**SOFTWARE DESIGN SPECIFICATIONS**

**PITCH FINDER**

# Prepared By Group 13

| **Elvis J** | **26** |  |
| --- | --- | --- |
| **Mohammed Ihsan** | **43** |  |
| **Mohammed Rizwan** | **45** |  |
| **Muhammed Rishan** | **48** |  |

# Contents

INTRODUCTION

DOCUMENT PURPOSE

PRODUCT SCOPE

INTENDED AUDIENCE AND DOCUMENT OVERVIEW

SYSTEM ARCHITECTURE DESIGN

THREE-TIER

PRESENTATION LAYER

APPLICATION LAYER

DATA LAYER

APPLICATION ARCHITECTURE DESIGN

DATABASE DESIGN

MOCKUI

TECHNOLOGY STACK

# Introduction

## This software design specification outlines the technical specifications and design decisions for the development of a sports venue booking mobile application using Flutter. The purpose of the application is to allow users to browse, book and pay for available sports venues such as soccer fields, basketball courts, and tennis courts.

## Document Purpose

The purpose of the Software Design Document is to provide a clear documented model of the designs of the application, that will provide following Designs of the project.

1.System architecture

2.Application architecture

3.MockUI

4.API Design

5.Database Design

## Product Scope

The sports venue booking mobile application will provide users with a platform to search, book and pay for sports venues in their local area. The application will target sports enthusiasts, individuals, and groups who are looking to book sports venues for recreational or competitive purposes.

## Intended Audience and Document Overview

Readers include the panel of professors and the team members consisting of developers and documentation writers. The document consist of description of all the various designs used in the projects. It has all the features that are to be implemented in the project.

## SYSTEM ARCHITECTURE DESIGN

### Description

The architecture consists of three main tiers. The Presentation Tier represents the front-end of the application that interacts with the user. The Presentation Tier will be developed using Flutter. The Application Tier represents the backend processing of the application. In this tier, we will implement the business logic and application processing using Firebase Functions. We will use Firebase Functions to handle tasks such as user authentication, data validation, and data processing. The Data Tier represents the database where the data is stored and retrieved. In this tier, we will use Firebase Realtime Database

**Architecture**Diagram

Description automatically generated

## APPLICATION ARCHITECTURE DESIGN

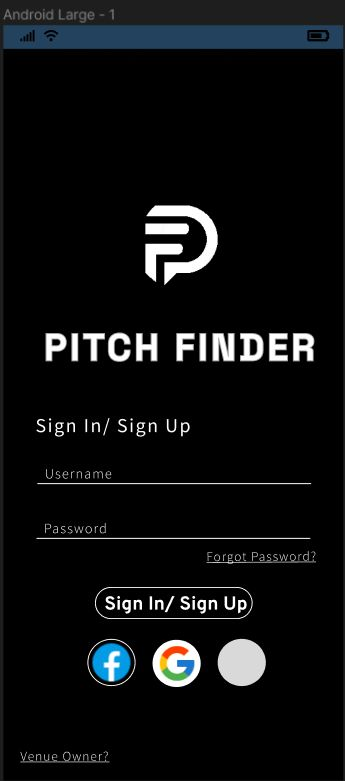
The Pitch Finder app will be a mobile sports venue booking application that can help sports enthusiasts find and book venues in their locality. The application architecture will consist of two main components: Front-end – Flutter and a back-end – Firebase Functions.

The frontend will handle the user interface and booking experience by ensuring responsiveness, performance and integration to backend.

The backend will handle venue management, booking management, user management, notifications and analytics.

Overall, the application architecture is designed to provide a friendly user experience for sports enthusiasts, utilizing the latest mobile app development technologies and API integration.

## GUI Design

1. Login Screen
2. Home Screen

Graphical user interface, application, website

Description automatically generated

1. Booking ScreensTable

   Description automatically generatedGraphical user interface, application

   Description automatically generated

## Technology Stack

Flutter: Flutter is an open-source mobile application development framework that is used to build high-performance, cross-platform applications. Flutter can be used to develop the front-end of the application, which includes the user interface, animations, and user interactions.

Firebase: Firebase is a cloud-based backend service that provides a range of features, including authentication, real-time database, cloud storage, and analytics. Firebase can be used to develop the backend of the sports venue booking application, which includes features such as user authentication, data storage, and real-time updates.

Google Maps API: The Google Maps API provides developers with a range of tools and services to integrate maps and location-based services into their applications. The Google Maps API can be used to provide users with the ability to view maps, search for venues, and get directions to the sports venues they want to book.

PayTM API: PayTM is a payment processing platform that can be integrated into the sports venue booking application to enable users to make payments securely and efficiently. The PayTM API provides developers with a range of tools to accept payments, manage subscriptions, and prevent fraud.

**IDE USED: ANDROID STUDIO**

**COMPONENT DESIGN**

1. User Management Component: This component will handle user registration, login, and authentication. It will also manage user profiles, which can be used to store user preferences, booking history, and other related information.
2. Venue Management Component: This component will manage data related to sports venues, including their location, facilities, availability, and pricing. It will also handle the process of booking venues, including checking for availability, reserving the venue, and managing payments.
3. Notification Component: This component will handle sending notifications to users about booking confirmations, cancellations, and reminders.
4. Payment Component: This component will handle the process of accepting payments from users and managing transactions related to bookings.
5. Search Component: This component will enable users to search for available venues based on criteria such as location, facilities, and price.
6. Map Component: This component will provide a visual representation of available venues on a map, enabling users to find nearby venues easily.
7. Review Component: This component will allow users to rate and review venues they have booked, providing valuable feedback to other users and venue owners.

**DATA DESIGN**

1. User Data: This data structure will store information about users, including their name, email, phone number, password, and booking history.
2. Venue Data: This data structure will store information about sports venues, including their name, address, facilities, availability, and pricing.
3. Booking Data: This data structure will store information about bookings made by users, including the date, time, venue, and payment details.
4. Review Data: This data structure will store information about reviews made by users, including the venue reviewed, the reviewer, and the rating.
5. Notification Data: This data structure will store information about notifications sent to users, including the type of notification, the recipient, and the content.

**Error Handling Design**

1. Error Logging: The application should log all errors, warnings, and exceptions that occur during its execution.
2. User-Friendly Error Messages: Whenever an error occurs, the application should display a user-friendly error message that describes the error in plain language and suggests possible solutions.
3. Input Validation: The application should validate all user inputs, including form data, search queries, and payment details, to ensure that they are in the correct format and within acceptable ranges/

**Performance Design**

1. Caching: The application should implement caching mechanisms to reduce the number of requests to the server and improve response times.
2. Compression: The application should compress all data that is transferred between the client and server, to reduce the amount of data that needs to be transmitted and improve response times.
3. Load Balancing: The application should be designed to scale horizontally by using load balancing techniques.
4. Database Optimization: The application should be optimized for database performance by using techniques such as indexing, query optimization, and database partitioning.

**Security Design**

1. Authentication and Authorization: The application should have robust authentication and authorization mechanisms to ensure that only authorized users can access the system. This can be achieved through password-based authentication, two-factor authentication, or integration with third-party authentication providers like Google, Facebook, etc.
2. Data encryption: Sensitive user data like passwords, payment information, and user profiles should be encrypted while in transit and at rest using strong encryption algorithms like AES, RSA, etc. This can prevent data theft and protect user privacy.
3. Role-based access control: The application should have a role-based access control mechanism to ensure that only authorized users can access specific system functionalities. This can be achieved by assigning roles like administrator, manager, or user and defining the privileges and permissions associated with each role.

**Testing Design**

1. Testing Approach: The testing approach for the Sports Venue Booking Application will be a combination of manual and automated testing. Manual testing will involve functional testing, integration testing, and user acceptance testing.
2. Testing Scope: The testing scope for the Sports Venue Booking Application will cover the following areas:
   1. Functional testing of all system functionalities including user registration, venue search, booking, payment, and cancellation.
   2. Integration testing to ensure that all system components work together seamlessly.
   3. Load testing to ensure that the system can handle high traffic and concurrent user requests.
3. Test Cases: The following are some examples of test cases that can be used to test the Sports Venue Booking Application:
   1. Verify that users can register and login to the system.
   2. Verify that users can search for venues based on location, date, and sport.
   3. Verify that users can view venue details including availability, price, and amenities.
   4. Verify that users can book a venue and make payment using a credit card or other payment methods.
   5. Verify that users can cancel a booking and receive a refund if applicable.
   6. Verify that the system can handle unexpected errors and exceptions gracefully.
4. Test Data: The test data for the Sports Venue Booking Application should include a range of test cases covering different scenarios, such as:
   1. Valid user registration data.
   2. Invalid user registration data (e.g., invalid email address, password, etc.).
   3. Test data for different sports and venues.
   4. Test data for different locations and dates.
   5. Test data for different payment methods.
   6. Test data for error and exception handling scenarios.

**Appendix A: Record of Changes**

| Version | Date | Author | Description of changes |
| --- | --- | --- | --- |
| 1.0 | 30/03/2023 | Rishan | Added System and Application Architecture |
| 1.0 | 30/03/2023 | Ihsan | Added Data Design |
| 1.0 | 02/04/2023 | Rishan | Started GUI Design and Testing Design.  Edited Component and Error Handling Design |
| 1.0 | 14/04/2023 | Elvis | Added Performance Design and Security Design |
| 1.0 | 15/04/2023 | Rizwan | Finalized Document for Submission |