**Chapter 1**

**INTRODUCTION**

There are many futsal grounds in Rawalpindi and Islamabad but if a local team wants to book a slot for their football game in any of these grounds, first they will have to go through each and every ground owner’s contact information to inquire if a slot is available or not. If the slot is already booked then they will have to repeat the entire process again which can prove to be quite the tedious process.Moreover, if they do find a slot they will still need to find players to play against. In this way such hurdles deter people in trying to play a simple game of football. **Futsal Field Reservation system** is a service that will allow local football teams, to reserve a pitch for their football games while simultaneously helping ground owners manage bookings for their facilities. Our platform offers a solution that streamlines facility management for providers. While also providing for the general public who want to book, practice and play on sports facilities. Our mission is to provide the local football by minimizing the ground booking hurdles that most of the players residing in our country face. And help turn ground renting into a lucrative business for the ground owners.

Our System will also offer an academy option, for people who are looking to learn, practice and hone their football skills, they can easily register through our website for academies of whichever ground is offering. We also aim to setup a donation system for impoverished players, who wish to play the game but are unable to due to various financial issues. So that everyone has equal chance. We are committed to increasing public access to football sports facilities.

* 1. **Project Domain:**

The category of our project is Web Application Development and Online Reservation Systems. The project will help tackle the real-life issue experienced by local footballers and futsal ground owners in Pakistan, especially in the twin cities of Rawalpindi and Islamabad. Our system fills this void by creating a web based platform through which teams can see, choose, and book existing slots online, and ground owners can add and maintain their venues. By eliminating this process through automation, the project maximizes convenience, minimizes time, and facilities a better-structured sports infrastructure.

* 1. **Problem Identification:**

No such system currently exist in Pakistan that serves the local football community .As a result Football players have to face so many issues when they were trying to book the ground. It’s a time consuming process on the owner’s and player’s, for the owner’s having to wait for people to contact them to get their grounds booked which from a business perspective isn’t profitable. And for the players having to contact each and every ground owner that they know of, and then trying to find if a team which will be available at the same time as them, which all in all proves to be quite the hassle.

So to provide the solution of these time consuming hurdles, we will offer our services both to the teams and ground owners, in the form of our website, **“Futsal Field Reservation System”** which will provide a friendly interface for both the footballers and ground owners to communicate. Our system will provide updates about slot is available or not , other teams can also chat with each other and discuss about the match should be held and other matters related to futsal.

**1.2.1. Proposed Solution:**

The proposed system will help footballers to make ground reservations for their games and at the same time facilitate ground owners to register their grounds with our system and get booking in an efficient and timely manner. The solution will come to play by providing a web-based service for both footballers and ground owners alike, where football players will have the option of viewing lists of grounds and instantly getting to know if a slot is available for booking or not. The system will offer features like user registration and login, ground and team profile management, real time chat for communication, academy registration, and a donation module.This project aims to help local football players of twin cities to book their matches, and professional manner rather than trying to contact each ground owner first for booking a slot and then rushing to find opponents and grounds.The project will be deployed as a web-based service with tools such as ReactJs, NodeJs, and NoSQL.This project will surely help in improving the futsal field reservation system in our twin cities.

**1.2.2. Objectives:**

* To design a comprehensive and user-friendly web platform for automating futsal ground reservations.
* To develop a real-time booking system that allows users to view, reserve, and manage time slots instantly.
* To develop and integrate an intelligent AI-powered chatbot capable of providing accurate, real-time information on ground availability, teams, and academies.
* To improve user accessibility and convenience by providing a 24/7 automated assistant for all inquiries.
* To increase the efficiency of the booking process, allowing for more reservations to be handled without human intervention.
* To reduce the manual administrative workload required for managing bookings and answering routine customer questions.
* To validate the developed system by testing its functionality, reliability, and user-friendliness to ensure it meets all project requirements.

**1.2.3. Scope of the Project**

The Futsal Field Reservation system is a service that allows local football teams to reserve a pitch for their games while also assisting ground owners in managing bookings for their facilities. Our platform provides a solution that greatly simplifies facility management for providers while also allowing the general public to book, pay for, and use sports facilities.

Our mission is to provide local football players in the twin cities with the opportunity to play a proper and professional game of football by reducing the ground booking obstacles that most players in our country face. And assist ground owners in turning ground renting into a profitable business. We also want to create a donation system for poor players, who want to play but cannot due to various financial issues.So everyone has the same chances.

* 1. **Effectiveness/ Usefulness of the System**

Our Futsal Ground Booking System is a real game-changer for the local football scene in Rawalpindi and Islamabad, as it's the first online platform built just for this sport in our area. At its heart, the system is designed to make booking a pitch incredibly simple and effective for everyone involved—from the players on the field to the owners who manage the grounds. Players no longer have to go through the hassle of calling one ground after another; instead, they can see every available time slot in one place and book their game instantly. For ground owners, it acts as an all-in-one tool to bring their business online, helping them fill up their schedules, reduce administrative headaches, and connect with more players than ever before. What makes it even more useful is the built-in AI assistant, which is ready 24/7 to answer questions and help users. Ultimately, this project is about more than just bookings; it’s about building a stronger football community by connecting players, teams, and academies, making it an essential tool for anyone passionate about the sport locally.

* 1. **Resource Requirement**

This section provides the necessary resources for the effective development and implementation of the project. It encompasses both the hardware and software resources, including development tools, programming languages, framework, and hosting environments.

**1.4.1. Hardware Requirement**

The creation and implementation of the this project need a computer with average specifications that can be used in web development. The system was developed and tested on devices that can support development tools such as Visual Studio Code, Nods.Js, and MongoDB. The Tabulation of the suggested hardware specifications are below:

**Table 1.1. Hardware Requirement**

|  |  |
| --- | --- |
| **COMPONENT** | **SPECIFICATIONS** |
| Processor | Intel Core i5 or Higher |
| RAM | 8 GB or higher |
| Storage | 256 SSD |
| OS | Windows 10, 11 |
| Internet | Stable Broadband |

**1.4.2. Software Requirement**

The creation of the Futsal Field Reservation System involved several software tools, frameworks, and programming languages in the implementation of both the front-end and back-end of the application. The tabulation of the major software and tools are mention below:

**Table 1.2. Software Requirements**

|  |  |
| --- | --- |
| **Tool / Software** | **Purpose** |
| ReactJs | Frontend Development |
| NodeJs & ExpressJs | Backend Server |
| MongoDB | Database Management |
| Firebase | Image and File Storage |
| Visual Studio Code | Code editing and Development |
| Google Chrome | Web application Testing / Browser |
| Google Ai Studio | Gemini Api Key |

**1.4.3. Data Requirement**

There are no machine learning models in this project, so no training or test datasets are necessary. All data, including user profiles, booking details, ground listings, and donations history, are stored and administrated dynamically via the MongoDB database as people use the site.

**1.5. Report Organization**

This report is organized into six chapters to give you a complete tour of our Futsal Reservation System. We begin in Chapter 1 by introducing the project, outlining the problem we aimed to solve, and defining our goals. Chapter 2 describes the system we built, detailing its key features and objectives. We then move into the technical specifics in Chapters 3 and 4, covering everything from user requirements to the system's architecture with UML diagrams. Chapter 5 explains how we tested the system to ensure it works correctly, and finally, Chapter 6 concludes the report with our final thoughts, achievements, and ideas for future improvements.

**CHAPTER 2**

**BACKGROUND AND EXISTING SYSTEMS**

The Futsal Field Reservation System caters to the increasing demand for a centralized and organized way to reserve futsal grounds in city towns, especially in the twin cities of Rawalpindi and Islamabad. Presently, ground owners and players are struggling with major obstacles since there is no digital platform available for ground management and booking.

In this chapter, we have talked about the available systems like Pitchbooking(UK), Gwsports(india), and other existing similar systems providing ground booking facilities in various countries. We have pointed out their features, usage, and major differences. We have determined the weakness of these systems, particularly their inability to support localized requirements and community based functionalities like academies and donations that our suggested system will address.

**Table2.1: (Bench Marking)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **System** | Chat  Box | Ground  Owner’s  Dash-  Board | Teams  Dash-  board | Academy | Listings | Map  Based | Donations |
| **Pitchbooking** | **☑** | **☑** | **🗵** | **🗵** | **☑** | **☑** | **🗵** |
| **Gwsports** | **🗵** | **🗵** | **🗵** | **🗵** | **☑** | **🗵** | **🗵** |
| **PlaySpots** | **☑** | **🗵** | **🗵** | **🗵** | **☑** | **🗵** | **🗵** |
| **Futsal Field**  **Reservation**  **System** | **☑** | **☑** | **☑** | **☑** | **☑** | **☑** | **☑** |

**2.1.Related Literature Review**

**Paper 1: Pitchbooking(UK)**

**Overview:** Pitchbooking is a UK-built online booking system where users can discover, observe, and reserve sports pitches such as football pitches and tennis courts within different cities.

**Findings:** It provides an easy-to-use interface for viewing real-time availability, automatic confirmation of booking, and facility management software for ground owners. The system enhanced operational effectiveness of the customer for private sports providers and local councils.

**Limitations:** Pitchbooking is primarily directed towards developed nations and lacks community features such as team matchmaking, academy integration, or donation support for needy players.

**Paper 2: Gwsports(India)**

**Overview:** Gwsports is an Indian website that brings players into contact with sports grounds for the likes of cricket, badminton, and football. It allows users to book pitches according to location and desired time using a web and mobile platform.

**Findings:** The system enhances access to sporting facilities and aids event management and sports coaching. It has a storing presence in tier-2 and tier-1 Indian cities, hence applicable for new markets in sports.

**Limitations:** Gwsports does not have any social interaction feature-teams, academy cannot converse directly. Nor does it offer any provision to assist cash-strapped players.

**Paper 3: PlaySpots(India)**

**Overview:** PlaySpots is yet another Indian platform that aims to make sports complexes accessible by enabling real-time bookings of courts and grounds for different sports such as futsal, tennis, and basketball.

**Findings:** It provides facilities such as location based venue search, time-slot reservation, and user reviews of venues. Its mobile application allows click booking and reminders.

**Limitations:** PlaySpots, although effective at booking facilities, lacks attributes such as infrastructure for donations, team profiles, and academy sign-ups--restricting the opportunities for community creation.

**Table 2.2: Summary of Reviewed Literature**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year** | **Authors** | **Contribution** | **Techniques** | **Limitations** |
| 2018 | PitchBooking  (UK) | Provided a centralized platform for booking sports fields | Web & mobile platform with dashboard | No team interaction, donations, or academy features |
| 2019 | GwSports | Connected users with sports grounds and coaching centers | Web & mobile Based booking system | No team interaction, and donations features |
| 2020 | PlaySpots | Allowed real-time booking of Spots venues via mobile app | Location-based  Facility search app | Lacks team features, no donations or academy management system |

**2.2. Related Systems / Applications**

Some systems already exists with the aim of streamlining sports ground booking, each one with special characteristics. Below are two dominant systems that motivated our solution.

**System 1: Pitchbooking(UK)**

Pitchbooking is an online booking platform for football and other sports pitches. It features live availability, automated confirmation of bookings, and facility management tools. It does not have team profile creation, chatting options, or support for community based sports development, which makes it less useful in developing nations such as Pakistan.

**System 2: Gwsports(India)**

Gwsports provides user with the opportunity to reserve sports centers in different Indian cities. It provides event organizing and training services. It is widely available, it lacks the facility of real-time chat between users as well as donation options for needy players, so it is less friendly to community-oriented objectives.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **System** | **Contribution** | **Tool / Technologies** | **Limitations** | **Application** |
| 2018 | Pitchbooking | Online sports pitch booking and management | Web & mobile platform | No team chat, no donations, lacks communities features | Sports facility booking |
| 2019 | Gwsports | Sports ground booking & coaching connection | Web & mobile app | No team profiles, no chat, no donation support | Event & venue booking |

**Table 2.3: Summary of Existing System**

**2.3. Identified Problem from Work**

When we looked at other sports booking websites, like Pitchbooking in the UK or Gwsports in India, we noticed they have some major limitations. For starters, they don't offer any way for teams and ground owners to chat in real-time, which makes communication difficult. They also miss key features that help build a football community, like letting teams set up their own profiles.

Beyond the tech, these platforms aren't designed for countries like Pakistan. They lack local focus and don't include options to support less privileged players through donations. That's where our project comes in. We're building a system that brings everything together easy communication, team and academy features, and a donation option to create a booking platform that truly works for local players and ground owners.

**2.4. Selected Boundary For Proposed Solution**

The Futsal Reservation System proposed in this project will incorporate major functionalities like user registration, ground listing and reservation, profile management, real-time chat, and academy registration. These functionalities are required to solve the issues recognized in Section 2.3, which involve in sufficient communication, centralized booking, and opportunities for skill development. The system will also incorporate a donation feature to help unprivileged players so that the system is more inclusive and community oriented.

However, some advanced features will not be supported because of time and resource limitations. They are mobile application creation, and support for sports other than futsal.

These features are useful, they are beyond the initial scope and can be taken into consideration at future upgrades. By explicitly specifying what is included and excluded, we define a realistic and workable scope for our project development.

**CHAPTER 3**

**SYSTEM REQUIREMENT AND   
SPECIFICATIONS**

This chapter presents the detailed functional and no-functional specifications of the Futsal Field Reservation System, encompassing a complete technical view and system require- ments for the development of the system. The requirements are divided to maintain clarity, feasibility, and accordance with the project goals. Functional requirements specify the system primary functions, including user registration, ground booking, and communications features, whereas non-functional requirements cover performance, usability, security, and compatibility.

The chapter divided into sections for orderly presentation. Section one presents the document structure and convention. The next sections cover the functional requirements with descriptions of the system’s features and interactions, followed by the non-functional which encompass operational constraints and quality attributes. Further, User interface design is shown to give a visual description of the interaction points of the system. This systematic methodology provides a clear system of understanding the requirements of the system and forms the basis of its design and implementation.

**3.1. System Specification**

This section provides the main technical and operational specifications of the Futsal Field Reservation System, providing a clear picture of its form and functions. Here, we explain the system’s functional requirements like user sign-up, pitch-booking, team messaging, AI chat bot, and donation tracking which are the essential features users will be working with. We also discuss non-functional requirements, security features, usability requirements, and cross platform compatibility. These requirements form the backbone for development, informing the design and implementation stages to build a smooth, effective, and user-friendly system for football teams and ground owners. By tackling these issues, we ensure the system is viable in real world scenarios as well as ensure reliability and scalability for future advances.

**3.2. System Modules**

System modules are the building blocks of a software project, where the program is broken down into separate, interchangeable components. Each module handles a specific function and contains all the necessary code to accomplish its task. We designed our system this way to keep it organized, making it easier to develop, test, and update each part independently. The main modules of our project include User Management, Booking and Scheduling, and AI Chat-bot.

**3.2.1. User Management Module**

This module handles all aspects related to user accounts and profiles. It allows new users to register as players, team captains, or ground owners, and manages the secure login and authentication process. Once logged in, users can view and edit their profile information, manage their teams, and track their booking history. This module is essential for creating a personalized and secure experience for every user on the platform.

**3.2.2. Booking and Scheduling Module**

The Booking and Scheduling Module is the core of the reservation system. It allows users to browse different futsal grounds, view their real-time availability on a calendar, and select their desired time slots. This module processes the booking requests, handles online payments through an integrated payment gateway, and sends out confirmation notifications. It is designed to be intuitive and efficient, making the process of reserving a ground quick and hassle-free.

**3.2.3. AI Chat-bot Module**

This module features the "Futsal Assistant," an intelligent chatbot designed to enhance user support. It integrates with the system's database to provide instant, accurate answers to user queries about ground availability, pricing, and team information. The AI is trained to understand natural language and provide conversational responses, offering 24/7 assistance and reducing the need for manual customer service.

**3.3. Functional Requirements/Software Features**

Functional requirements define the specific behaviors and features of the system, outlining what it must do to meet the project's objectives. For the Futsal Field Reservation System, these requirements are designed to create a comprehensive and user-friendly platform for the local football community. The overall features focus on providing seamless user registration, efficient booking management, intelligent user support through an AI chat-bot.

**3.3.1. User Registration and Authentication**

The system shall allow users to create a unique account by providing essential information such as name, email, and a password. It must support different user roles, including Player, Team Captain, and Ground Owner, each with distinct permissions. The system will enforce secure password policies and provide a login mechanism to authenticate users before granting them access to the platform's features. It will also include a profile management section where users can update their personal information.

**3.3.2. Ground Booking and Management**

This feature will enable authenticated users to search for available futsal grounds based on location and other criteria. The system must display a real-time calendar showing available and booked time slots for each ground. Users shall be able to select a desired slot, proceed to checkout, and make a secure online payment to confirm their booking. Upon successful booking, the system will automatically send a confirmation email to the user.

**3.3.3. Team Communication**

The system shall include an in app chat feature that enables teams to communicate with each other and with ground owners. This functionality facilities match coordination, dispute resolution, and general inquires, enhancing user interaction and convenience.

**3.3.4. AI Chat-bot for User support**

The system shall include an AI-powered chatbot to provide 24/7 automated support to users. This chatbot must be able to understand and respond to natural language queries regarding ground availability, booking procedures, pricing, and other frequently asked questions. It will access the system's database in real-time to provide accurate and up-to-date information, enhancing user experience and reducing the need for manual support.

**3.3.5. Team and Academy Profiles**

The system shall allow users to create and manage profiles for their teams and academies. This feature will enable users to add team members, upload logos, and showcase their history or mission. Other users will be able to view these profiles, fostering a sense of community and allowing teams and academies to connect with potential players or opponents.

**3.4. Non-Functional Requirements**

While functional requirements describe what a system does, non-functional requirements (NFRs) define how the system performs. Also known as quality attributes, these requirements are crucial for ensuring the system is usable, reliable, and efficient. For our Futsal Field Reservation System, the key non-functional requirements are usability, reliability, performance, and security, as these qualities directly impact the user's experience and trust in the platform.

Each non-functional requirement is described in detail below.

**3.4.1. Performance**

Performance relates to the speed and responsiveness of the system. In today's fast-paced digital world, users expect websites to load quickly. A slow or lagging system will lead to a poor user experience and may cause users to abandon the site before completing a booking. Performance will be measured by page load time; all critical pages, including the homepage and booking calendar, must load completely in under 3 seconds on a standard internet connection.

**3.4.2. Security**

Security involves protecting the system and its data from unauthorized access, use, or disruption. As the platform will handle personal user information and financial transactions, robust security is non-negotiable. A breach of security could lead to data theft and a complete loss of user trust. The system's security will be measured by its ability to protect against common web vulnerabilities. It must successfully pass regular security audits based on the OWASP Top 10 security risks.

**3.4.3. Usability**

Usability refers to the ease with which users can interact with the system. For a public-facing booking platform, high usability is essential because it ensures that even first-time visitors can navigate the site and make a booking without frustration. A system that is difficult to use will quickly lose customers. We will measure usability by observing new users; if a user can successfully register, find a ground, and complete a booking in under three minutes without assistance, the system will be considered highly usable.

**3.4.4. Reliability**

Reliability is the measure of the system's ability to perform its functions correctly and consistently over time. In a booking system that handles schedules and payments, reliability is critical. If the system frequently crashes or loses booking information, it will cause significant problems for both players and ground owners, destroying the platform's credibility. The system's reliability will be measured by its uptime; it must be operational and available to users 99.5% of the time, excluding planned maintenance periods.

**Chapter 4**

**SYSTEM MODELING AND DESIGN**

Before you can build something great, you need a solid plan. That's exactly what system design is for. It helps us figure out what the system needs to do and then creates a technical blueprint to guide the developers. This process clears up confusion, keeps everyone on the same page, and makes sure we build the system in a smart, organized way.

For our Futsal Field Reservation System, we used a few different diagrams to map everything out. We created diagrams to show how data flows through the system, how it interacts with the outside world, and what the overall technical structure looks like. These models were chosen to give a clear picture of how the system works, making the development process smoother and ensuring the final product is exactly what users need.

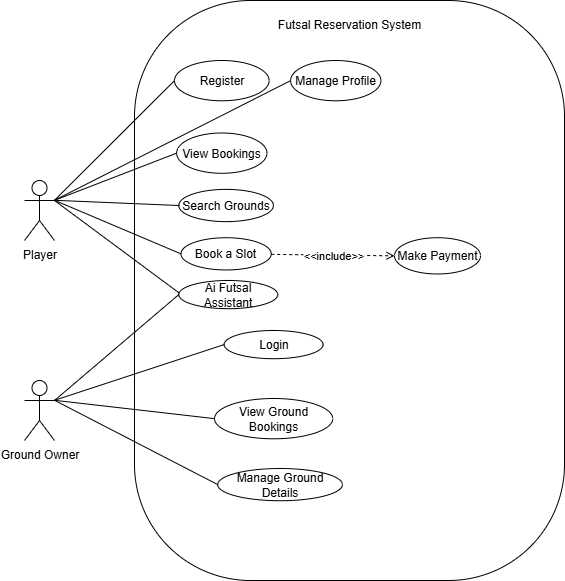
**4.1. System Design and Analysis**

To build our Futsal Reservation System, we started by creating a few key diagrams to map everything out. We used a Context Diagram to get a big-picture view of how the system interacts with users like teams and ground owners. Then, we created Data Flow Diagrams to trace how information moves through the system, from a user signing up to booking a ground. We also put together a System Architecture diagram to show the technical backbone how the website, server, and database all connect. Finally, we made a High-Level Design table to list out the main components and tools we used. These models gave us a clear blueprint, making the development process much more organized and efficient.

**4.2. Use Case Diagrams**

Think of a Use Case diagram as a simple map of what users can do on our site. It's a must-have for our project because it puts us in the shoes of our users and helps us define what the system needs to do before we start building.

As you can see in the diagram below (Figure 4.1), we focus on our two main users: the Team and the Ground Owner. It clearly lays out their tasks a Team can book a slot, and an Owner can manage their ground details. They also share common actions like logging in and getting help from the AI Futsal Assistant. This gives us a straightforward look at all the system's features from the user's point of view.



**Figure 4.1:**Use Case Diagram

**4.3. Full Dress Use Case/Detailed Use Case**

While the diagram gives us a quick look at what the system can do, a "full dress" or detailed use case dives much deeper. Think of it as a step-by-step story that describes the back-and-forth between a user and the system for one specific feature. We don't write these for everything that would be overkill. Instead, we reserve them for the most important or complex functions of our system, especially those with multiple steps where things could go differently than planned.

**4.3.1. Full Dress Use Case for Book Slot**

Table 4.1: Full Dress use case for book slot

|  |  |
| --- | --- |
| **Use Case Selection** | **Comment** |
| Use Case Name | Book Slot |
| Scope | Futsal Field Reservation System |
| Level | User goal |
| Primary Actor | Team |
| Stakeholders and Interest | **Team:** Wants to find and book an available ground easily.  **Ground** **Owner:**  Wants their available slots to be booked to generate revenue. |
| Pre-conditions | The Team must be logged into the System. |
| Success Guarantee | The selected time slot is reserved for the Team, an entry is made in the system’s booking record, and the ground Owner is notified. |
| Main Success Scenario | 1. Team navigates to the Booking or Search Grounds page. 2. System displays a list of a available grounds. 3. Team Selects a specific ground. 4. System displays the ground’s details and a calender of available slots. 5. Teams selects a desired date and slot. 6. Team clicks ‘Confirm Booking’. 7. System validates the slot is still available, reserves it and displays a confirmation message to the Team. |
| Extensions | **No slots available:** If the selected ground has no available slots, the system displays a message indicating this.  **Slot just became Unavailable:** If another teams books the slot during the process, the system displays an error message and prompts the user to select another slot. |
| Special Requirements | The system must provide real-time to prevent double booking. The interface must be responsive and work on mobile devices. |

**4.3.2. Full Dress Use Case for Manage Ground Details**

Table 4.2. Use case For manage ground details

|  |  |
| --- | --- |
| **Use Case Selection** | **Comment** |
| Use Case Name | Manage Ground Details |
| Scope | Futsal Field Reservation System |
| Level | User goal |
| Primary Actor | Ground Owner |
| Stakeholders and Interest | **Ground Owner:** Wants to present their facility attractively and accurately to attract bookings.  **Team:** Wants clear, accurate information to choose a suitable ground. |
| Pre-conditions | The Ground Owner must be logged into their account. |
| Success Guarantee | The ground's profile is successfully created or updated in the system database and is visible to all teams using the platform. |
| Main Success Scenario | 1. Ground Owner logs in and navigates to their dashboard/profile. 2. Ground Owner selects the option to 'Add' or 'Edit' a ground.   3. System presents a form for ground details (name, location, price, etc.).  4. Ground Owner fills in the details.  5. Ground Owner uploads photos of the facility.  6. Ground Owner sets the available time slots and any specific rules.  7. Ground Owner saves the changes.  8. System validates the data, saves it to the database, and displays a 'Successfully Updated' message. |
| Extensions | **Invalid file format:** If the user tries to upload a file that is not a supported image format (e.g., JPG, PNG), the system shows an error message.  **Missing required information:** If a mandatory field (e.g., ground name) is left blank, the system prevents saving and highlights the required fields. |
| Special Requirements | Image uploads must be optimized for web display. The location should be linkable to a map service. |

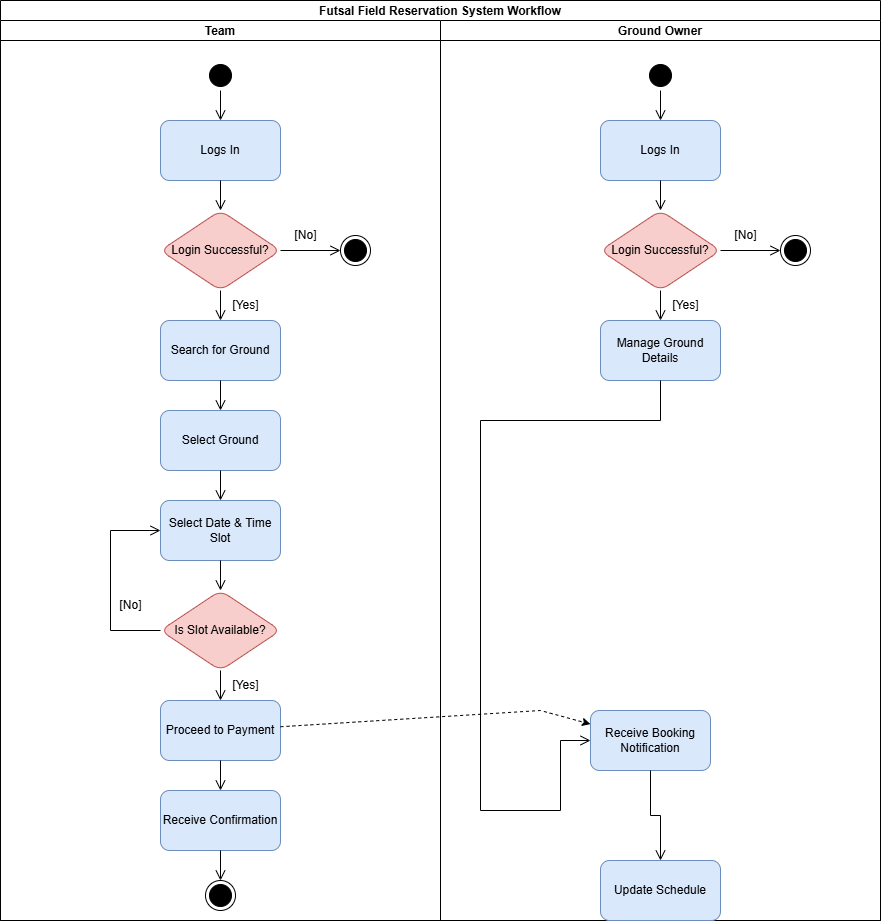
**4.3.3. Full Dress Use Case for Futsal Assistant**

Table.4.3. Use case for Futsal Assistant

|  |  |
| --- | --- |
| **Use Case Selection** | **Comment** |
| Use Case Name | Use Futsal Assistant |
| Scope | Futsal Field Reservation System |
| Level | Subfunction |
| Primary Actor | Team, Ground Owner |
| Stakeholders and Interest | **User (Team/Ground Owner):** Wants quick answers to common questions without having to search the site or contact support. **System Owner:** Wants to reduce the support load and improve user satisfaction. |
| Pre-conditions | The user is on any page of the website where the chatbot icon is visible. |
| Success Guarantee | The user receives a relevant and helpful answer to their query or is guided to the correct section of the website. |
| Main Success Scenario | 1. User clicks on the 'Futsal Assistant' chat icon.  2. The chatbot interface opens with a welcome message and suggested topics.  3. User types a question into the chat box (e.g., "How do I book a ground?").  4. The Futsal Assistant processes the natural language query.  5. The Assistant provides a direct answer and may include a link to the relevant page (e.g., the bookings page).  6. The user's query is resolved. |
| Extensions | **Query not understood:** If the Assistant cannot understand the query, it responds with a message like, "I'm sorry, I don't understand. Could you rephrase, or would you like to see a list of common topics?"  **Request for human support:** If the user's query is too complex or they explicitly ask to speak to a person, the Assistant provides contact information for human support. |
| Special Requirements | The chatbot must be able to process natural language. Responses should be fast (under 2 seconds). |

**4.4. Activity diagram:**

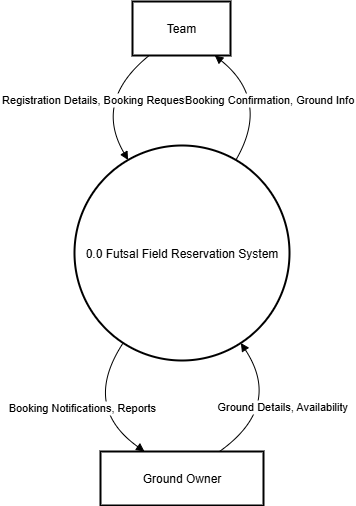
Think of an activity diagram as a detailed flowchart that maps out a process from start to finish. It’s perfect for visualizing complex workflows that have multiple steps and decisions. For our project, we've used an activity diagram to show the core journey for both a Team user and a Ground Owner. This clearly illustrates the most important actions in our system, like booking a ground or managing a facility, from beginning to end.



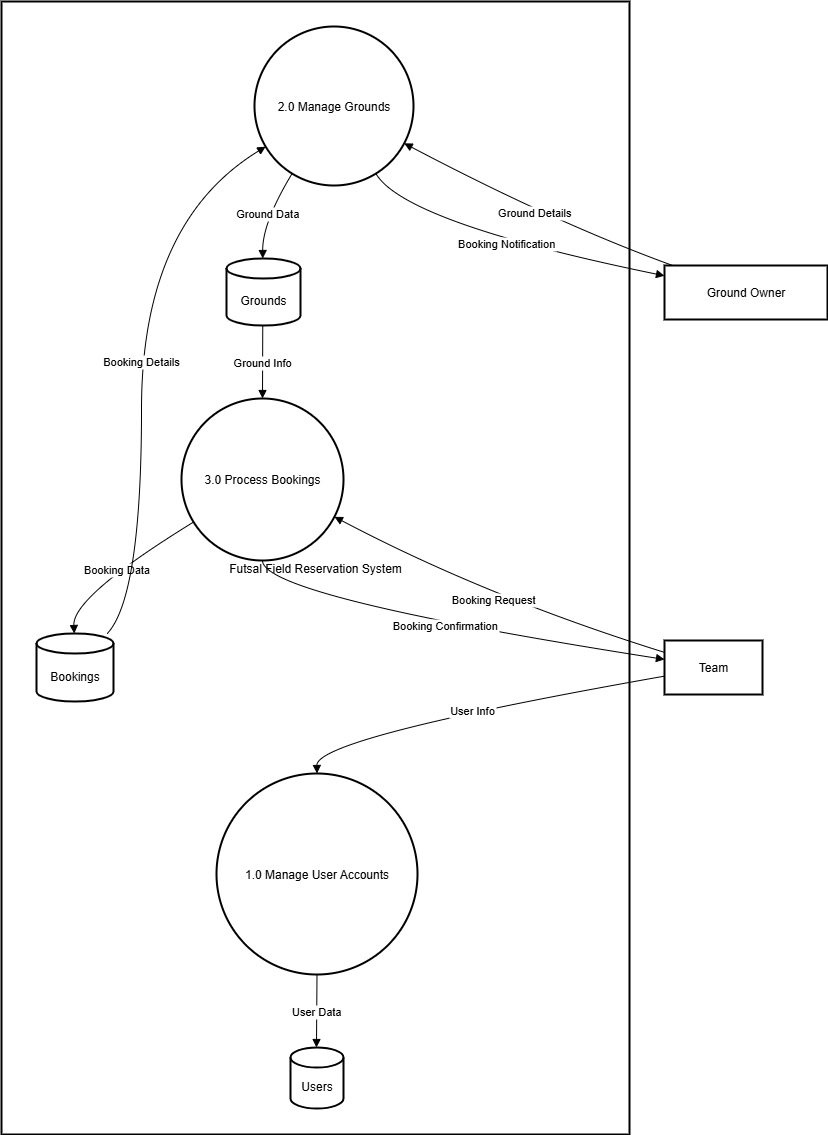
**Figure 4.2.** Activity Diagram

**4.5. Data Flow Diagram:**

Think of a Data Flow Diagram, or DFD, as a map that shows how information travels through our system. It tracks everything from the moment a user signs up to when they get a booking confirmation, showing where the data comes from, how it gets changed, and where it's stored. Unlike other diagrams, a DFD doesn't care about decisions or steps it's purely focused on the journey of the data itself. Since our Futsal system is all about handling data like user details and booking requests using DFDs is essential. We've created a couple of levels of these diagrams to give both a simple overview and a more detailed look at how the data flows.



**Figure 4.3.** DFD Level 0

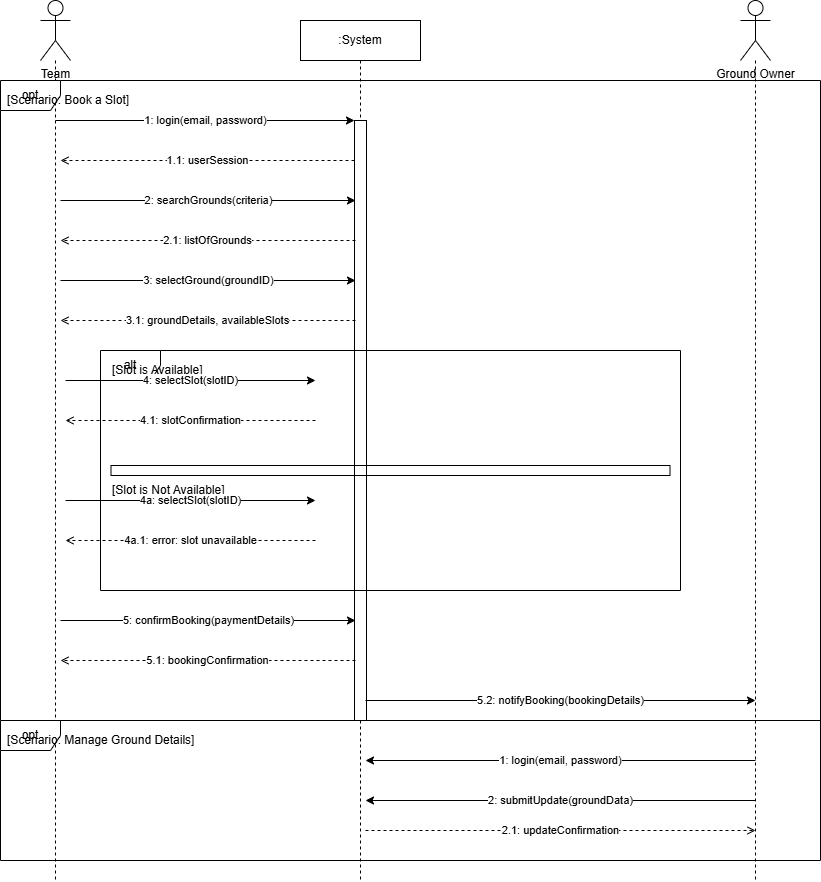


**Figure 4.4.** DFD Level 1

**4.6. System Sequence Diagram**

A System Sequence Diagram (SSD) basically shows the back-and-forth conversation between a user and our system for a specific task, treating the whole system as one black box. We need these diagrams because they turn our written use case stories into a clear visual map, which helps developers see the exact order of events needed to build a feature.

For this project, we're creating an SSD for the "Book Slot" use case. We picked this one because it's the most important and complex interaction a user will have, so getting the sequence of steps perfect is critical.

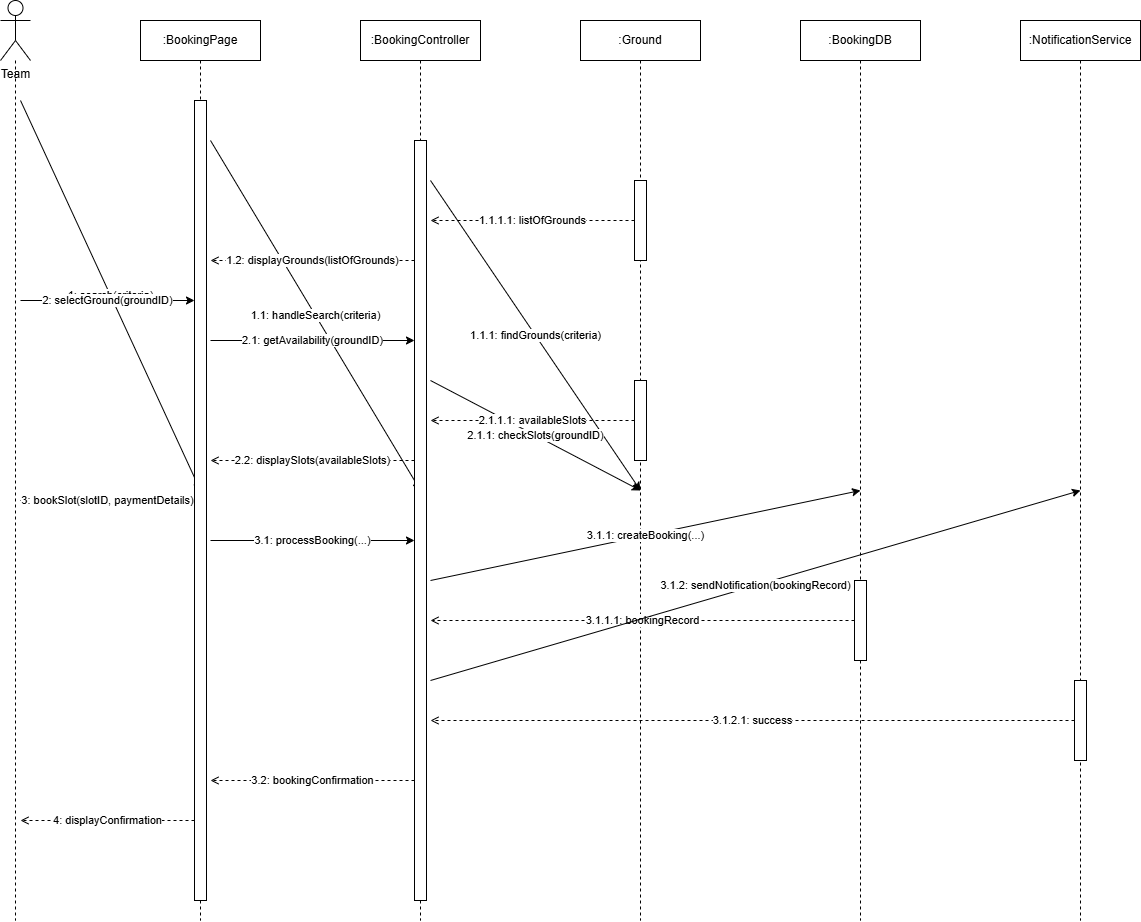


**Figure 4.5.** System Sequence Diagram

**4.7. Sequence Diagram**

While the last diagram showed the conversation with the system from the outside, a Sequence Diagram lets us peek behind the curtain and see how all the internal parts work together. It's a detailed map that shows the step-by-step messages passed between the website, the server, and the database to make something happen.

For this project, we're focusing on the most critical action: a team successfully booking a ground. We chose this because it involves all the key components coordinating perfectly. The diagram will show the entire flow—from the moment a user searches for a ground, to checking its availability, processing the booking, and finally, confirming it.

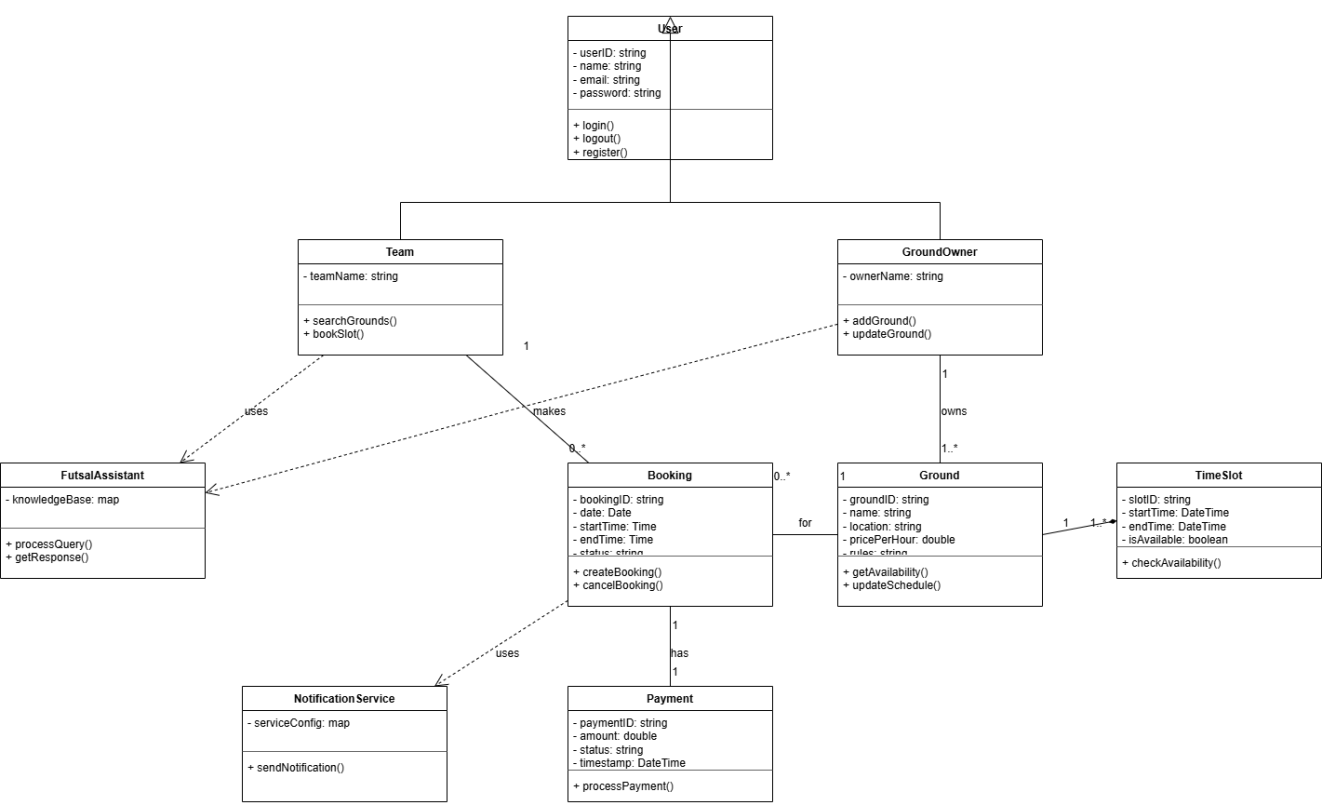


**Figure 4.6.** Sequence Diagram

**4.8. Design Class Diagram**

Think of the Design Class Diagram as the main blueprint for our code. It's essential because it lays out all the system's key components the classes and shows how they're all connected. This gives us a clear and organized map to follow, which is vital for building a scalable and easy-to-maintain app.

For our Futsal system, we've identified the most important classes, like User, Team, Ground, and the Booking class that ties them all together. As you can see in the diagram below, it clearly shows the relationships between these parts for example, how a Team and Ground Owner are specific types of User, and how a Booking links a team to a ground. This blueprint provides a complete overview of our system's structure.



**Figure 4.7.** Design Class Diagram

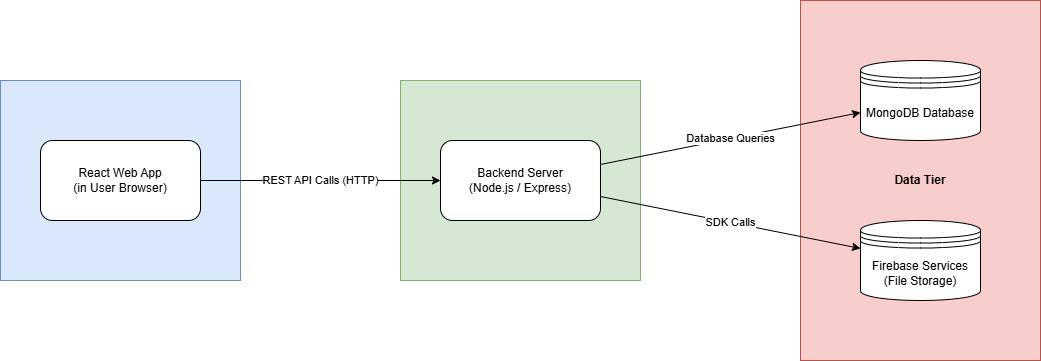
**4.9. Architectural diagrams**

Think of an architectural diagram as the master plan for our software. Its main job is to give us a high-level, big-picture view that makes a complex system easy to understand and helps all the different parts work together smoothly. We create these diagrams to make sure we build a system that is well-organized, can grow in the future, and won't be a headache to maintain. It's the best way to show the overall strategy for how the system is put together.

**4.9.1. Interface Design**

This diagram simply shows how the different parts of our system talk to each other. It maps out the main communication channels, like how your web browser connects to our server, and how our server then connects to the database and other cloud services.

Basically, our React website sends requests to our Node.js server. The server then gets the main data like user info and bookings from our MongoDB database and handles other things like photos using Firebase. Keeping these parts separate ensures that data flows through the system in a clean and organized way.

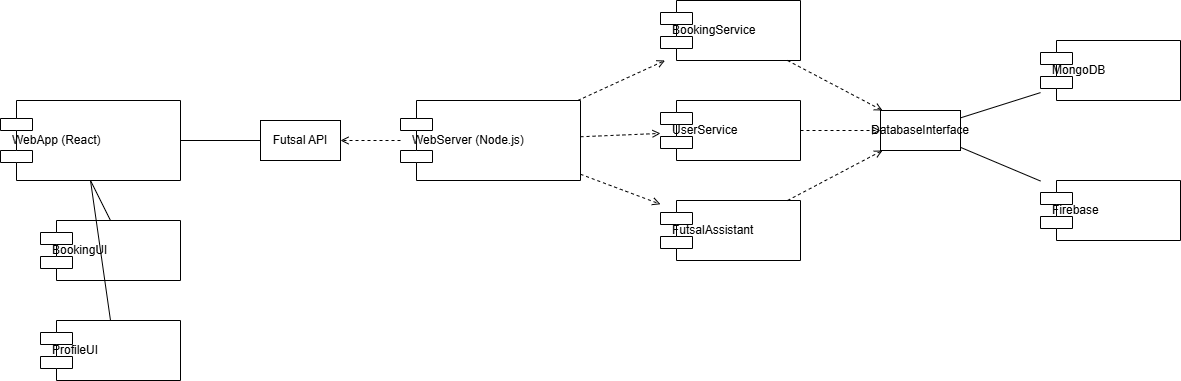


**Figure 4.8.** Interface Design

**4.9.2. Component Level Design**

A component diagram gives us a look "under the hood" at our system's architecture. It breaks the whole application down into its main building blocks and shows how they all fit together. This is important because a well-organized system is much easier to test, update, and understand.

As you can see in the diagram, our website is made up of smaller pieces like the Booking and Profile sections, and it talks to our main Futsal API on the server. The server then uses different services like the Booking Service or User Service to handle specific jobs, and these services connect to the database to get the information they need.



**Figure 4.9.** Component Level Design

**4.9.3. Deployment**

A Deployment Diagram shows us where all the different parts of our software actually live on physical hardware. Think of it as a map of the real-world setup, which is essential for planning our server needs and network configuration. It helps us visualize how the software is distributed across different machines.

Our diagram for this project lays out three main areas. First, there's your device (like a laptop or phone), which runs our website in a browser. That browser talks over the internet to our cloud server, where the main application lives. In turn, the server connects to our database and cloud services, which are handled by MongoDB Atlas and Firebase.

**CHAPTER 5**

**SYSTEM TESTING AND VALIDATION**

This chapter is dedicated to the most critical phase of our project: testing. The main goal here was to put our Futsal Field Reservation System through its paces to ensure it’s robust, reliable, and ready for real-world use. We systematically tested every aspect of the application to find and fix any potential bugs whether in the underlying code, the features themselves, or the user interface. This process was essential to confirm that the system not only works as intended but also delivers a high-quality and dependable experience for every user.

To give you a clear picture of our approach, this chapter is broken down into a few key sections. We’ll begin by outlining our overall test design and strategy. From there, we’ll walk you through the detailed test cases we created for the system's core functions. We will then cover the specific testing methods we used, including White-Box testing to examine the internal code and Black-Box testing to validate the system's functionality from a user's perspective. Finally, we’ll discuss our Graphical User Interface (GUI) testing to ensure the platform is not only functional but also intuitive and easy to navigate.

**5.1. System testing**

System testing is a critical step where we check that the entire, integrated software works as intended and meets all its requirements. It's how we ensure quality, find bugs, and build confidence before launch. For this project, we used a few different techniques. We performed "White-Box" testing to look inside the code and check its logic, "Black-Box" testing to confirm that all the features worked correctly from a user's perspective, and finally, GUI testing to make sure the website was easy to use and looked good.

The great news is that the system passed its tests with flying colors. All the core features like booking a ground, registering a new account, and managing a profile work exactly as they should. This confirms that we've achieved our main goal of giving local players a streamlined way to reserve a pitch. The successful tests also show that we've provided a valuable tool for ground owners to manage their facilities more effectively. Ultimately, the results validate that the system is a user-friendly and reliable platform for the local football community.

**5.2. Testing Techniques**

To make sure our system was solid, we used a mix of testing methods to look at it from all angles. This approach helped us systematically hunt down bugs and confirm everything worked as planned. This section details the two primary testing techniques we used: White-Box Testing and Black-Box Testing.

**5.2.1. White box Testing**

White-Box Testing is where our developers looked "under the hood" at the application's source code. The goal was to inspect the internal logic and structure to ensure everything was working correctly from the inside out.

We focused this effort on the most critical back-end functions like user registration, booking, and payment processing because their internal logic had to be flawless and secure. Our team conducted detailed code reviews and ran tests on specific functions to verify that data was handled correctly and that security measures, like password hashing, were properly implemented.

The results from this testing were very positive. We confirmed that the core engine of our application is robust, secure, and functions exactly as designed, giving us full confidence in the system's internal integrity.

**5.2.2. Black Box Testing**

Black-Box Testing is where we test the application just like a regular user would, without peeking at the code. The whole point is to treat the system as a "black box" and focus only on whether the features work as they should from the outside.

This approach was essential for our project because it allowed us to validate the entire user experience from start to finish. We simulated real-world scenarios like registering an account, searching for a ground, and completing a booking to ensure every feature was intuitive and delivered the right results.

We’re happy to report that the system passed all our black-box tests with flying colors. This confirms that the platform is functionally solid, easy to use, and successfully meets the needs of our users, proving it's ready for deployment.

**5.3. Test Cases**

Think of test cases as a simple checklist we follow to make sure a specific feature is working exactly as it should. They're a core part of testing because they give us a structured way to confirm the system is meeting its most important requirements. We don't try to test every single minor detail; instead, we design our test cases to focus on the most critical, high-risk features to ensure they are absolutely rock-solid.

**Test Case 1: User Registration**

Table 5.1: Test Case 1

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** | | | |
| **Test Stage:** | Unit ☑Functionality Integration System Interface  Performance Regression Acceptance Pilot | | |
| **Test Date:** |  | **System Date, if applicable:** |  |
| **Tester:** | Saadullah | **Test Case Number:** | TC-01 |
| **Test Case Description:** | To verify that a new user (Team or Ground Owner) can successfully create an account on the platform. | | |
| **Results:** | Pass Fail | **Incident Number, if applicable:** |  |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | 01: The System shall allow users to set up their respective accounts.  02: The System shall allow the user to set email for sign up.  FR003: The System shall allow the user to set password for sign up. | | |
| **Roles and Responsibilities:** | The Tester is responsible for executing the test steps and recording the outcome. | | |
| **Set Up Procedures:** | 1. Open the Futsal Field Reservation System website in a compatible browser. 2. Ensure no user is currently logged in.   3. Navigate to the "Sign Up" page. | | |
| **Stop Procedures:** | Close the browser tab after the test is complete. | | |
| **ENVIRONMENTAL NEEDS** | | | |
| **Hardware:** | Standard desktop or laptop computer with an internet connection. | | |
| **Software:** | Google Chrome web browser. | | |
| **Procedural Requirements:** | None. | | |
| **TEST** | | | |
| **Test Items and Features:** | User Registration feature, including input fields for name, email, password, and user type selection. | | |
| **Input Specifications:** | Valid and unique user email, a strong password, and other required profile information. | | |
| **Procedural Steps:** | 1. Fill in all required fields in the registration form. 2. Click the "Submit" button.   3. Observe the system's response. | | |
| **Expected Results of Case:** | The system should validate the inputs, create a new user account in the database, and redirect the user to the login page or their new dashboard, displaying a success message. | | |
| **ACTUAL RESULTS** | | | |
| **Output Specifications:** | The system successfully created the new user account. The user was redirected to their dashboard, and a confirmation was implicitly received. The database correctly reflected the new user entry. The test passed. | | |

**Test Case 2: Accessing Ground Owner Profile**

Table 5.2: Test Case 2

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** | | | |
| **Test Stage:** | Unit ☑Functionality Integration ☑System Interface  Performance Regression Acceptance Pilot | | |
| **Test Date:** | mm/dd/yy | **System Date, if applicable:** | mm/dd/yy |
| **Tester:** | Saadullah | **Test Case Number:** | TC-02 |
| **Test Case Description:** | To verify that a registered Ground Owner can securely log in and access their specific profile dashboard. | | |
| **Results:** | ☑Pass Fail | **Incident Number, if applicable:** | N/A |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | 1: The System shall allow the user to login with correct credentials.  2: The System shall allow the ground owners to provide all theirs and ground details. | | |
| **Roles and Responsibilities:** | The Tester is responsible for executing the test and verifying access to the correct dashboard. | | |
| **Set Up Procedures:** | 1. Ensure a Ground Owner account exists in the system.   2. Navigate to the login page. | | |
| **Stop Procedures:** | Log out of the account and close the browser. | | |
| **ENVIRONMENTAL NEEDS** | | | |
| **Hardware:** | Standard desktop or laptop computer with an internet connection. | | |
| **Software:** | Google Chrome web browser. | | |
| **Procedural Requirements:** | Valid login credentials for a Ground Owner are required. | | |
| **TEST** | | | |
| **Test Items and Features:** | Login functionality, user authentication, and role-based redirection. | | |
| **Input Specifications:** | A registered Ground Owner's email and password. | | |
| **Procedural Steps:** | 1. Enter the Ground Owner's email and password into the login form. 2. Click the "Login" button.   3. Verify that the page redirects to the Ground Owner's dashboard. | | |
| **Expected Results of Case:** | The system should authenticate the user and grant access to the Ground Owner dashboard, where they can manage their facilities. Access to other user-type dashboards (e.g., Team) should be denied. | | |
| **ACTUAL RESULTS** | | | |
| **Output Specifications:** | The user was successfully authenticated and redirected to the correct Ground Owner dashboard. All profile management features were visible and accessible. The test passed. | | |

**Test Case 3: Booking a Ground Slot**

Table 5.3: Test Case 3

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** | | | |
| **Test Stage:** | Unit ☑Functionality Integration ☑System Interface  Performance Regression Acceptance Pilot  Specify the testing stage for this test case. | | |
| **Test Date:** | mm/dd/yy | **System Date, if applicable:** | mm/dd/yy |
| **Tester:** | Saadullah | **Test Case Number:** | TC-03 |
| **Test Case Description:** | To verify the complete, end-to-end workflow of a Team user successfully booking a ground slot. | | |
| **Results:** | ☑Pass Fail | **Incident Number, if applicable:** | N/A |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | 1: The System will provide a view of grounds for booking.  2: The System will allow teams to view grounds and click on whichever they want to book..  3: Teams can book a slot if available at their desired time. | | |
| **Roles and Responsibilities:** | The Tester is responsible for simulating the Team user's actions and verifying the booking confirmation. | | |
| **Set Up Procedures:** | 1. Log in as a registered Team user. 2. Ensure there is at least one ground listed by a Ground Owner with available slots.   3. Navigate to the "Bookings" or "Search Grounds" page. | | |
| **Stop Procedures:** | Log out of the account. The booking can be manually cancelled in the database if needed. | | |
| **ENVIRONMENTAL NEEDS** | | | |
| **Hardware:** | Standard desktop or laptop computer with an internet connection. | | |
| **Software:** | Google Chrome web browser. | | |
| **Procedural Requirements:** | A logged-in Team user session. | | |
| **TEST** | | | |
| **Test Items and Features:** | Ground search, ground details view, availability calendar/slot selection, and booking confirmation process. | | |
| **Input Specifications:** | Selection of a ground, a date, and an available time slot. | | |
| **Procedural Steps:** | 1. Select a ground from the list. 2. On the ground's detail page, select an available date and time slot. 3. Click the "Book Now" button.   4. Proceed through any confirmation steps. | | |
| **Expected Results of Case:** | The system should reserve the selected slot, make it unavailable for other users, record the booking in the database, and display a success confirmation to the Team user. | | |
| **ACTUAL RESULTS** | | | |
| **Output Specifications:** | The slot was successfully booked. A confirmation message was displayed, and the slot was no longer shown as available on the ground's schedule. The test passed. | | |

**Test Case 4: Accessing Teams Profile**

Table 5.4: Test Case 4

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** | | | |
| **Test Stage:** | Unit ☑Functionality Integration ☑System Interface  Performance Regression Acceptance Pilot  Specify the testing stage for this test case. | | |
| **Test Date:** | mm/dd/yy | **System Date, if applicable:** | mm/dd/yy |
| **Tester:** | Ehtisham Naveed | **Test Case Number:** | TC-04 |
| **Test Case Description:** | To verify that a registered Team can securely log in and access their specific profile dashboard. | | |
| **Results:** | ☑Pass Fail | **Incident Number, if applicable:** | N/A |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | 1: The System shall allow the user to login with correct credentials.  2: The System shall provide teams with dedicated profiles.  3: The System shall allow the teams to update their profiles. | | |
| **Roles and Responsibilities:** | The Tester is responsible for executing the test and verifying access to the correct dashboard. | | |
| **Set Up Procedures:** | 1. Ensure a Team account exists in the system.   2. Navigate to the login page. | | |
| **Stop Procedures:** | Log out of the account and close the browser. | | |
| **ENVIRONMENTAL NEEDS** | | | |
| **Hardware:** | Standard desktop or laptop computer with an internet connection. | | |
| **Software:** | Google Chrome web browser. | | |
| **Procedural Requirements:** | Valid login credentials for a Team are required. | | |
| **TEST** | | | |
| **Test Items and Features:** | Login functionality, user authentication, and role-based redirection to the Team dashboard. | | |
| **Input Specifications:** | A registered Team's email and password. | | |
| **Procedural Steps:** | 1. Enter the Team's email and password into the login form. 2. Click the "Login" button.   3. Verify that the page redirects to the Team's dashboard. | | |
| **Expected Results of Case:** | The system should authenticate the user and grant access to the Team dashboard, where they can manage their profile and view bookings. | | |
| **ACTUAL RESULTS** | | | |
| **Output Specifications:** | The user was successfully authenticated and redirected to the correct Team dashboard. All profile management features were visible and accessible. The test passed. | | |

**Test Case 5: Making a Donation**

Table 5.5: Test Case 5

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** | | | |
| **Test Stage:** | Unit ☑Functionality Integration ☑System Interface  Performance Regression Acceptance Pilot  Specify the testing stage for this test case. | | |
| **Test Date:** | mm/dd/yy | **System Date, if applicable:** | mm/dd/yy |
| **Tester:** | Ehtisham Naveed | **Test Case Number:** | TC-05 |
| **Test Case Description:** | To verify that a user can successfully make a monetary contribution through the donation feature. | | |
| **Results:** | ☑Pass Fail | **Incident Number, if applicable:** | N/A |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | 1: The System will provide a donation feature.  2: The System will allow users to make the amount of payment they desire. | | |
| **Roles and Responsibilities:** | The Tester is responsible for navigating to the donation page and completing a test transaction. | | |
| **Set Up Procedures:** | 1. Navigate to the website's "Donation" page.   2. Ensure test payment credentials are available if using a sandbox environment. | | |
| **Stop Procedures:** | Close the browser tab. | | |
| **ENVIRONMENTAL NEEDS** | | | |
| **Hardware:** | Standard desktop or laptop computer with an internet connection. | | |
| **Software:** | Google Chrome web browser. | | |
| **Procedural Requirements:** | None. | | |
| **TEST** | | | |
| **Test Items and Features:** | Donation page UI, payment form integration (e.g., Stripe), and transaction processing. | | |
| **Input Specifications:** | A monetary amount and valid (test) payment details. | | |
| **Procedural Steps:** | 1. Click the "Donate" button on the donation page.   2. Enter a donation amount and payment information into the form. 3. Click the "Pay" or "Submit" button. | | |
| **Expected Results of Case:** | The system should securely process the payment and display a "Thank You" or success confirmation message. | | |
| **ACTUAL RESULTS** | | | |
| **Output Specifications:** | The test transaction was processed successfully, and a confirmation message was displayed to the user. The test passed. | | |

**Test Case 6: Chat-Box**

Table 5.6: Test Case 6

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** | | | |
| **Test Stage:** | Unit ☑Functionality ☑Integration ☑System Interface  Performance Regression Acceptance Pilot  Specify the testing stage for this test case. | | |
| **Test Date:** | mm/dd/yy | **System Date, if applicable:** | mm/dd/yy |
| **Tester:** | Ehtisham Naveed | **Test Case Number:** | TC-06 |
| **Test Case Description:** | To verify that users can communicate with each other in real-time using the integrated chat feature. | | |
| **Results:** | ☑Pass Fail | **Incident Number, if applicable:** | N/A |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | 1: The System shall allow the teams to communicate with other teams.  2: The System will allow communication between users. | | |
| **Roles and Responsibilities:** | Two testers are required, one for each user account, to test the real-time message exchange. | | |
| **Set Up Procedures:** | 1. Log in with two different user accounts (e.g., Team A and Ground Owner B) in two separate browser windows.   2. Navigate to a page where the chat functionality is accessible. | | |
| **Stop Procedures:** | Log out of both accounts. | | |
| **ENVIRONMENTAL NEEDS** | | | |
| **Hardware:** | Two computers or two separate browser sessions on one computer. | | |
| **Software:** | Google Chrome web browser. | | |
| **Procedural Requirements:** | Two active, logged-in user sessions. | | |
| **TEST** | | | |
| **Test Items and Features:** | Chat interface, message input field, send button, and real-time message display. | | |
| **Input Specifications:** | Text messages from both users. | | |
| **Procedural Steps:** | 1. User A initiates a chat with User B. 2. User A types and sends a message. 3. Verify that User B receives the message instantly. 4. User B types and sends a reply.   5. Verify that User A receives the reply instantly. | | |
| **Expected Results of Case:** | For common questions, the chatbot should provide a direct, accurate answer or a link to the relevant page. For unrecognized queries, it should provide a polite message indicating it doesn't understand and offer alternative help options. | | |
| **ACTUAL RESULTS** | | | |
| **Output Specifications:** | Messages should be sent and received between the two users in real-time without significant delay. The chat history should be correctly displayed for both users. | | |

**Test Case 7: Using the Futsal Assistant**

Table 5.7: Test Case 7

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** | | | |
| **Test Stage:** | Unit ☑Functionality ☑Integration ☑System Interface  Performance Regression Acceptance Pilot  Specify the testing stage for this test case. | | |
| **Test Date:** | mm/dd/yy | **System Date, if applicable:** | mm/dd/yy |
| **Tester:** | Ehtisham Naveed | **Test Case Number:** | TC-07 |
| **Test Case Description:** | To verify that a user can interact with the Futsal Assistant chatbot and receive relevant, helpful responses. | | |
| **Results:** | ☑Pass Fail | **Incident Number, if applicable:** | N/A |
| **INTRODUCTION** | **INTRODUCTION** | | |
| **Requirement(s) to be tested:** | The system shall provide an AI chatbot ("Futsal Assistant") to answer user queries.. | | |
| **Roles and Responsibilities:** | The Tester is responsible for interacting with the chatbot and evaluating the relevance and accuracy of its responses. | | |
| **Set Up Procedures:** | 1. Navigate to any page on the website where the Futsal Assistant icon is present. | | |
| **Stop Procedures:** | Close the chat interface. | | |
| **ENVIRONMENTAL NEEDS** | **ENVIRONMENTAL NEEDS** | | |
| **Hardware:** | Standard desktop or laptop computer with an internet connection. | | |
| **Software:** | Google Chrome web browser. | | |
| **Procedural Requirements:** | None. | | |
| **TEST** | **TEST** | | |
| **Test Items and Features:** | Chatbot UI, natural language processing, and response generation. | | |
| **Input Specifications:** | A set of common user queries, such as "How to book a ground?", "What are the prices?", and "I forgot my password." | | |
| **Procedural Steps:** | 1. Click the Futsal Assistant icon to open the chat window. 2. Type a common question into the input field and press Enter. 3. Observe the chatbot's response.   4. Type a more complex or nonsensical query to test its fallback response. | | |
| **Expected Results of Case:** | For common questions, the chatbot should provide a direct, accurate answer or a link to the relevant page. For unrecognized queries, it should provide a polite message indicating it doesn't understand and offer alternative help options. | | |
| **ACTUAL RESULTS** | **ACTUAL RESULTS** | | |
| **Output Specifications:** | The chatbot correctly answered all predefined common questions. When presented with an unknown query, it responded with a helpful fallback message. The test passed. | | |

**5.4. Non-Functional Requirements**

While functional requirements cover what our system does, non-functional requirements (NFRs) are all about how well it does it. Think of them as quality goals, like how fast the pages load, how reliable the system is, or how easy it is to use. It's important to define these with clear, measurable targets—for example, saying "pages must load in under 3 seconds" is much better than just saying "the site should be fast."

This section will break down the key NFRs for our Futsal system and explain how we measured them, using the criteria in the table below as our guide.

**5.4.1. Compatibility**

Compatibility refers to the ability of the software to run on specific hardware, operating systems, applications, or network environments. For a web-based application, browser compatibility is a primary concern.

Table 5.8. Compatibility

|  |  |  |
| --- | --- | --- |
| **Property** | **Measure** | **Target** |
| Compatibility | Number of target systems (browsers). | The system must be fully functional on the latest versions of Google Chrome, Mozilla Firefox, and Microsoft Edge. |

**5.4.2. Usabality**

Usability, or ease of use, measures how easy it is for a user to learn, operate, and get satisfactory results from the system. It is a critical factor for user adoption and satisfaction.

Table 5.9. Usability

|  |  |  |
| --- | --- | --- |
| **Property** | **Measure** | **Target** |
| Ease of use | Training time | A new user should be able to successfully register and book a ground in under 5 minutes without any external training or help documentation. |
| Ease of use | User/event response time | All pages and primary interactive elements (e.g., booking calendar) must load and respond to user input in under 3 seconds. |

**5.4.3. Safety and Security**

Safety and Security requirements are constraints that protect the system and its data from accidental or malicious harm. This includes protecting user data and ensuring that only authorized users can access specific functionalities.

Table 5.10. Safety and Security

|  |  |  |
| --- | --- | --- |
| **Property** | **Measure** | **Target** |
| Security | Password-based authentication | All user accounts must be protected by passwords that are hashed before being stored in the database. |
| Security | Access control | Only authenticated users can access their respective dashboards. A user with a 'Team' role cannot access the 'Ground Owner' dashboard, and vice versa. |
| Robustness | Probability of data corruption on failure | The system must ensure that in the event of a failed booking transaction (e.g., payment failure), no partial or corrupt booking data is saved to the database. The probability should be less than 0.1%. |

**CHAPTER 6**

**CONCLUSION**

This final chapter wraps everything up, summarizing what we've accomplished and looking ahead to what's next. We'll start with some concluding thoughts on the project's success and then discuss potential ideas for future improvements.

Our Futsal Field Reservation System was created to solve a real problem in our local sports community: the lack of a simple, centralized way to book a pitch. We successfully built a platform that streamlines the entire process for both teams and ground owners, getting rid of the old-fashioned hassle of back-and-forth phone calls. By delivering on our core features like easy registration, a live booking calendar, and secure logins we've met our main goals. The project provides a practical solution that makes the futsal experience better for everyone in the twin cities.

While we're proud of what we've built, there's always room to grow. A major next step would be to take the system nationwide, adding grounds from other cities and maybe even other sports. A dedicated mobile app for Android and iOS is another key enhancement we'd love to tackle, as it would make booking on the go much more convenient. Looking further ahead, we could add advanced features like tournament management and detailed analytic for owners, transforming the platform from a simple booking tool into a true hub for the national football community.

**6.1. Conclusion**

We built the Futsal Field Reservation System to fix the messy and inefficient way people book grounds in the twin cities. Our project delivers a single, easy-to-use platform that connects teams directly with ground owners, making the whole process much smoother. With key features like user profiles, a live booking engine, and our smart Futsal Assistant, we've hit our main goal. Successful testing proves that our system is a reliable solution that finally gets rid of the old booking headaches and makes the local football scene more organized and accessible for everyone.

**6.2. Limitations and Future work**

While we're really proud of what our Futsal system can do, we also see its current limits as a clear roadmap for the future. Right now, its biggest drawback is that it’s only for futsal in the Rawalpindi and Islamabad area, which means we're missing out on a huge number of players across the country. On top of that, it’s currently just a website, and we know that a dedicated mobile app offers a much more convenient and rich experience.

With that in mind, we have a few key recommendations for what comes next. First and foremost, we believe the platform should go national, expanding to other major cities across Pakistan. To welcome even more players, we'd also love to add other popular sports like cricket, badminton, and basketball. Finally, building dedicated mobile apps for both Android and iOS is a top priority. An app would not only make booking easier for everyone but would also let us add powerful features like push notifications for game reminders, making it a truly essential tool for the entire sports community.

**APPENDIX-I**

**User Manual: Futsal Field Reservation System**

**1. Introduction**

This manual provides a comprehensive guide for users of the Futsal Field Reservation System. It covers all the necessary steps for registering an account, logging in, and utilizing the platform's core features. The manual is divided into sections based on user type: Team and Ground Owner, and also includes instructions for using the Futsal Assistant.

**2. Getting Started: Account Registration**

To access the system, all users must first create an account.

1. Navigate to the Website: Open your web browser and go to the homepage of the Futsal Field Reservation System.
2. Select User Type: Click on the "Sign Up" button. You will be prompted to register as either a 'Team' or a 'Ground Owner'.
3. Fill in the Details:

* For Teams: Provide your Team Name, Captain's Name, Email, Password, Mobile Number, and Address.
* For Ground Owners: Provide the Ground's Name, Owner's Name, Email, Password, Mobile Number, and the Ground's Address.

1. Submit the Form: After filling in all the required information, click the "Submit" button. Your account will be created, and you will be redirected to the login page.

**3. Logging In**

Once you have a registered account, you can log in to access your dashboard.

1.Click the "Login" button on the homepage.

2. Enter the email and password you used during registration.

3. Click "Login". You will be redirected to your personalized dashboard based on your user type.

**4. For Team Users**

As a Team user, you can search for grounds, book slots, and manage your profile.

**4.1 Searching for and Booking a Ground**

1. Navigate to Bookings: From your dashboard or the main navigation bar, click on "Bookings" or "Grounds".

2. View Grounds: A list of all available futsal grounds will be displayed. You can use the search bar to find a specific ground by name.

3. Select a Ground: Click on the ground you are interested in to view its details, including location, price, and photos.

4. Check Availability: The ground's detail page will show a calendar with available time slots.

5. Book a Slot: Click on a desired date and time slot, then click the "Book Now" button.

6. Confirm Booking: Follow the on-screen instructions to confirm your booking. This may involve a payment step. Once completed, you will receive a booking confirmation.

1. **For Ground Owner Users**

As a Ground Owner, you can manage your facility's listing, view bookings, and update your schedule.

**5.1 Managing Your Ground Details**

1. Access Your Dashboard: After logging in, you will land on your Ground Owner dashboard.

2.Add/Edit Ground: You will have options to "Add a New Ground" or "Edit" an existing one.

3.Update Information: Fill in or update the form with your ground's name, location, price per hour, rules, and upload new photos.

4.Set Availability: Use the calendar tool to set or block out the time slots that are available for booking.

5.Save Changes: Click "Save" or "Update" to make your changes live on the platform.

**5.2. Viewing Booking**

Your dashboard will display a list or calendar view of all upcoming and past bookings for your facility. You will receive a notification when a new booking is made.

1. **Using the Futsal Assistant**

The Futsal Assistant is an AI-powered chatbot available to all users for instant help.

1. Open the Assistant: Click on the chat icon, usually located in the bottom-right corner of the screen.
2. Ask a Question: The chat window will open. Type your question into the text box (e.g., "How do I cancel a booking?", "What are the payment methods?").
3. Receive an Answer: The assistant will process your question and provide an immediate response. If it cannot answer, it will provide guidance on how to contact human support.

**REFERENCES**

1. Deploying A Free Cluster, [Online] Available: [https://www.mongodb.com/docs/atlas/tutorial/deploy-free-tier-cluster/#deploy-a-free-cluster](https://www.mongodb.com/docs/atlas/tutorial/deploy-free-tier-cluster/%23deploy-a-free-cluster).
2. Uploading Images & Files To Firebase Storage In React JS | Node JS | MongoDB [Online] Available: <https://dev.to/cyberwolve/uploading-images-files-to-firebase-storage-in-react-js-node-js-mongodb-express-ngi>
3. Building a Real-Time Chat App with React and Firebase,[Online] Available:[https://css-tricks.com/building-a-real-time-chat-app-with-react-and-firebase/#top-of-site](https://css-tricks.com/building-a-real-time-chat-app-with-react-and-firebase/%23top-of-site)
4. BlockedbyCORSpolicy:The'Access-Control-Allow-Origin'-Mean Stack,[Online]Available:<https://www.digitalocean.com/community/questions/blocked-by-cors-policy-the-access-control-allow-origin-mean-stack> .