign up to the system
* Find out the relevant ticket information according to the define settings
* Check the ticket information according to selected ticket
* Book the ticket slot
* Update/ delete the already booked slot

## Architecture And Dependencies

### Design Architecture

* Languages : Reactjs

### Dependencies

* React
* Mui/Icon-materal
* Mui/materal

These are the required npm module dependencies that we can use for the design solution.

## Development Plan

### Key milestones

1. Requirement analysis and documentation
2. API level design and identify the type
3. Design the API inputs and outputs
4. Develop the API
5. Test the API and finalize the API endpoints

### Timeline

1. Week 03 - 04 : Requirement analysis and documentation
2. Week 05 : Discuss with supervisor and get approval for the project
3. Week 05-09: Learn about the React and Redux components and state transitions
4. Week 10-12: Design the components
5. Week 12-14 : Develop the component and state transitions
6. Week 14 -16 :Integrate with the backend
7. Week 17 : Test and finalize the documentation

## Test Strategy

The testing of the components is done as a manual test, and it will ensure application efficiency and customer-oriented design.

Conclusion

In this part, will describe the plan according to the frontend level of the application. The overall documentation related to planning it will ensure the features, requirements, and architecture according to design component design and integration with endpoints.

# Design Solution

## Solution Architecture

In the design application, frontend level is designed by using the ReactJS.

## Front-end Design Architecture

Basic Modules are like this:

1. Login
2. SignUp
3. Movie filter
4. MovieInfo (Show specific information according to selected movie)
5. Ticketbooking (edit/update)
6. Admin settings (User edit- update/delete, ticket edit-add/update/delete)

In the front-end level is design on for handle via using the React and it will test via using manual test.

## Structure of the Application

A diagram of a process flow

Description automatically generatedThe front end of the application is working like this.

In the design application, basically work as a monolithic based design and it will ensure the basic functionality that the ticket booking application has.

# Time Management

In the application level, I used the Git as the version control system, and it will show all the things like this.

Use: <https://github.com/Nandalochana/Application-Framework-CA-MAS-/tree/main/CA>

* Start date : 2024- January 29
* End date : 2024 April 26
* Total time : 88 Days
* Learning duration : 50 + days
* Development time : 30+ days

It takes some time to learn react. Because this is the first time that I have worked with ReactJS. I took more time to learn this. But I think I did my best to learn, and right now I have a good knowledge of this JS framework.

# Developed Solution structure

Here, I attached the basic react component codes and logic. But, If you need more, please check with : <https://github.com/Nandalochana/Application-Framework-CA-MAS-/tree/main/CA>

## Login.js

A screen shot of a computer program

Description automatically generated

## Admin.js

A screen shot of a computer screen

Description automatically generated

## Movies.js

A screen shot of a computer code

Description automatically generated

## Usermanagement.js

A computer screen shot of code

Description automatically generated

# Additional Information

In the frontend level, I used few libraries, and it will ensure the data security and smoothness of the application.

A screen shot of a computer program

Description automatically generated

# Conclusion

In the design application, the frontend level will focus on React. I use a few libraries to generate unique keys and transfer data as JSON. In the application development phase, more than 60% of the time was spent learning the framework and the redux state concept. This concept is new for me, and I think I achieved my target very well.

In this application, I learn the basic design principles related to React, and I think this will be a great opportunity to work on and learn React. My great experience was connecting the redux with multiple repetitive components in the ticket showing component, and I consumed more time in there. The hardest part is that. But other overall functionality developments are very easy, and I think this is a great chance for me to learn these concepts.

One of the drawbacks is that I couldn’t use alerts and notifications in here. Because I am more focused on the React framework. I can give the final grade as 4 (out of 5). Because of my hard work and research skills. As an application. The implemented functions are 100% fully working and finalized.

I think this assignment will be a turning point in my life.