# A Project Report
# On
# "FUTSA"



### Submitted To
## School of Engineering
## Faculty of Science and Technology
# POKHARA UNIVERSITY

In the Partial Fulfillment of the requirements for the degree of Bachelor in Software Engineering of the Pokhara University

## Supervised By
## Er. Bhesh Bahadur Thapa

### Submitted By
Saroj Aryal [19180079]

Suraj Gautam  [19180089]

Amrit Adhikari [19180047]

Chiranjibi Ranabhat [19180056]

# (September 2023)

# "FUTSA"

**Submitted By**

Saroj Aryal [19180079]

Suraj Gautam  [19180089]

Amrit Adhikari [19180047]

Chiranjibi Ranabhat [19180056]

**Supervised By**

Er. Bhesh Bahadur Thapa

**Submitted To**

School of Engineering

Faculty of Science and Technology

# POKHARA UNIVERISTY

# COPYRIGHT

The author has granted authorization to the Library of Pokhara University's School of Engineering to grant unrestricted access to this report for examination. Furthermore, the author has agreed to the potential for extensive reproduction of this project report for educational purposes by either the project supervisors or, in their absence, by the program coordinator associated with the project's location. It is understood that proper acknowledgment will be accorded to both the author of the report and the School of Engineering, Pokhara University in any utilization of the report's contents. Unauthorized copying, publication, or any other form of exploitation of this report for financial gain without explicit consent from the University (School of Engineering) and the author is strictly prohibited. Requests seeking permission to replicate or employ any segment of the report's content should be directed to:

Program Coordinator

School of Engineering

Faculty of Science and Technology

Pokhara University

# STUDENT'S DECLARATION

We hereby declare that this project work entitled "**FUTSA**" is based on our original work. All concepts, data, code, and any other work from external sources have been properly cited and referenced in accordance with the guidelines provided by **SOE, Pokhara University** We owe all the liabilities relating to the authenticity and originality of this project work and project report.

Sincerely yours,

Saroj Aryal

Suraj Gautam

Amrit Adhikari

Chiranjibi Ranabhat

# SUPERVISOR'S RECOMMENDATION

This is to certify that this project report entitled **"FUTSA"** prepared and submitted by the below listed team of students in partial fulfillment of the requirements of the degree of Bachelor of Engineering in Computer/Software Engineering awarded by Pokhara University, has been prepared and completed under my supervision.

I hereby recommend the same for acceptance by Pokhara University.

Saroj Aryal

Suraj Gautam

Amrit Adhikari

Chiranjibi Ranabhat

…………………………..

**Er. Bhesh  Bahadur Thapa**

School of Engineering

Faculty of Science and Technology

Pokhara Univeristy

Date : ………………………

**School of Engineering, Faculty of Science and Technology
Pokhara University, Pokhara-30, Kaski**

# LETTER OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Computer and Software Engineering for acceptance, a project report entitled "*FUTSA*", submitted by Amrit Adhikari, Chiranjibi Ranabhat, Saroj Aryal and Suraj Gautam in partial fulfillment of the requirement for the Bachelor's degree in Software Engineering.

**Supervisor**                                                       **Program Coordinator**

...........................................                                  ………………………

Er. Bhesh Bahadur Thapa                                    Dr. Udaya Raj Dhungana

**External Examiner**

.............................................

Date : ………………………

# COVER LETTER

The Head of the Department

Department of Science and Technology,

School of Engineering

Pokhara University

Pokhara Metropolitan City - 30, Kaski


Dear Sir,

We the students of Software Engineering 8th Semester, a group intend to develop a project named "**FUTSA".** Our project is for the fulfillment of Project III, which is under the syllabus of Pokhara University in $8^{th}$ Semester.

Please find the report attached here for your kind approval and necessary arrangements.

Sincerely yours,

Saroj Aryal

Suraj Gautam

Amrit Adhikari

Chiranjibi Ranabhat

# ABSTRACT

FUTSA is a technical way of managing the futsal booking process. It will manage the time and the user can book futsal wherever and whenever. We will try to develop the project in such a way that users may not have any difficulty in using it and are highly secure. This system can be used by any Futsal owners, who intend to make the booking process in their business Online and simple to the players.

FUTSA, an innovative solution for futsal booking and management, has been successfully developed and implemented. This comprehensive system effectively streamlines the futsal court booking -process, offering users the convenience of booking courts at their preferred time and location. With a focus on user-friendliness and security, FUTSA has been designed to provide a seamless experience for both futsal admirers and court owners.

Keywords: *futsal, booking, bids, events, payment*

# ACKNOWLEDGEMENT

# TABLE OF CONTENT

# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATION

| | |
|------|---------------------------------|
| API  | Application Programming Interface |
| FCF  | Firebase Cloud functions |
| FK   | Foreign Key |
| JS   | Java Script |
| REPO | Repository |

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

In recent years, futsal has become more popular, and more people are playing futsal to stay active and socialize with friends and family. As a result, the demand for futsal pitches has increased accordingly, making it a challenge to book and manage them efficiently.

Generally, the traditional method of reserving a futsal spot with a phone call or a visit is cumbersome and can lead to double bookings and also seat conflicts. This can lead to customer dissatisfaction and possibly even loss of sales for the company offering the futsal pitch.

Similarly, addressing these challenges requires a modern, streamlined futsal booking and management system. An online system that allows customers to easily book futsal spots in real-time, make payments, and receive automatic reminders can greatly improve the booking experience for both customers and businesses.

Thus, Futsal management systems not only improve the booking process, but also help businesses manage their pitches more efficiently. This includes features such as managing pitch availability, managing leagues and teams, and tracking match stats. All of these features help businesses conduct futsal activities more effectively and improve the overall customer experience.

## 1.2 Problem Statement

Booking futsal pitches is often a slow and inefficient process that can lead to double-booking, conflict and frustration for both customers and businesses. Traditional methods of booking futsal pitches, such as phone calls and walk-ins, are clumsy and error-prone, which can lead to lost sales and a poor customer experience.

Additionally, managing futsal pitches and leagues can be a challenging task for businesses. Pitch owners can have trouble managing pitch availability, pricing, and tracking match statistics. This can lead to operational inefficiencies and ultimately affect the quality of the customer's futsal experience.

A modern futsal booking and management system is required to meet these challenges. The system should provide a streamlined and efficient booking process, real-time availability, automatic reminders, and secure payment processing. Additionally, the system should provide comprehensive pitch and league management tools, including: Track schedules, league and team management, and match statistics.

## 1.3 Objectives

The main objective of the project is to modernize the futsal industry's booking and management processes. Following are the objectives of our project:

➢ To develop a mobile application for browsing and booking futsal court in real time.
➢ To create a web application for court owners to manage futsal activities.

## 1.4 Implications

The scope of this project encompasses the design and development of a fully functional online futsal booking and management system. The system will cover the entire booking lifecycle, from checking court availability to making payments. However, the project does not include physical court maintenance or extend its features to cover sports other than futsal.

# CHAPTER 2
# LITERATURE REVIEW

## 2.1 Document Overview

Futsal has become an increasingly popular sport in recent years, with more and more people around the world joining the game. As the demand for futsal courts grows, so does the need for reservation systems and Efficient and friendly management also increases. This literature review is intended to provide an overview of existing studies and research related to the management and reservation systems in futsal.

### 2.1.1 Online reservation system for sports facilities

A study by Khan and Rahman (2015) [1] focused on online booking systems for sports facilities and their impact on customer satisfaction. Research shows that online booking systems can improve the efficiency of the booking process and improve customer satisfaction by providing real-time availability and reducing wait times. However, the study also notes that some customers may still prefer to book by phone or in person, underscoring the importance of offering multiple booking options.

### 2.1.2 Application of technology in the sports industry

Another study by Khorshid and Swatman (2019) [2] looked at the adoption of technology in the sports industry and the potential benefits of implementing technology in indoor facilities. Research shows that adopting technology can improve customer experience, increase revenue, and improve operational efficiency.

Specifically, the study highlighted the benefits of online booking systems, mobile applications, and electronic payment systems.

## 2.2 Background Study

The proposed futsal reservation and management system is based on existing research and studies related to online booking systems and technology adoption in the sports industry. Some of the main sources that have been studied include:

- Khan, M., & Rahman, M. (2015) [1]. Online booking system for sports facilities and customer satisfaction. Journal of Industrial Engineering and Management, 8(1), 259-276
- Khorshid, M., & Swatman, P. (2019) [2]. Applying technology in the sports industry: A study on factors affecting the adoption of electronic ticketing systems in futsal halls. Journal of Technology and Innovation Management, 14(3), 26-35.

These studies highlight the importance of applying technology and online booking systems to improve the efficiency and customer experience of sports facilities, including indoor soccer fields. In addition, several related projects provide similar functionality to the proposed futsal reservation and management system, including:

- **Futsal Now:** A web-based futsal reservation and management system that allows users to book and manage futsal courts and tournaments. The platform offers real-time availability, automated reminders, and payment processing, and is built using Reactjs, React native, Firebase. Futsal Now also offers a mobile app for customers to book and manage their bookings on the go [3].
- **Futsal Coach:** A mobile app-based indoor soccer booking and management system that allows users to book and manage indoor soccer fields and teams. The platform offers real time availability, team and player management tools, and payment processing [4].
- **FutsalBooking.com:** An online futsal reservation and management system that allows users to book and manage futsal courts and tournaments. The platform offers real-time availability, automated reminders, and payment processing, and is built with PHP and MySQL. FutsalBooking.com also offers a mobile app for customers to book and manage their bookings on the go [5].

This project "FUTSA"  futsal booking and management system was built on these results and existing projects by providing real-time availability, automated reminders, payment processing

and comprehensive tournament and court management tools through web and mobile apps. The system web app is built using ReactJS, while the mobile app is built using React Native and backend is provided by Firebase, providing a user-friendly and accessible platform for streamlined customers and businesses. Streamline the booking process and improve the overall futsal experience.

# CHAPTER 3

# SYSTEM ANALYSIS AND DESIGN

## 3.1 Methodology

## Agile Model

Agile is the combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software products. 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 [6]. Following are the Agile Manifesto principles:

- **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like colocation 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.
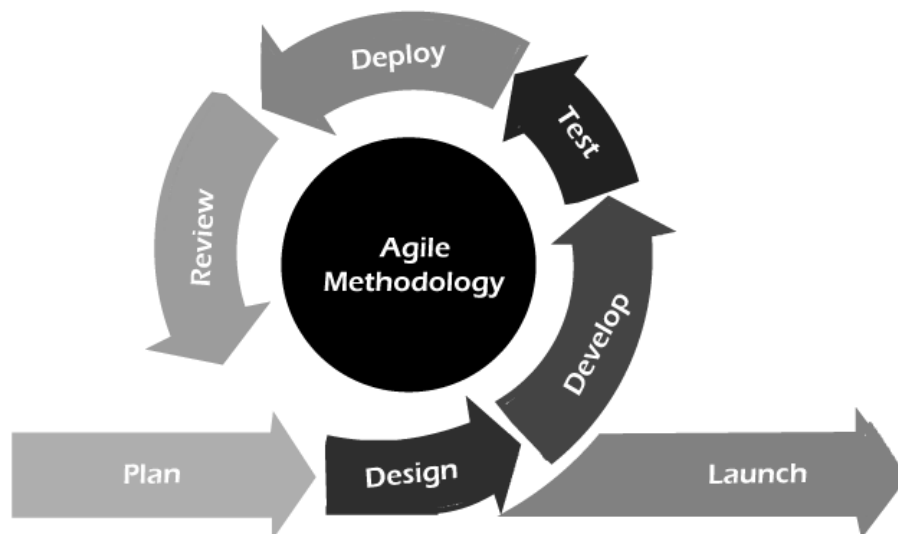


**Figure 1: Agile method**

- **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

## 3.2 System Analysis

System analysis is the process of evaluating requirements and processes in the development of the "FUTSA" futsal booking and management system. During this phase, system requirements were collected, analyzed and recorded.

Use cases and scenarios were created to define the functionality and features of the system. To understand the project and its functionality we gathered artifacts required. We also referred to other similar systems and learned how they worked.
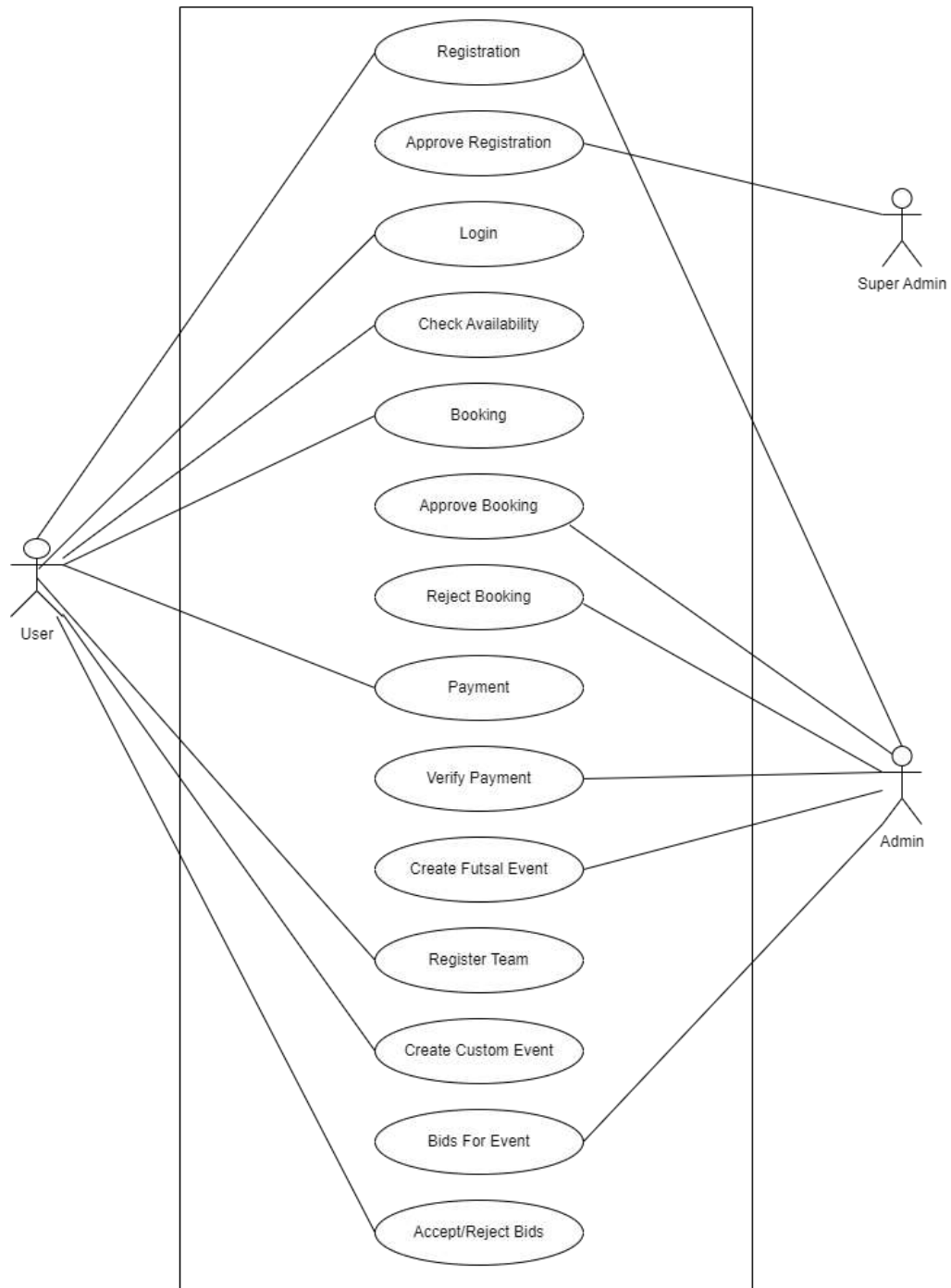
## 3.3 System Design

During this phase, the system architecture, components, and database schema will be designed. User interfaces, data flows and security protocols will also be designed.

### 3.3.1 System Diagram/Architecture
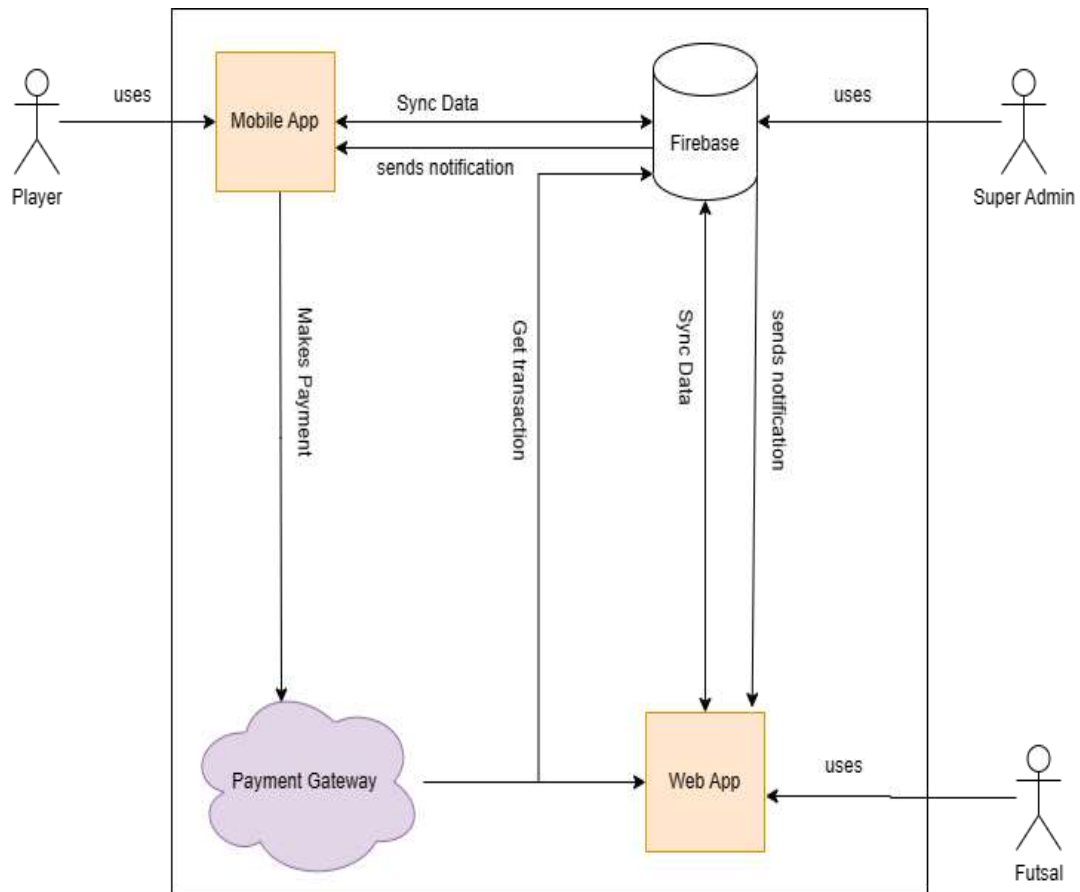


**Figure 2: System Architecture Diagram**

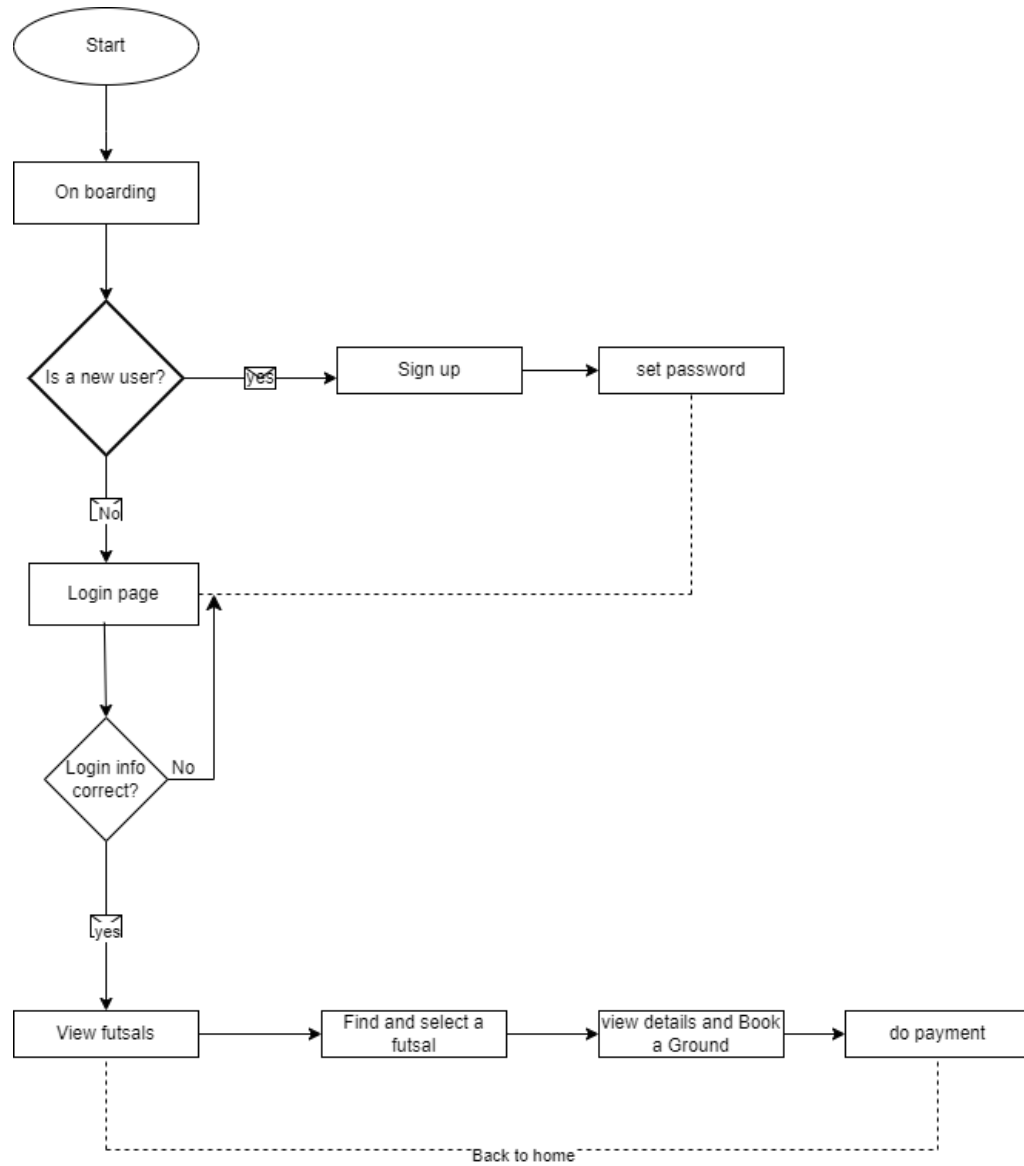## 3.3.2 Functionality modeling using Use Case Diagrams



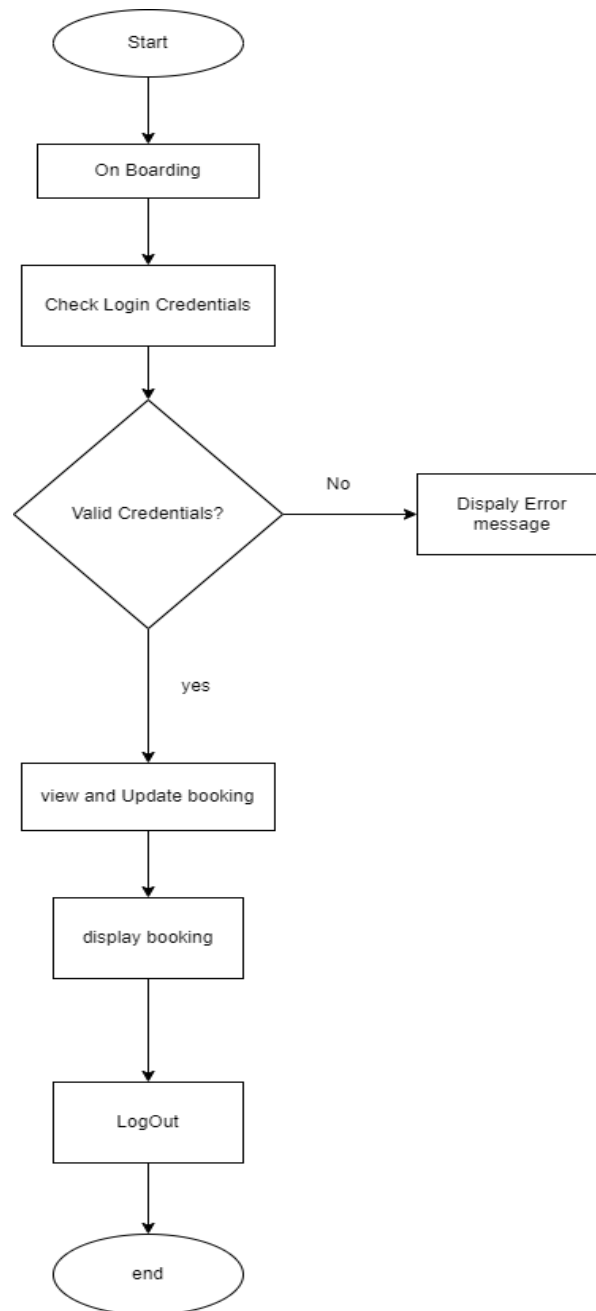**Figure 3: Use Case Diagram**

### 3.3.3 Activity Diagrams

1.  **Players Activity Diagram:**



**Figure 4: Players Activity Diagram**

**2. Futsal Activity Diagram:**



**Figure 5: Futsal Activity Diagram**

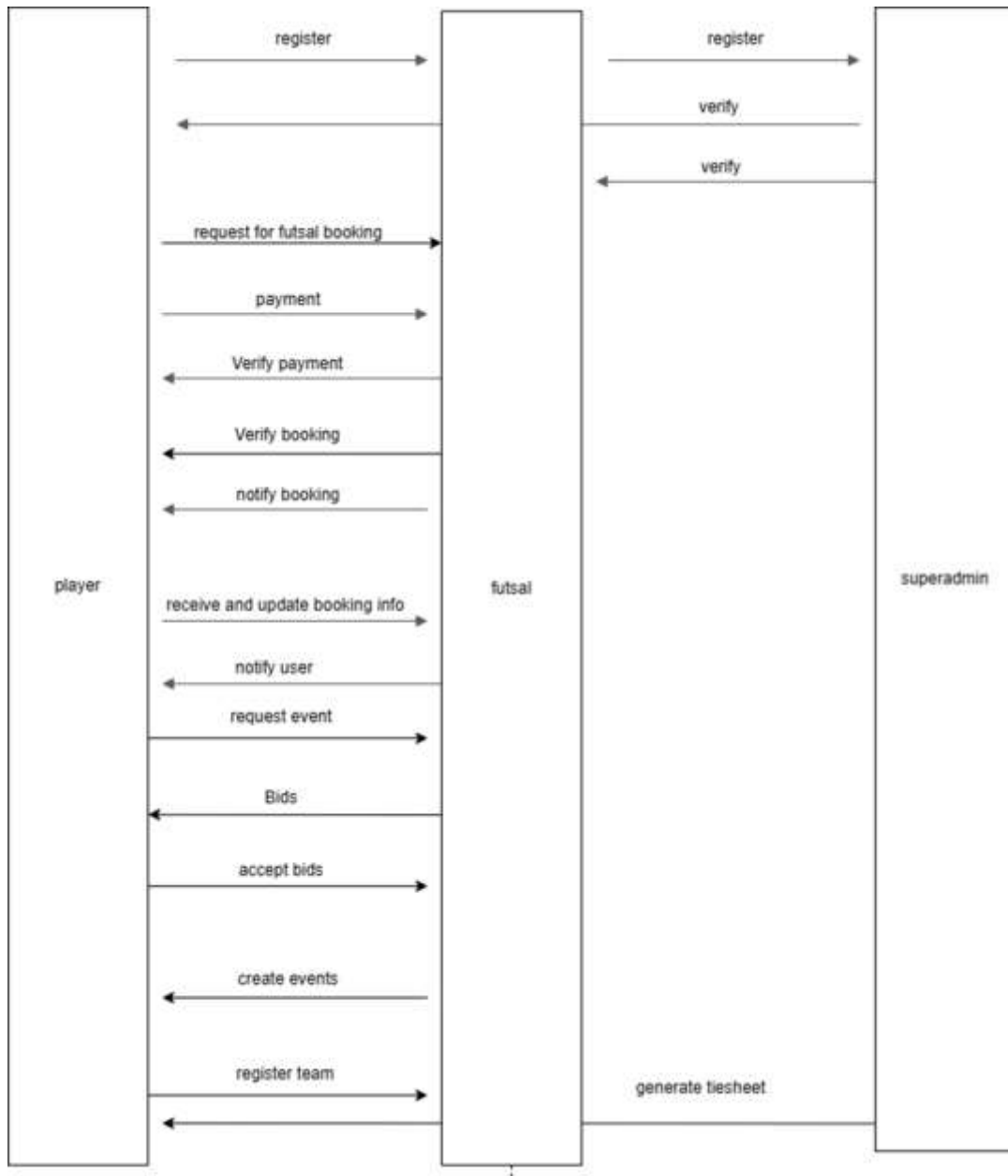## 3.3.4 Dynamic modeling using Sequence Diagrams

**Figure 6: System Sequence Diagram**

## 3.3.5 Data Directory

**Entity Name:** Futsal

**Entity Description:** Collection of registered futsals.

**Table 1: Data Directory of Futsal**

| Column Name | Column Description | Data Type | Primary Key | Nullable | Unique |
|---|---|---|---|---|---|
| id | Uniquely identifies a futsal | String | Yes | No | Yes |
| full_name | Name of a futsal | String | No | No | Yes |
| email | Email of a futsal | String | No | No | Yes |
| contact_number | Phone number of a futsal | Integer | No | No | No |
| location | Geolocation of a futsal | String | No | No | Yes |
| image | Cover image of futsal | String | No | No | No |
| status | Status of futsal Approved \| Declined | Enum | No | No | No |
| ratings | Average rating of the futsal | Integer | No | Yes | No |

**Entity Name:** Player

**Entity Description:** Collection of registered players.

## Table 2: Data Directory of Player

| Column Name | Column Description | Data Type | Primary Key | Nullable | Unique |
|---|---|---|---|---|---|
| id | Uniquely identifies a player registered | String | Yes | No | Yes |
| full_name | Name of the player | String | No | No | No |
| contact_number | Phone number of a player | number | No | No | Yes |
| email | Email of the player | String | No | No | Yes |

**Entity Name:** Bookings

**Entity Description:** Collection of booking data.

## Table 3: Data Directory of Bookings

| Column Name | Column Description | Data Type | Primary Key | Nullable | Unique |
|---|---|---|---|---|---|
| id | Uniquely identifies a booking. | String | Yes | No | Yes |
| user_id | Id of player making a book | String | No (FK) | No | No |
| futsal_id | Id of booked futsal | String | No (FK) | No | No |
| bookedDate | Time and date of booking | Date | No | No | No |
| status | Status of the booking | Enum | No | No | No |

**Entity Name:** Transactions

**Entity Description:** Collections of transactions.

**Table 4: Data Directory of Transactions**

| Column Name | Column Description | Data Type | Primary Key | Nullable | Unique |
|---|---|---|---|---|---|
| id | Uniquely identifies the transactions | String | Yes | No | Yes |
| send_from | User's Id | String | No ( FK) | No | No |
| send_to | Futsal's id | String | No (FK) | No | No |
| tnx_id | Unique identification of the transaction from API. | String | No | No | Yes |
| status | Status of a payment | Enum | No | No | No |

**Entity Name:** Custom Event

**Entity Description:** Collection of the custom request of the tournament.

**Table 5: Data Directory of Futsal**

| Column Name | Column Description | Data Type | Primary Key | Nullable | Unique |
|---|---|---|---|---|---|
| id | Unique identifier of a custom_event | String | Yes | No | Yes |
| title | Title of the event | String | No | No | No |
| Budget | Budget of the event | Integer | No | No | No |
| Description | Description about event | String | No | No | No |

15

**Entity Name:** Bids

**Entity Description:** Collection of bids for each Custom Events.

**Table 6: Data Directory of Bids.**

| Column Name | Column Description | Data Type | Primary Key | Nullable | Unique |
|---|---|---|---|---|---|
| id | Unique identification of a bid. | String | Yes | No | Yes |
| budget | Bid amount | Integer | No | No | No |
| message | Message regarding the bid | String | No | No | No |
| isSelected | Check if the bid has been selected | Boolean | No | No | No |
| event_id | Unique identifier of the event. | String | No (FK) | No | Yes |

# CHAPTER 4
# IMPLEMENTATION AND TESTING

## 4.1 Implementation

Futsa is a multi-platform sports management application developed on top of the React and React Native frameworks. It utilizes Firebase for its backend and real-time database needs, and Cloud Functions for server less computing. The project is structured using Turbo Repo, which is a system for managing multiple codebases in a single repository.

## 4.2 Tools Used

During the development of "FUTSA", various tools and technologies were used for the development. Here are some lists of tools and technologies used in the project:

### 4.2.1 Tech Stack

- **React and React Native:** These JavaScript libraries are used for building the web and mobile user interfaces of the Futsa application, respectively. React allows for creating reusable UI components and efficiently rendering them, while React Native extends this functionality to mobile platforms.

- **Firebase:** Firebase is a comprehensive cloud-based platform offered by Google, providing a wide range of tools and services for building web and mobile applications. In the context of Futsa, Firebase is used for user authentication, real-time database management, and cloud storage.

- **Cloud Functions:** Cloud Functions for Firebase enables server less computing, allowing developers to run custom backend code in response to events or HTTP requests. This is used for various backend tasks, such as sending notifications, data validation, and more.

- **Turbo Repo:** Turbo Repo is a setup for managing multiple codebases within a single repository. It can be particularly useful for projects that have both web and mobile components, such as Futsa.

### 4.2.2 Libraries and APIs

- **Khalti API:** Khalti payment gateway is  one of the best payment gateway services for nepali fintech industry. We implemented the API provided by them in our system to properly implement the transactions.

- **Material UI Library:** MUI is a popular and widely used open-source React UI framework that provides a set of pre-designed, customizable, and responsive user interface components following the principles of Google's Material Design. Material Design is a design language that emphasizes clarity, simplicity, and consistency in the visual and interactive aspects of user interfaces.

- **Google Maps API:** GMA is a powerful and widely used set of tools and services provided by Google that enables developers to integrate interactive maps, location-based features, and geospatial data into our applications. This API allows developers to leverage Google's extensive mapping data and services to enhance user experiences and build location-aware applications.

## 4.3 Project Structure

- **Web Application (React) :**  The web application is built using React and serves as the frontend interface for users accessing Futsa through web browsers. It provides features like registration, login, profile management, and event booking.

- **Mobile Application (React Native):** The mobile application is developed using React Native and is designed for users who prefer using Futsa on their smartphones. It mirrors many of the features available in the web application, with a mobile-friendly user interface.

- **Firebase Backend:** Firebase is used as the backend for both the web and mobile applications. It handles user authentication (sign-up, login) and stores data related to events, user profiles, bookings, and more. Firebase's real-time database ensures that data updates are immediately reflected across all clients.

- **Cloud Functions:** Cloud Functions are used to extend the backend functionality of Firebase. For example, when a user books an event, a Cloud Function can trigger notifications to be sent to other users attending the same event.

## 4.4 Project Repository structure

We used a concept called "MONO REPO" to structure our project in a single repository. It enables shareables codes for mobile and web app to be written only once and used in different development environments. Our folders within the project repo are as follows:

**Table 7: Table of project repository**

| Root Folder | Purpose |
|---|---|
| apps | It is the root folder consisting of different apps. |
| apps/mob | It contains all the files and folders related to react native for mobile app development. |
| apps/web | It contains all the files and folders related to Reactjs for web app development. |
| packages | It consists of the reusable resource for both mobile and web app |
| packages/core | It consists of database functions, models and queries. |
| packages/cfunc | It consists of cloud functions that help in enabling triggers, cronjobs and https methods. |

## 4.5 Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an undiscovered error.

### 4.5.1 Objectives of testing

- Testing helps to determine the accuracy of the system.
- A successful test is one that uncovers an as-yet-discovered error.
- To make sure that the results meets the business and user requirement.
- To study the behavior of the system.

### 4.5.2 Testing Goals

The goals in testing this application included validating the quality, usability, reliability and performance of the application. Testing was performed from a black-box approach, not based on any knowledge of internal design or code. Testing was done around requirements and functionality.

Another goal was to make the tests repeatable for use in regression testing during the project lifecycle, and for future application upgrades. A part of the approach in testing was to initially perform a 'Smoke Test' upon delivery of the application for testing. Smoke Testing is typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort.

For example, if the new software is crashing frequently, or corrupting databases, the software is not in a stable enough condition to warrant further testing in its current state. This testing was performed first. After acceptance of the build delivered for system testing, functions were tested based upon the designated priority (critical, high, medium, and low).

### 4.5.3 What were tested

The following features of the mobile app were tested for accuracy:

- Login player

- Register player

- Search Futsal

- Book Futsal

- Make payment

- Request custom event

- View and accept futsal bids

The following features of the web app were tested for accuracy:

- Login futsal

- Register futsal

- View/Accept/Reject Bookings

- Verify Transaction

- Create/Read/Update/Delete bids on event request by futsal

### 4.5.4 Entrance criteria

- All design specifications were reviewed and approved.

- Unit testing was completed by the development team, including vendors.

- All hardware needed for the test environment were available.

- The application delivered to the test environment were of reliable quality.

- Initial smoke test of the delivered functionality was approved by the testing team.

- Code changes made to the test site went through a change control process.

### 4.5.6 Exit Criteria

All test scenarios were completed successfully.

- All issues prioritized and priority issues resolved.

- All outstanding defects were documented in a test summary with a priority and severity status.

- Go/No-go meetings were held to determine acceptability of product.

### 4.5.7 Test Execution

The test execution phase was the process of running test cases against the software build to verify that the actual results meet the expected results. Defects discovered during the testing cycle were introduced to the developers. Once a defect was fixed by a developer, the fixed code was incorporated into the application and regression tested. These following testing phases was completed:

- **Unit Testing:** Unit testing was performed by the LGIS development team in their development environment. The developers knew and tested the internal logical structure of each software component. The application was passed on unit testing before working on a different model.

- **Functional Testing:** Functional testing focused on the functional requirements of the software and was performed to confirm that the application operates accurately according to the documented specifications and requirements, and to ensure that interfaces to external systems are properly working.

- **Regression Testing:** Regression testing was performed to verify that previously tested features and functions do not have any new defects introduced, while correcting other problems or adding and modifying other features.

- **Integration Testing:** Integration testing was the phase of software testing in which individual software modules were combined and tested as a group. In its simplest form, two units that have already been tested were combined into a component and the interface between them was tested. In a realistic scenario, many units were combined into components, which were in turn aggregated into even larger parts of the program. The idea was to test combinations of pieces and eventually expand the process to test your modules with those of other groups. Eventually all the modules making up a process were tested together.

- **Interface Testing:** This testing follows a transaction through all of the product processes that interact with it and tests the product in its entirety. Interface testing was performed to ensure that the product actually works in the way a typical user would interact with it.

- **Destructive Testing:** Destructive testing focused on the error detection and error prevention areas of the product. This testing was exercised in an attempt to anticipate conditions where a user may encounter errors. Destructive testing was less structured than other testing phases and was determined by individual testers.

- **User Acceptance Testing:** User acceptance testing activities were performed by the users. The purpose of this testing was to ensure the application meets the user's expectations. This also focuses on usability and includes appearance, consistency of controls, consistency of field naming, accuracy of drop-down field information lists, spelling of all field name/data values, accuracy of default field values, tab sequence, and error/help messaging.

- **Browser Testing:** Functional and Regression as defined in this test strategy were executed using Chrome, Safari, Firefox.

- **Mobile Testing:** Functional and Regression as defined in this test strategy were executed for android 8.0.0+ and iOS.

## 4.5.8 Test Result

The test results of each unit test and integration test are done while developing the system and are reviewed to identify and remove errors. The Following table consists of the test results of Black Box Testing which are performed to validate the system with respect to the requirement.

**Table 8: Table of test cases**

| Test Case ID | Test Scenario | Expected Results | Actual Results | Pass/Fail |
|---|---|---|---|---|
| TC1 | User Registration with incomplete information | User should not be registered if information is incomplete | User registration rejected | Pass |
| TC2 | Valid User Registration | User registration accepted | User registered successfully | Pass |
| TC3 | Unique Email address for each user | Cannot login with already used email address | User should login to system | Pass |
| TC4 | Password with length less than character | Password cannot be less than 8 character | User should not login to system | Pass |
| TC5 | Minimum 8-character valid password | Password accepted | User should login to system | Pass |
| TC6 | Password and Confirm Password did not match | Please enter same password | User should not login to system | Pass |
| TC7 | Booking | Users should be | Futsal slot | Pass |

| Test Case ID | Test Scenario | Expected Results | Actual Results | Pass/Fail |
|---|---|---|---|---|
| | Futsal Slot | able to book futsal | booked | |
| TC8 | Futsal Slot Double Booking Prevention | Users cannot book the same slot twice | Double booking prevented as expected | Pass |
| TC9 | Payment Processing | Payment transactions should be secure and accurate | Payment processing validated | Pass |
| TC10 | Event Creation | Futsal can post their event. | Event creation was successful | Pass |
| TC11 | Users register their team for the event. | User add their team | Futsal were able to view the registered team | Pass |
| TC12 | Users register their teams twice in a single event. | Prevent user from registering their team twice in same event | Double registration is prevented. | Pass |

# CHAPTER 5
# CONCLUSION AND FUTURE RECOMMENDATION

## 5.1 Conclusion

The FUTSA project represents a significant step toward modernizing the futsal industry's booking and management processes. By streamlining booking, enhancing user experience, and providing valuable tools for court owners, FUTSA contributes to the growth and efficiency of the industry.

Future enhancements and marketing efforts will further solidify its position as a valuable resource for futsal enthusiasts and businesses alike. We believe that the system's implementation and continued development will have a lasting positive impact on the futsal community.

## 5.2 Limitations

The project assists well to book the futsal venue and manage the booking process. The futsal can accept the post and the player can give review to the futsal after having successful service. Users can request custom events to which futsal can make their bids. Users approve the bid and contact the futsal. However, this project has some limitations:

● There is low Bid security.
● Difficult to handle fake customers

## 5.3 Future Research and Recommendations

To further enhance the capability of this application, we recommend the following features to be incorporated into the system:

● Nepali language interface.
● Provide backup and recovery of data.
● Provides a better user interface for users.

- Proper event management and details about tournament will be added.

- User and roles management for futsal.
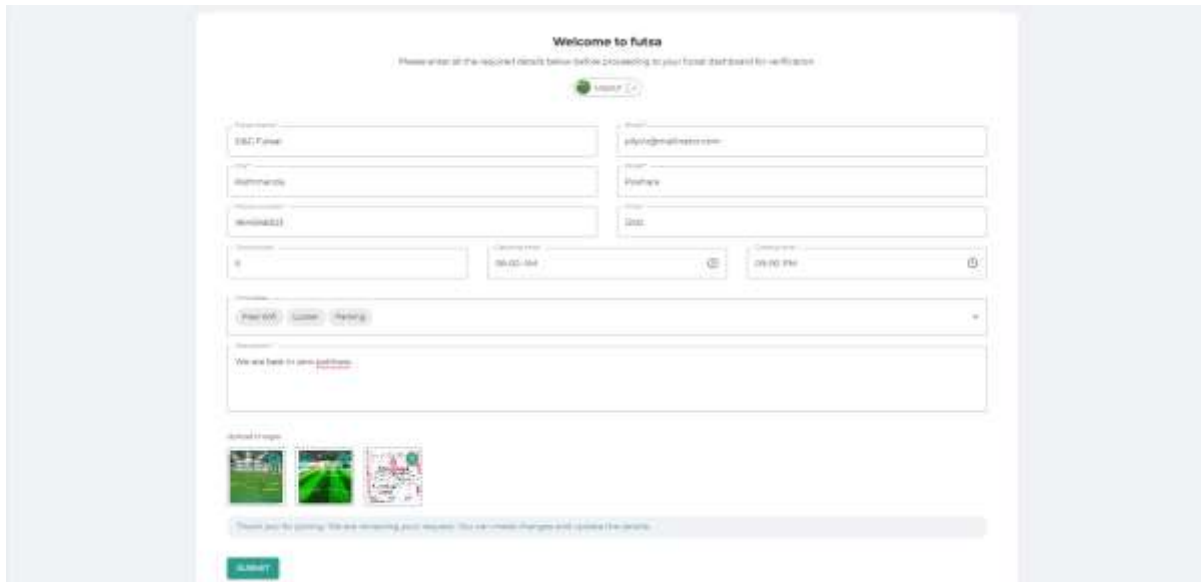
# CITATION AND REFERENCING

[1] M. Khan and M. Rahman, "Online booking system for sport facilities and customer satisfaction," *Journar of Industrial Engineering and Management,* vol. 8, no. 1, pp. 259-276, 2015.

[2] M. Khorshid and P. Swatman, "Technology adoption in the sports industry: A study of factors affecting electronic ticketing system adoption in futsal venues.," *Journal of Technology Management & Innovation,* vol. 14, no. 3, pp. 26-35, 2019.

[3] "FutsalNow," [Online]. Available: https://futsalnow.com. [Accessed 14 May 2023].

[4] "Futsal Manager," [Online]. Available: https://futsalmanager.com. [Accessed 1 June 2023].

[5] "Futsal Booking," [Online]. Available: https://futsalbooking.com. [Accessed 1 June 2023].

[6] Javatpoint, "Javatpoint," [Online]. Available: https://www.javatpoint.com/agile-vs-waterfall-model.

# APPENDICES
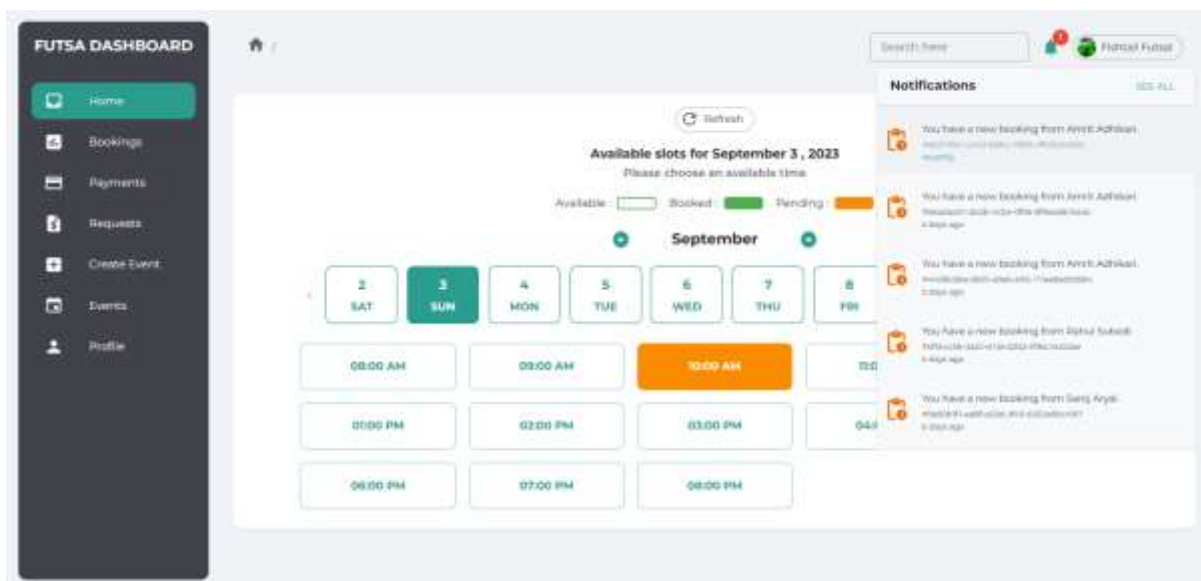
**User Interface**

**Web App ( Futsal Dashboard)**



**Figure 1 : Onboarding for futsal registration**



**Figure 2 : Futsal homepage with received booking notification.**

**Figure 3 : Booking requests table**


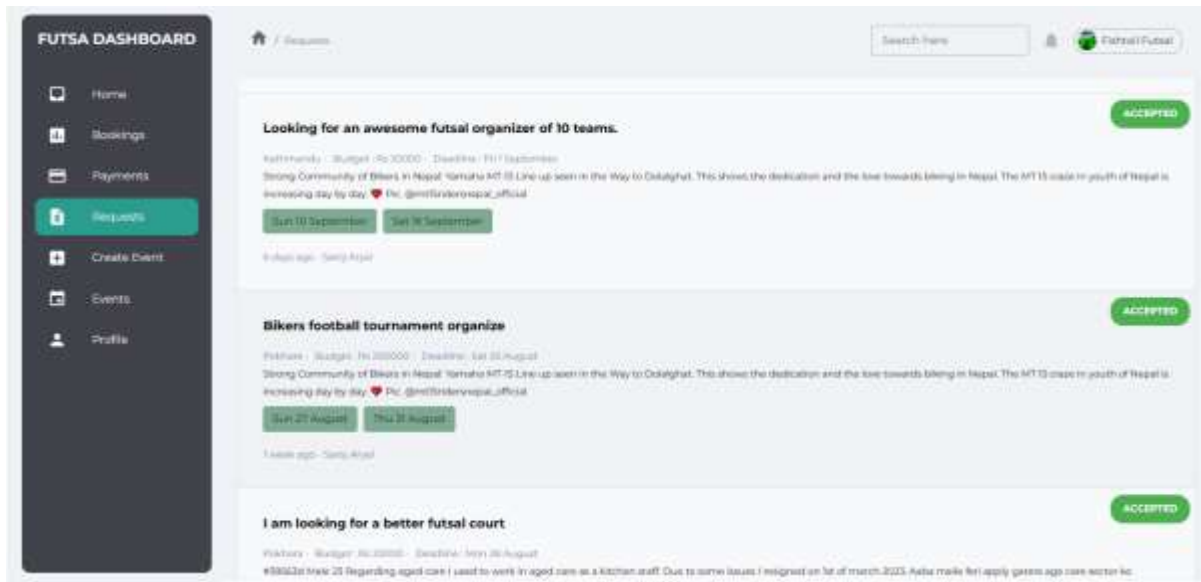
**Figure 3 : Booking details modal**

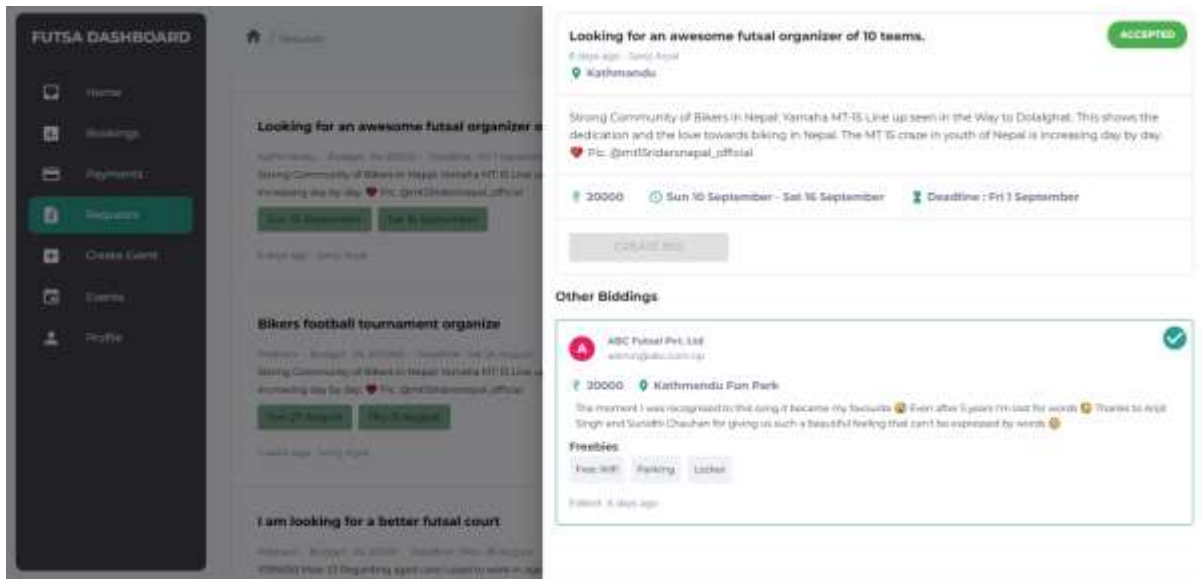**Figure 3 : Event Requests from the players/users to futsal**



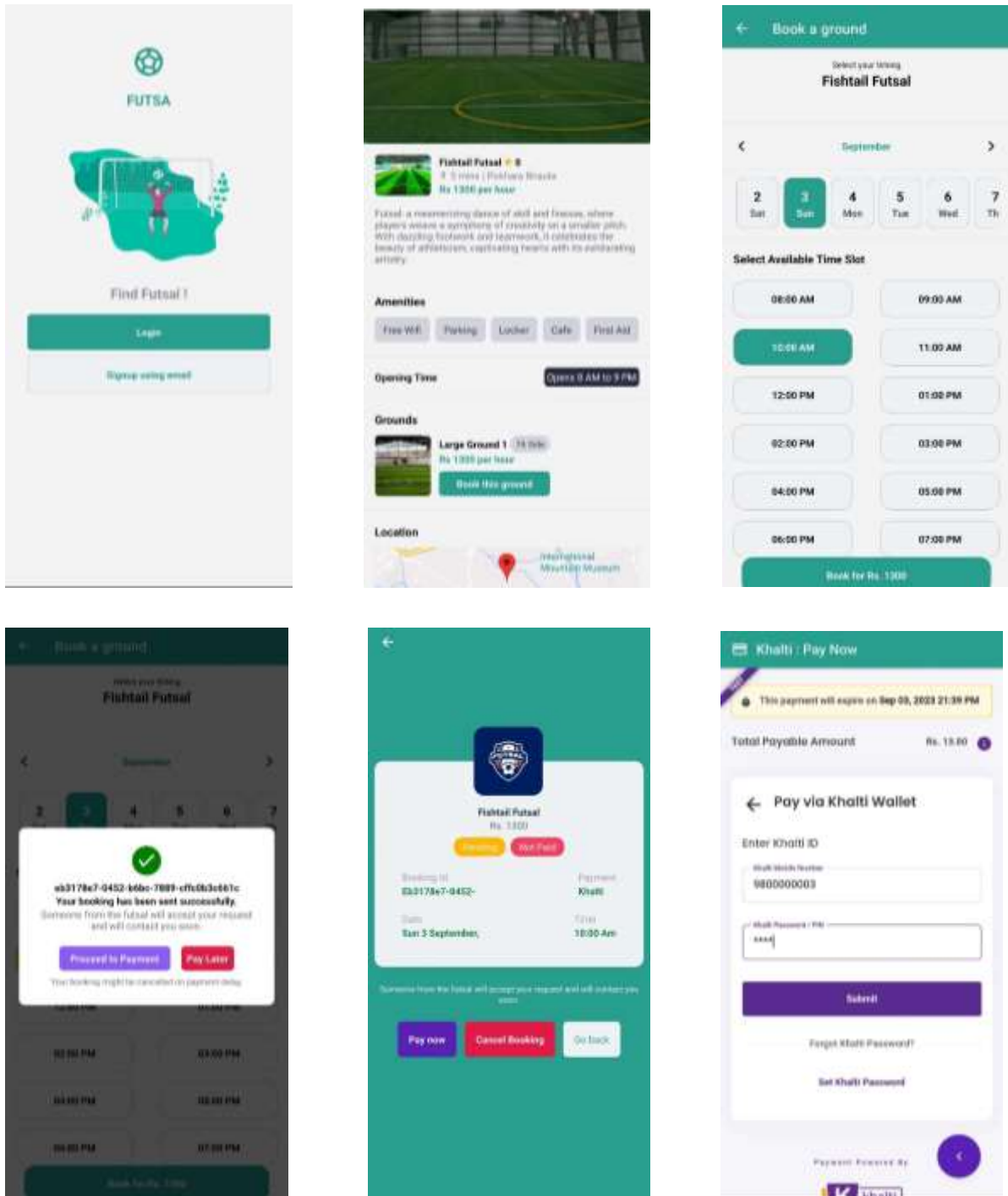**Figure 4 : Events request details with biddings.**

## Mobile app (Players)
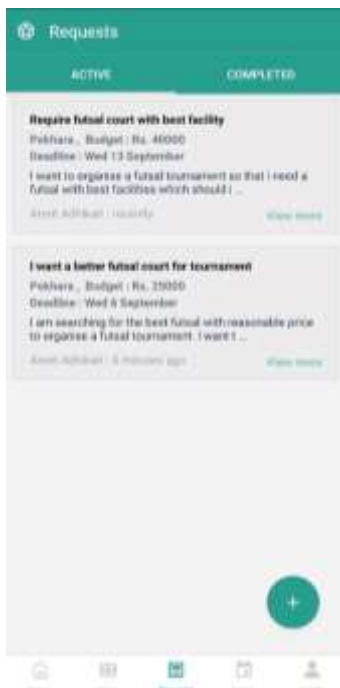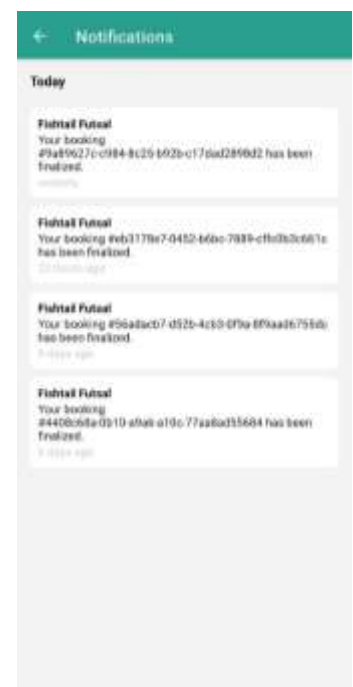


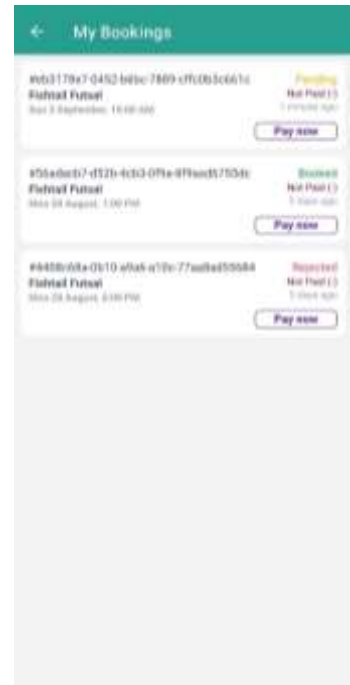Figure : Booking Futsal Flow screens.

**Figure : Event Flow Screens**

**Figure : Profile Screens**