Department of Computer Science and Engineering
T.K.M College of Engineering, Kollam
July 2023

PITCH FINDER: SPORTS VENUE BOOKING APPLICATION

Mini Project Report

Submitted By

MOHAMMED IHSAN NADEER(TKM20CS085)
MOHAMMED RIZWAN AHSRAF (TKM20CS087)
MUHAMMED RISHAN PALLIYAKATH (TKM20CS094)
ELVIS KOCHUMUTTOM(TKM20CS049)

to

APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of B.Tech Degree
in Computer Science and Engineering

DECLARATION

We undersigned hereby declare that the project report on “Pitch Finder: Sports Venue Booking Application”, submitted as part of our curriculum, Mini Project under APJ Abdul Kalam University, Kerala is a bonafide work done by us under supervision of Dr.Dimple A Shajahan, TKMCE.

This submission represents our ideas in our own words and from other sources that have been adequately and accurately cited and referenced. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission.

We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kollam
Date: 30/07/2022

Mr. Mohammed Ihsan Nadeer
Mr. Mohammed Rizwan Ashraf
Mr. Muhammed Rishan Palliyakath
Mr. Elvis Kochumuttom

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
T.K.M COLLEGE OF ENGINEERING, KOLLAM

CERTIFICATE

This is to certify that the report titled "**Sports Venue Booking Application**"submitted by **Mohammed Ihsan Nadeer**, **TKM20CS085**; **Mohammed Rizwan Ashraf**, **TKM20CS087**; **Muhammed Rishan Palliyakath**, **TKM20CS094**; **Elvis Kochumuttom**,**TKM20CS049** to the APJ Abdul Kalam Technological University in completion of the requirements for the award of Bachelor ofTechnology Degree in Computer Science and Engineering during 2023 – 2024 is a Bonafide record of the **Mini Project** work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for anypurpose.

Project Coordinators

Head of the Department

External Examiner

ACKNOWLEDGEMENT

We take this opportunity to express my deep sense of gratitude to the Almighty and sincere thanks to all who helped me to complete the project successfully.

We express our sincere gratitude to **Dr. T. A. Shahul Hameed**, Principal, TKMCE,for providing us with all the necessary facilities and support for doing the project.

We are extremely grateful to Dr. Dimple A Shajahan , Project Coordinator and Professor, Department of Computer Science and Engineering and Prof. Reena George ,Assistant Professor, Department of Computer Science and Engineering, Prof.Nisa A K, Assistant Professor, Department of Computer Science and Engineering and Prof. Jesna J S ,Assistant Professor, Department of Computer Science and Engineering for their constructive guidance, advice, constant support and technical guidance provided throughout the making of this project. Without their intellectual support and apt suggestions at the perfect time, this project work would not be possible.

We extend our immense gratitude to all Faculties and Technical Staffs in the Department of Computer Science and Engineering, for their help and necessary facilities to complete the project. Our humble gratitude and heartiest thanks also go to our parents and friends, who have supported and helped us on the course of this work.

Mr. Mohammed Ihsan NadeerMr.
Mohammed Rizwan Ashraf Mr.
Muhammed Rishan Palliyakath
Mr. Elvis Kochumuttom

ABSTRACT

Sports Venue Booking Application was developed as part of the Mini Project course under Kerala Technical University. The "Pitch Finder" application is a comprehensive sports venue booking solution designed to cater to sports enthusiasts and venue owners. This mobile app aims to simplify the process of finding, booking, and managing sports venues, making it convenient for users to secure their preferred locations for various sports activities. With user-friendly interfaces and robust functionality, the app facilitates seamless interactions between users and venue owners.

The app's primary goal is to bridge the gap between sports enthusiasts in search of suitable venues and venue owners looking to optimize their facility bookings. Users can easily search for available sports venues based on location, sport type, date, and time. They can review detailed venue information, including amenities, pricing, and photos, to make informed decisions. The booking process is streamlined, allowing users to reserve their desired sports venues effortlessly. Secure payment integration ensures smooth and reliable transactions.

For venue owners, the application offers an intuitive dashboard where they can list their sports venues, manage availability, update pricing, and view booking history. This platform enables venue owners to enhance their visibility and reach a wider audience of potential customers.

The app incorporates essential security measures, including authentication and data encryption, to protect user information and payment details. It also ensures data integrity through input validation and secure data storage.

Developed using the Flutter framework, the "Pitch Finder" app is designed to provide a consistent and native user experience on both iOS and Android platforms. The frontend interfaces are designed to be user-friendly and intuitive, offering smooth navigation and interactive features.

The application encompasses a robust backend architecture that facilitates efficient communication between users, venue listings, and booking data. APIs for user authentication, and venue management are seamlessly integrated to ensure a cohesive user experience.

In conclusion, the "Pitch Finder" application addresses the needs of both sports enthusiasts and venue owners by offering a comprehensive and user-centric platform for sports venue discovery and booking. Its intuitive design, security measures, and seamless functionality make it a valuable tool for anyone seeking sports facilities for recreational activities or events.

CONTENT

1. INTRODUCTION

The rapid growth and popularity of sports and recreational activities have created a high demand for convenient and efficient ways to book sports venues. To address this need, we present PitchFinder, a sports venue booking application designed to simplify the process of finding and reserving sports facilities. PitchFinder aims to provide users with a user-friendly platform that offers a wide range of sports venues, along with a seamless booking experience.

2. OBJECTIVES

The main objectives of the PitchFinder project are as follows:

1.4.1 To provide a user-friendly interface: The application will offer a simple and intuitive user interface, ensuring that users can easily navigate through the platform, search for venues, and make reservations.

1.4.2 To streamline the booking process: PitchFinder will automate the venue booking process, eliminating the need for manual inquiries and negotiations. Users will be able to check real-time availability, receive instant confirmation of bookings, and make secure payments.

1.4.3 To increase accessibility and convenience: The application will enhance the accessibility of sports venues by providing a centralized platform where users can explore a wide range of options. It will enable users to find venues based on their specific requirements, such as location, sports type, and available amenities.

1.4.4 To facilitate effective communication: PitchFinder will foster efficient communication between users and venue owners. It will provide a messaging system or contact information, allowing users to inquire about details, negotiate terms, and address any concerns they may have.

1.4.5 To support venue owners: The application will empower venue owners by providing them with a platform to showcase their facilities, manage bookings, and maximize their business potential. It will assist them in effectively marketing their venues and reaching a broader customer base.

By achieving these objectives, PitchFinder aims to revolutionize the sports venue booking experience, offering convenience, accessibility, and efficiency to users and venue owners alike.

3. ORGANISATION OF THE REPORT

The remainder of the report is laid out as follows. As part 4, details of the software model used in the project are described. Part 5 describes the Software Development life cycle using the Waterfall model. It shows the flow of project building through different phases of the software model. The phases include Requirement analysis, System design, Implementation, Testing, Deploy and Maintenance. Part 6 concludes with some thoughts on the work’s future potential and the references used for the report.

4. SOFTWARE MODEL USED

Software Development Life Cycle is broadly categorized into the Waterfall model and Agile module.

In a *waterfall model*, each phase must be completed before the next phase can begin and there is no overlapping in the phases. It is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow.

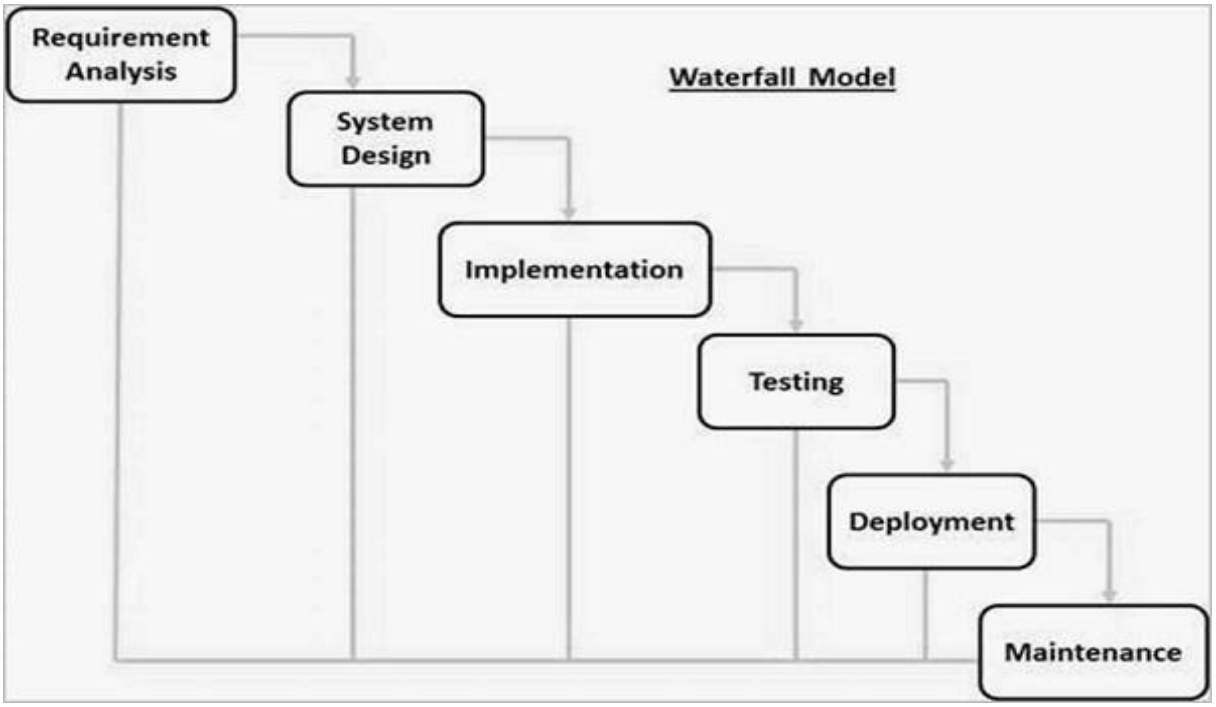


Figure 4.i: Waterfall model

- **Requirement Gathering and analysis**−All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design**−The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation**−With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing**−All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration, the entire system istested for any faults and failures.

- **Deployment of system**−Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance**−There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions have been released. Maintenance is done to deliver these changes in the customer environment.

The Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided into time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

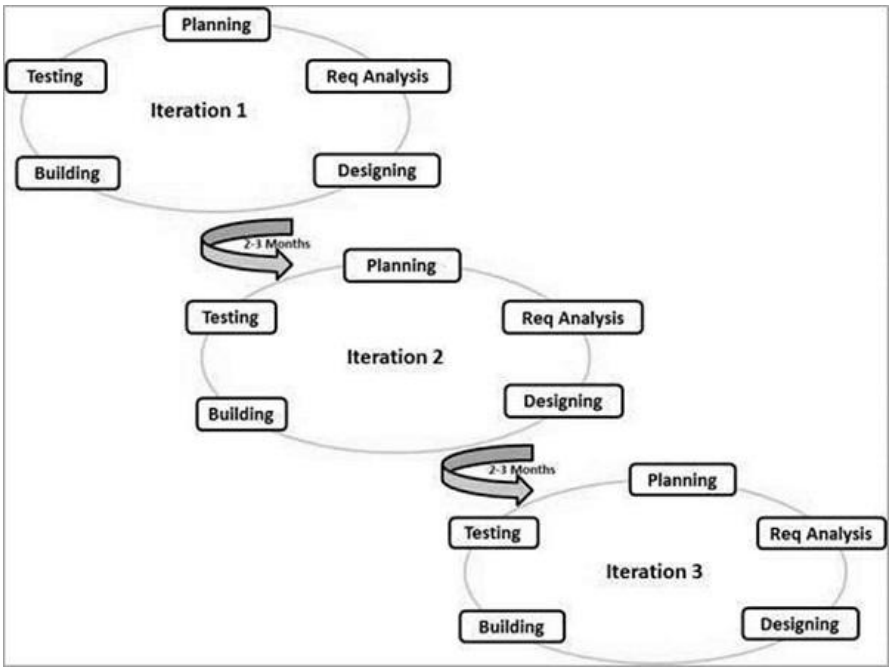


Figure 4.ii: Agile model

Following are the Agile Manifesto principles−

- **Individuals and interactions**−In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software**−Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- **Customer collaboration**−As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.

- **Responding to change**−Agile Development is focused on quick responses to change and continuous development.

“Sports Venue Booking Application” initially used the Waterfall model approach as it was important to define the requirements of the mobile application. Before the implementation of the project, a detailed survey had to be performed before deciding the tech-stack to be used. However, with course of time, the Waterfall model was converted to Agile model as it was necessary to test the prototype during the implementation stage. A few of the requirements like changing image captioning to object detection, were redefined during the process of implementation. Unit testing was performed after implementing each feature. Therefore, the project uses a waterfall model and in depth it uses the Agile Model.

5. SOFTWARE DEVELOPMENT LIFE CYCLE:WATERFALL MODEL PHASES

5.1 PHASE 1: REQUIREMENT ANALYSIS

Requirement analysis is a crucial phase in the software development process that involves gathering and defining the functional and non-functional requirements of a system. In the context of the "Pitch Finder" sports venue booking app, requirement analysis would involve identifying and documenting the specific needs and features of the application. Here's an overview of the requirement analysis for the app:

5.1.1 Functional Requirements Definitions

Functional requirements define the specific functionalities and features that the PitchFinder sports venue booking application should possess. These requirements are derived from the needs of the users and the objectives of the project. The following functional requirements are identified for PitchFinder:

5.1.1.1 User Registration and Authentication
- Users should be able to create an account and provide necessary information for registration.
- The application should have a secure authentication system to verify user identities.

5..1.1.2 Venue Search and Filtering
- Users should be able to search for sports venues based on location, sports type, amenities, availability, and other relevant criteria.
- The application should provide advanced filtering options to refine search results based on user preferences.

5.1.1.3 Booking and Reservation Management
- Users should be able to select a desired sports venue and make a booking request.
- Users should have the ability to view their booking.

5.1.1.4 User Reviews and Ratings
- Users should be able to provide reviews and ratings for sports venues they have booked.

5.1.1.5 Admin Panel
- An admin panel should be available to manage the overall system, including user management, venue verification, and handling reported issues.

5.1.2 Nonfunctional Requirements Definitions

Nonfunctional requirements describe the qualities and constraints that govern the behavior of the PitchFinder application. These requirements focus on aspects such as performance, security, usability, and scalability. The following nonfunctional requirements are identified for PitchFinder:

5.1.2.1 Performance
- The application should have low latency and response times to ensure a smooth and seamless user experience.
- It should be capable of handling multiple concurrent users and large amounts of data without significant performance degradation.

5.1.2.2 Security
- The application should implement robust security measures to protect user data and transactions.
- User authentication and authorization should be secure to prevent unauthorized access to user accounts and sensitive information.

5.1.2.3 Usability
- The user interface should be intuitive, visually appealing, and easy to navigate.
- Users should be able to perform tasks efficiently without confusion or frustration.

5.1.2.4 Reliability
- The application should always be reliable and available for users, minimizing downtime and disruptions.
- Data backup and recovery mechanisms should be in place to prevent data loss.

5.1.2.5 Scalability
- The application should be designed to handle increasing user demands and accommodate a growing number of sports venues and bookings.
- It should be scalable to support potential future enhancements and feature additions.

5.1.2.6 Compatibility
- The application should be compatible with different devices and operating systems, allowing users to access it from desktops, laptops, and mobile devices.

5.1.2.7 Localization and Internationalization
- The application should support multiple languages and currencies to cater to users from various regions.

5.2 PHASE 2: SYSTEM DESIGN

System design involves creating a comprehensive plan for how the various components of a software application will work together to meet the specified requirements. In the context of the "Pitch Finder" sports venue booking app, the system design would encompass the architecture, components, interactions, and technologies used to build the application. Here's an overview of the system design for the app:

System Architecture:
The app can follow a client-server architecture. The client side would consist of the mobile app developed using the Flutter framework, and the server side would involve backend services, APIs, and databases.

Components and Modules:
Frontend: Flutter and Android Studio
Backend: Django
Database: The storage system (MySQL) to store user data, venue information, bookings, and reviews.

UI/UX DESIGN: WIREFRAME

The following are the final wireframes made based on the functional features:

Figure 5.2.1.i: Login Page

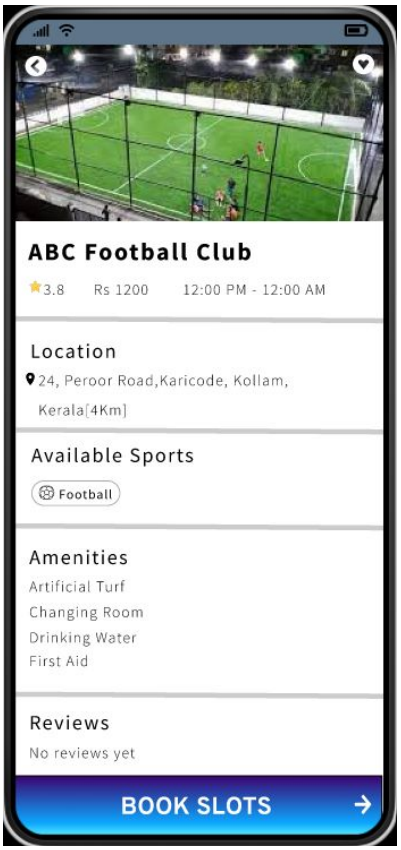


Figure 5.2.1.ii: Venue page

Figure 5.2.1.iii: Booking page

5.2.1 UX IMPLEMENTATION

The app has mainly 5 windows:

- Login Page
- Venue page
- Booking page
- User registration page
- Manager registration page

5.2.2.1: The Venu page will display venues based on location, rating, and price.

5.2.2.2: Users can view the details for each venue.

5.2.2.3: Once a venue is decided, users must select a slot.

5.2.2.4: Managers can register their venue in the app. They must update details of venue, slot available, and slots booked.

5.2.2 FINAL DESIGN

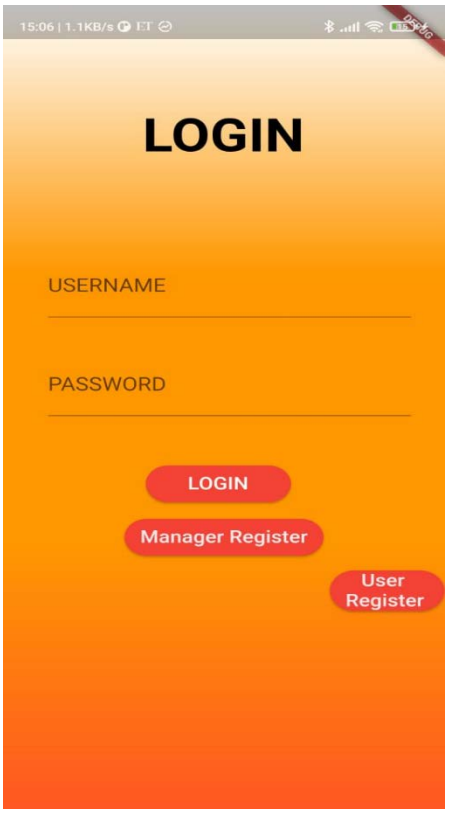


Figure 4.2.4.i:Login Page

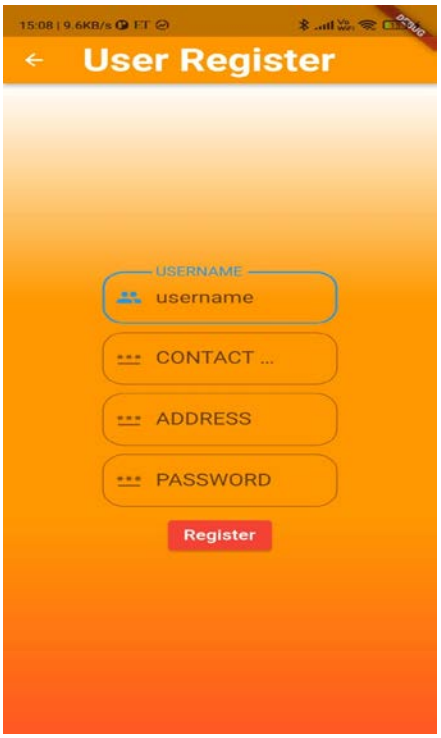
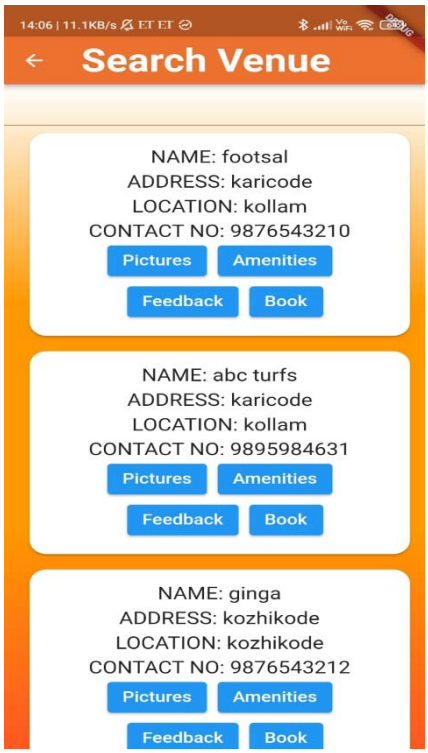
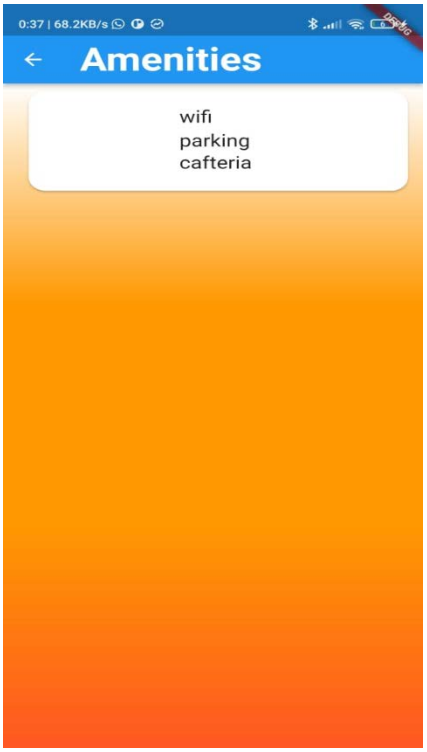


Figure 4.2.4.ii: User Registration page





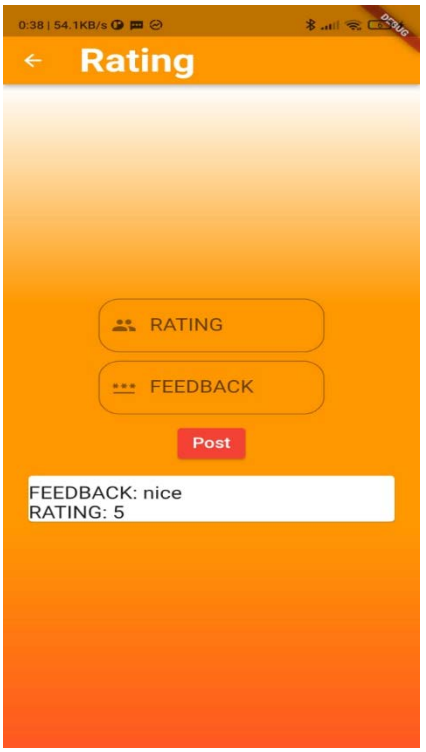
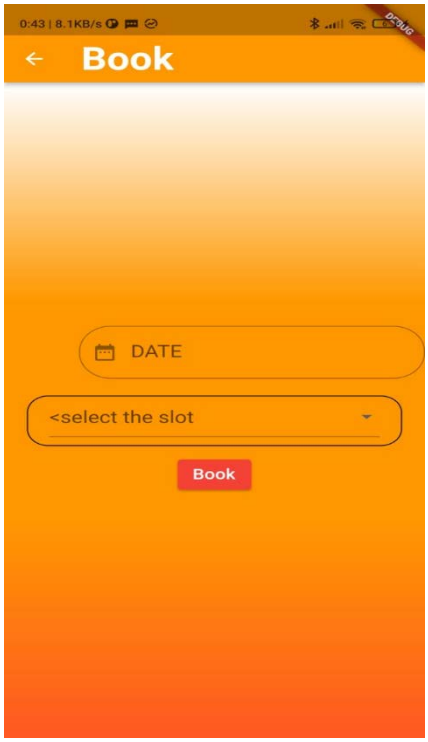
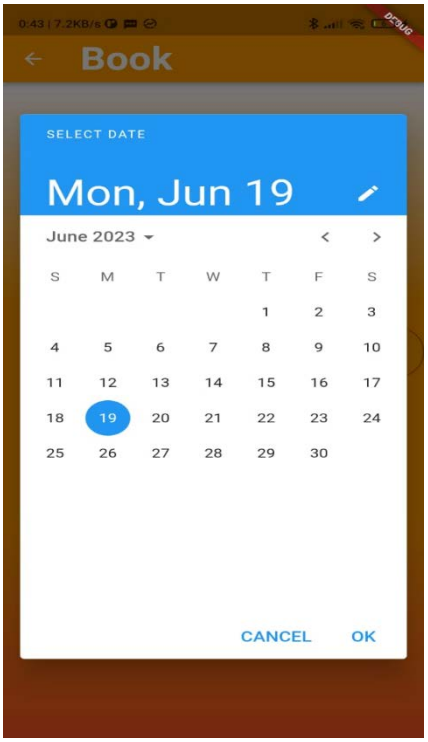


Figure 4.2.4.iii: Venue Page





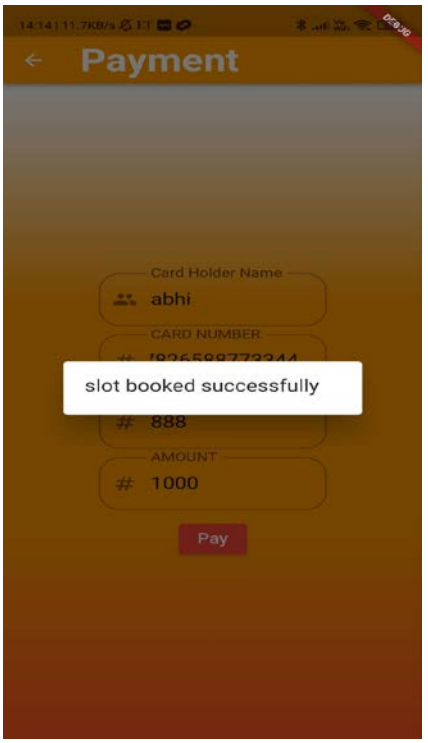
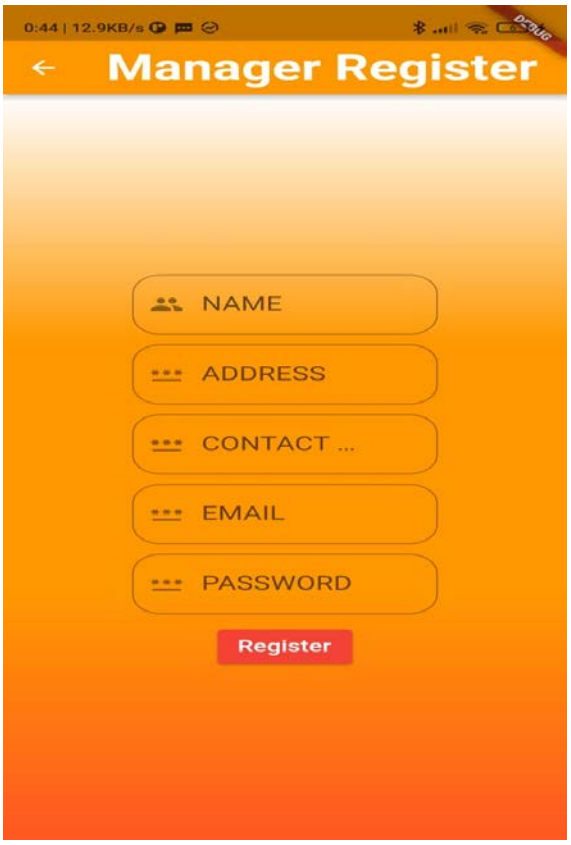
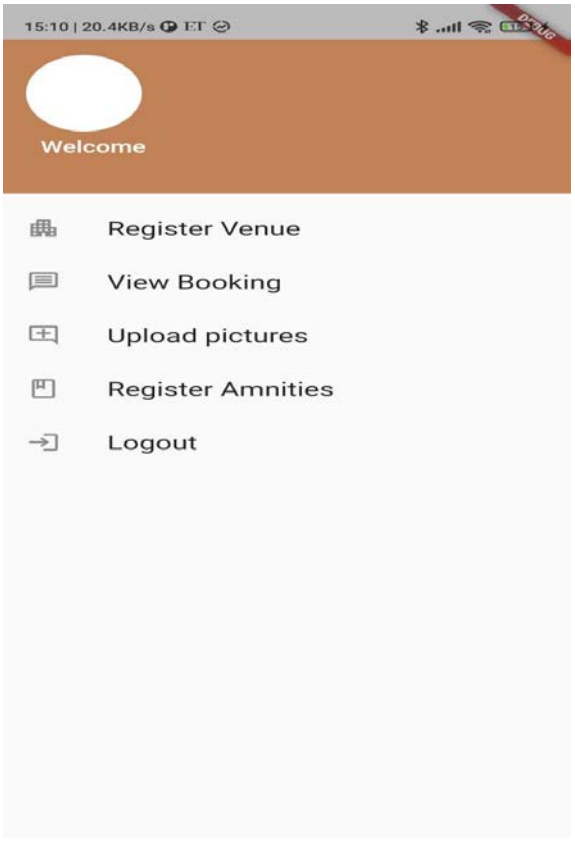


Figure 4.2.4.iv: Booking page



4.2.4.v: Manager Registration page



4.2.4.vi:Manager Dashboard Page

5.3 PHASE 3: IMPLEMENTATION

The implementation phase is where the project team does the project work to produce the deliverables. The word “deliverable” means anything the project delivers. The deliverables for the project include all of the products or services that the team is performing for the client, customer, or sponsor, including all the project management documents put together. The steps undertaken to build each deliverable will vary depending on the type of project undertaken.

Sports Venue Booking Application is an app for sports enthusiasts. Its main deliverables are:

- Fast performance
- Compatible with any android version
- Reliable
- Simple UI/UX
- Less device storage
- Location enabled.

The tech-stack used for implementing the project are:

5.3.1 Frontend

Flutter 3.0.1 (Dart 2.17.1)

Android Studio Chipmunk | 2021.2.1

5.3.2 Backend

Django 4.2.4 (LTS)

The whole project was subdivided based on mobile app phases. The sub-divisions werethen implemented by the respected team members.

- Login Page – Muhammed Rishan
- Venue page – Mohammed Ihsan
- Booking page – Mohammed Ihsan
- User registration page – Muhammed Rishan
- Manager registration page – Elvis Kochumuttom

5.4 PHASE 4: TESTING

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer Software testing definition as a White Box and Black Box Testing. In simple terms, Software Testing means the Verification of Application Under Test (AUT).

Software Testing is important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software products ensure reliability, security, and high performance which further results in time saving, cost effectiveness and customer satisfaction.

Benefits of software testing include:

- **Cost-effective:** It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money in the long term. In case the bugs caught in the earlier stage of software testing, it costs less to fix.
- **Security:** It is the most vulnerable and sensitive benefit of software testing. Peopleare looking for trusted products. It helps in removing risks and problems earlier.
- **Product quality:** It is an essential requirement of any software product. Testingensures a quality product is delivered to customers.
- **Customer satisfaction:** The main aim of any product is to give satisfaction to theircustomers. UI/UX Testing ensures the best user experience.

Testing is broadly categorized as follows,

1. Functional testing
2. Non-functional testing
3. Maintenance testing

5.4.1 Functional Testing

5.4.1.1 User Registration and Login

- Test Scenario: Verify that new users can register successfully and existing users can log in with their credentials.
- Test Cases:
    - Test new user registration with valid and invalid details.
    - Test user login using correct and incorrect credentials.
    - Test password recovery functionality.

5.4.1.2 Venue Booking

- Test Scenario: Validate the process of booking sports venues for specific dates and times.
- Test Cases:
    - Test successful venue booking for different sports activities.
    - Test the handling of overlapping bookings and time slots.

5.4.1.3 Venue Availability

- Test Scenario: Ensure that users can view the availability of sports venues for specific dates and times.
- Test Cases:
    - Test accurate display of venue availability information.
    - Test the display of unavailable venues for booked slots.
    - Test the real-time update of venue availability.

5.4.1.4 Error Handling

- Test Scenario: Validate the application's ability to handle errors gracefully.
- Test Cases:
    - Test error messages and their appropriateness for different scenarios.
    - Test the application's behavior when encountering unexpected errors.

5.4.2 Non-functional Testing

Under Non-functional Testing, parameters like, performance test, loading capacity, efficiency and portability were tested. Following are the inferences after performing the tests.

5.4.1 Performance Testing

- Load Testing: Evaluated the application's response time and stability under different user loads.

5.4.2 Usability Testing

- Conducted usability testing sessions with potential end-users to gather feedback on the application's user interface and overall user experience.

5.4.3 Security Testing

- Reviewed the application's security measures to identify vulnerabilities and ensure data protection.

5.4.4 Compatibility Testing

- Tested the application on various devices (smartphones and tablets) and operating systems (iOS and Android) to ensure cross-platform compatibility.

5.4.3 Maintenance Testing

Under Maintenance testing, the mobile app was installed in android phones of different companies to check the *overflow condition* of the widgets due to different dimensions of the screen.

5.5 PHASE 5: DEPLOYMENT AND MAINTENANCE

5.5.1 DEPLOYMENT

*SDLC Deployment Phas***e** provides for production installation and customer acceptance for the software, requiring all test cases to verify successful software execution, completeness, and correctness. It's an essential phase in *mobile app development*.

This phase includes the work necessary to deploy the final solution into the target production environments. In addition to it, other features include creating guides for installation, system operations, system administration, and end-user functionality. Besides, you need to create a detailed plan for implementing a solution across the organization.

5.5.1.1 Objectives – SDLC Deployment Phase

It is the final phase of the software development life cycle (SDLC), putting the product into consideration. Once the project team tests the product and passes each testing phase, it is ready to go live. Thus, the product is ready for use in the real environment by all the product's end-users.

There are various phases in the deployment process that the project team must follow to ensure that the code and the technology are deployed appropriately. It includes deployment preparation and procedures, product deployment, transferring product ownership, and closing the deployment phase.

5.5.1.2 Tasks and Activities – SDLC Deployment Phase

The tasks and activities in the SDLC Deployment Phase are associated with specific deliverables. They are performed depending on the nature of the project.

5.5.1.2.1 Communicate New Deployment to the Users

The notice must include:

- The deployment schedule.
- A brief description of the benefits of the new system.
- The difference between the old and the new system.

- The responsibilities of the end-user affected by the deployed changes.
- The process of obtaining technical support, including phone numbers and contact numbers.

5.5.1.2.2 Execute the Training Plan

It ensures that you are executing the desired training plan.

5.5.1.2.3 Perform the Data Entry or Conversion

It makes sure that you are performing the data entry or the necessary conversion if required.

5.5.1.2.4 Install the System

It ensures that the system is fully operational and installs the system in a production house.

5.5.1.2.5 Post-Deployment Review

It documents the deployment experiences, recommends system enhancements, and guides future projects.

5.5.1.2.6 Revise Previous Documentations

All the relevant documents must be reviewed and updated, considering all the changes introduced by the project. For example, report documentation must always reflect the current state of the report.

5.5.1.3 Pitch Finder Deployment Phase

Currently the app is hosted on GitHub. Sports Venue Booking Application is supported by Android phones and can easily be deployed in the Play Store, from where the users can download and use it for their personal needs. A detailed description of the app along with a tutorial will be made available in the Play Store.

Benefits of the app:

- Smooth UI/UX
- User friendly

Advantage of the app:

- Targets multiple user population.
- Light weight and loads quickly.

5.5.2 MAINTENANCE

The maintenance phase of the SDLC occurs after the product is in full operation. Maintenance of software can include software upgrades, repairs, and fixes of thesoftware if it breaks.

Software applications often need to be upgraded or integrated with new systems the customer deploys. During the maintenance phase, errors or defects may exist, which would require repairs during additional testing of the software. Monitoring the performance of the software is also included during the maintenance phase.

6. CONCLUSION AND FUTURE SCOPE

In the coming versions, Pitch Finder will have a feature to make teams and invite people to fill in. A tournament feature is also a feature that could be added. In the real world the app can be easily downloaded from the Play Store.

This project has helped the team members to be considerate about the whole population and become inclusive to all. It motivated us to think about the difficulties faced by sports enthusiasts who have to go through the hassle of calling each venue to ask for slots.

PROJECT DEPENDENCIES

- https://flutter.dev/
- https://pub.dev/packages/speech_to_text
- https://pub.dev/packages/text_to_speech
- https://pub.dev/packages/tflite_flutter
- https://pub.dev/packages/tflite_flutter_helper

DOCUMENTATION

- https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document
- https://proofreadingmalaysia.com/five-elements-to-include-in-your-abstract/
- https://www.guru99.com/software-development-life-cycle-tutorial.html#8
- https://www.tutorialspoint.com/sdlc/sdlc_overview.htm

Software Requirements Specification

for

Sports Venue Booking Application

Prepared by Rishan
Ihsan
Elvis
Rizwan

13/03/2023

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the Sports Venue Booking Application is to provide a user-friendly and efficient platform for users to book sports venues in their locality. The application aims to simplify the process of booking sports venues and provide a solution that is easy to use, convenient, and accessible.

1.2 Document Conventions

This document follows the IEEE Std 830-1998 standard for software requirements specification. The document includes functional and non-functional requirements, project plan, timeline, and team details.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document includes the development team, project stakeholders, and any other individuals interested in the Sports Venue Booking Application. The document can be read by anyone who is familiar with software development and has a basic understanding of web technologies.

1.4 Project Scope

The Sports Venue Booking Application mobile app will be a platform that allows users to search for, book, and pay for sports venues in their locality using their mobile devices. The application will have a user management module, venue management module, booking management module, and payment management module. The application will be accessible on both iOS and Android mobile devices and will be developed using modern mobile app development technologies. The scope of the project will be limited to the development of the mobile app only, and it will not include the development of a web-based platform.

1.5 References

1. IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications".

2. OpenSports: https://opensports.net/
3. Playfinder: https://www.playfinder.com/
4. SportEasy: https://www.sporteasy.net/.>

2. Overall Description

2.1 Product Perspective

The Sports Venue Booking Application is a standalone mobile application that allows users to search for, book, and pay for sports venues in their locality. The application will be developed as a native mobile app for both iOS and Android platforms and will provide a user-friendly and efficient interface for users to easily search and book sports venues.

2.2 Product Features

1. User registration and login
2. Venue search and filtering
3. Venue booking and payment
4. Venue management
5. User profile management
6. Team creation
7. Player requests
8. Tournament creation
9. Leagues
10. Push notifications and reminders
11. Review and rating system for venues

2.3 User Classes and Characteristics

The application will have two types of users:

1. Customers - Sports enthusiasts who want to book a sports venue for personal use.
2. Venue Owners - Individuals or organizations who own sports venues and want to list their venues on the platform.

2.4 Operating Environment

The application will be developed as a native mobile app for both iOS and Android platforms. The app will require an internet connection to function and will be compatible with the latest mobile operating systems.

2.5 Design and Implementation Constraints

The application will be developed using flutter and will be optimized for performance and user experience. The design will be based on the latest UI/UX trends and will provide an intuitive and user-friendly interface.

2.6 Assumptions and Dependencies

The success of the application depends on the availability of accurate and up-to-date venue information. The application will rely on venue owners to provide accurate information about their venues, such as location, availability, pricing, and amenities. The app will also depend on payment gateway integrations for payment processing.

3. System Features

- **User Registration and Login:** Users can create an account and log in to access the app's features.

- **Venue Search and Filtering:** Users can search for sports venues based on location, availability, pricing, and amenities. They can filter search results based on their preferences.

- **Venue Booking and Payment:** Users can book a sports venue and make payment using the integrated payment gateway.

- **Venue Management:** Venue owners can manage their venue listings, including updating venue information, setting availability, and managing bookings.

- **User Profile Management:** Users can manage their profiles, including updating personal information, viewing booking history, and leaving reviews and ratings.

- **Friend Request:** Users can send and accept friend requests.

- **Team Creation:** Users can upload their team details and can request for a matchup.

- **Player Requests:** Users can request for available players to join them.

- **Push Notifications and Reminders:** Users can receive push notifications and reminders for upcoming bookings and other important information.

- **Review and Rating System:** Users can leave reviews and ratings for venues they have booked, providing valuable feedback to other users.

- **Secure Payment Gateway Integration:** The app will be integrated with a secure payment gateway to ensure safe and reliable payment processing.

- **Multilingual Support:** The app will support multiple languages to cater to a wider audience.
- **24/7 Customer Support:** The app will provide round-the-clock customer support to users, addressing any issues or concerns they may have.

4. External Interface Requirements

4.1 User Interfaces

- **Registration and Login Screens**: The user can create a new account or log in to an existing account.
- **Venue Search Screen:** Users can search for sports venues and apply filters to refine their search.
- **Venue Details Screen:** Users can view detailed information about a venue, including location, amenities, pricing, and availability.
- **Booking Screen:** Users can book a venue by selecting a date and time and making payment.
- **User Profile Screen:** Users can manage their profile information, view booking history, and leave reviews and ratings.
- **Venue Owner Dashboard:** Venue owners can manage their venue listings, including updating venue information, setting availability, and managing bookings.

4.2 Hardware Interfaces

The app will be compatible with smartphones and tablets running on iOS and Android platforms.

4.3 Software Interfaces

- The app will be built using the Flutter framework and the Dart programming language.

- The app will integrate with third-party services, including payment gateways and location-based services.

4.4 Communications Interfaces

- The app will communicate with the server through APIs to fetch venue information, manage bookings, and process payments.
- The app will use push notifications to alert users of upcoming bookings and other important information.
- The app will provide customer support through chat or email communication.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The app should load quickly, and all features should respond within two seconds.
- The app should be responsive and provide seamless user experience on low-end devices.
- The app should be able to handle multiple requests simultaneously without crashing.
- The app should have efficient data storage and retrieval mechanisms to handle a large volume of data.

5.2 Safety Requirements

- The app should not allow users to book a venue that is already occupied by another user at the same time.
- The app should provide necessary safety instructions for users, especially in case of outdoor venues.

- The app should ensure that all venues listed are safe and meet the necessary safety requirements.

5.3 Security Requirements

- The app should be secure and protect user data from unauthorized access.
- The app should use encryption mechanisms to secure sensitive user data, such as payment information.
- The app should implement secure login mechanisms, such as two-factor authentication, to prevent unauthorized access to user accounts.
- The app should have a mechanism to handle security breaches and data breaches and alert users if necessary.

5.4 Software Quality Attributes

- The app should be easy to use and intuitive, with a user-friendly interface.
- The app should be reliable and provide consistent performance.
- The app should be scalable and able to handle a growing number of users and bookings.
- The app should be maintainable and easy to update with new features or bug fixes.
- The app should be portable and able to run on different platforms without any significant changes.

6. Other Requirements

- Compatibility Requirements: The app should be compatible with the latest versions of iOS and Android operating systems.

- Usability Requirements: The app should have a user-friendly interface, with clear navigation and easy-to-understand instructions.
- Performance Requirements: The app should be fast and responsive, with minimal loading times for features and functions.
- Testing Requirements: The app should undergo rigorous testing to ensure that all features and functions work as expected and that the app is free from bugs and errors.
- Maintenance Requirements: The app should be easy to maintain and update, with regular bug fixes and feature enhancements.
- Documentation Requirements: The app should have comprehensive documentation, including user manuals, developer guides, and system specifications.

Appendix A: Glossary

- Sports Venue: A location where sporting events or activities take place, such as a stadium, gym, or field
- Booking: The process of reserving a sports venue for a specific date and time.
- User: An individual who uses the Sports Venue Booking Application to search and book sports venues.
- Venue Owner: The owner or manager of a sports venue who lists their venue on the Sports Venue Booking Application.
- API: Application Programming Interface. A set of protocols and tools used to build software applications and communicate between different software systems.
- Encryption: The process of converting plain text data into a coded message to prevent unauthorized access.

- Push Notification: A message that is sent from a server to a mobile device, even when the app is not open.
- UX/UI: User experience/User Interface.