

VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY
SOFTWARE ENGINEERING FACULTY



INTRODUCTION TO
SOFTWARE ENGINEERING

PROJECT
FOOTBALL FIELD MANAGEMENT

Lecturer: Ms. Thai Thuy Han Uyen

Class: SE104.N23.PMCL

Students: Nguyen Thanh Phuc

20521769

Table of Contents

Chapter 1. Introduction	1
1.1 Introduction.....	1
1.2 Reasons for choosing.	1
Chapter 2: Technology	2
2.1 Introduction:.....	2
2.2 React Native:.....	2
2.3 Benefits of React Native:	2
2.3.1 Cross-Platform Development:.....	2
2.3.2 Reusability of Code:.....	2
2.3.3 Native-like Performance:	2
2.3.4 Developer Productivity:	2
2.3.5 Vibrant Community and Ecosystem:	3
2.4 Justification for Choosing React Native:	3
2.4.1 Cross-Platform Compatibility:.....	3
2.4.2 Efficiency and Time Savings:	3
2.4.3 Native-like Performance:	3
2.4.4 Strong Community Support:	3
Chapter 3. Analysis	4
3.1 List of Requirements.....	4
3.2 Description.....	4
3.2.1 Account	4
3.2.2 Booking.....	5
3.2.3 Statistical.....	7
Chapter 4. Use Case Diagram	8
4.1 General use case.....	8
4.2 Use Case Description.....	8
4.2.1 Login/Logout	8
4.2.2 Create/edit/delete Field	9
4.2.3 View detail revenue.....	10
Chapter 5. Class diagram	12
Chapter 6. Activity diagrams	13
Chapter 7. Database	15

Chapter 8. Interface.....	16
Chapter 9. Installation.....	26
Chapter 10. Testing.....	28
10.1 Testing for the Login/Logout Application Function:	28
10.2 Create/Edit/Delete Field feature:	28
10.3 View Detail Revenue functionality:	29
Chapter 11. Conclusion.....	30
11.1 Final result.....	30
11.2 Advantages:.....	30
11.3 Disadvantages:	30
11.4 Future Developments:	31
Bibliography	32

Image Catalog

<i>Picture 1: General use case.....</i>	<i>8</i>
<i>Picture 2: Class diagram.....</i>	<i>12</i>
<i>Picture 3: Add field.....</i>	<i>13</i>
<i>Picture 4: Booking field.....</i>	<i>14</i>
<i>Picture 5: Database.....</i>	<i>15</i>
<i>Picture 6: Login screen.....</i>	<i>16</i>
<i>Picture 7: Register screen.....</i>	<i>17</i>
<i>Picture 8: Main screen.....</i>	<i>18</i>
<i>Picture 9: Add field.....</i>	<i>19</i>
<i>Picture 10: Booking screen.....</i>	<i>20</i>
<i>Picture 11: Booked screen</i>	<i>21</i>
<i>Picture 12: Detail booked.....</i>	<i>22</i>
<i>Picture 13: Chart screen.....</i>	<i>23</i>
<i>Picture 14: Setting screen</i>	<i>24</i>
<i>Picture 15: Available field</i>	<i>25</i>

Chapter 1. Introduction

1.1 Introduction

In recent years, the rapid advancement of technology has revolutionized various aspects of our lives. One area greatly impacted by technological innovations is sports management. With the growing popularity of sports and the increasing demand for efficient organization and administration, the development of digital solutions has become essential. In this chapter, we will explore the background and context of developing a mobile application for managing football pitches, highlighting the need for such a solution in the current sports landscape.

1.2 Reasons for choosing.

The decision to choose the topic of developing an app for managing football pitches is justified by the growing demand for efficient management systems, the inefficiencies in traditional methods, the convenience and accessibility for users, the potential for improved facility management, and the technological advancements in mobile applications. This research aims to provide a solution that revolutionizes pitch management and enhances the experience for players and facility owners.

Chapter 2: Technology

2.1 Introduction:

In this chapter, we will provide an overview of the technology used in developing the mobile application for managing football pitches. Specifically, we will introduce React Native, a popular framework for building cross-platform mobile applications. We will discuss the benefits and features of React Native and explain why it was chosen for this project.

2.2 React Native:

React Native is an open-source framework developed by Facebook for building mobile applications using JavaScript. It allows developers to create high-performance, native-like applications for both iOS and Android platforms from a single codebase. React Native utilizes a combination of JavaScript and native platform components to deliver a seamless and efficient user experience.

2.3 Benefits of React Native:

There are several advantages to using React Native for mobile app development:

2.3.1 Cross-Platform Development:

One of the major benefits of React Native is its ability to create applications that can run on multiple platforms, such as iOS and Android. By using a single codebase, developers can save time and effort compared to developing separate applications for each platform.

2.3.2 Reusability of Code:

React Native allows for code reuse, enabling developers to write once and deploy on multiple platforms. This not only improves development efficiency but also simplifies maintenance and updates as changes made to the codebase are automatically applied to both platforms.

2.3.3 Native-like Performance:

With React Native, developers can build applications that provide a native-like experience to users. The framework uses native components, allowing the app to leverage the device's capabilities and deliver smooth animations, fast loading times, and optimal performance.

2.3.4 Developer Productivity:

React Native offers a simplified development process by using JavaScript, a widely adopted programming language. This makes it easier for developers to learn and work with React Native, resulting in increased productivity and faster development cycles.

2.3.5 Vibrant Community and Ecosystem:

React Native has a large and active community of developers, which means there is extensive support, documentation, and third-party libraries available. This vibrant ecosystem provides developers with access to a wide range of resources and tools to enhance their development process.

2.4 Justification for Choosing React Native:

The decision to use React Native for developing the mobile application for managing football pitches is based on the following reasons:

2.4.1 Cross-Platform Compatibility:

By utilizing React Native, we can ensure that the application is compatible with both iOS and Android platforms. This allows us to reach a larger user base and provide a consistent experience across different devices.

2.4.2 Efficiency and Time Savings:

React Native's code reuse capabilities significantly reduce development time and effort. By sharing a single codebase, we can develop and maintain the application more efficiently, resulting in cost savings and quicker time-to-market.

2.4.3 Native-like Performance:

React Native's ability to render native components provides a seamless user experience with fast performance and smooth animations. This ensures that the application performs well and meets user expectations.

2.4.4 Strong Community Support:

React Native has a large and active community of developers, which means there is ample support, resources, and third-party libraries available. This community support ensures that we have access to a wealth of knowledge and tools to overcome any challenges during development.

Chapter 3. Analysis

3.1 List of Requirements

ID	Name of re	Priority	Diagram	Test case
1. Account				
1.1	Login/Logout application	High		
1.2	Register account	High		
2. Booking				
2.1	Create/edit/delete field	High		
2.2	Booking/Cancel field	Medium		
2.3	View field available	Medium		
2.4	View booked field, cancel	Medium		
3. Statistical				
3.1	View detail revenue	High		

3.2 Description

3.2.1 Account

+ Login/Logout application:

Firebase Authentication is a user authentication service provided by Google Firebase. Using Firebase Authentication in a soccer field booking app makes managing users and account authentication easier and safer.

The login function of the soccer field booking app using Firebase Authentication allows users to log in to the app using their accounts. Users can use their Google or Facebook accounts, or register a new account directly in the app. After logging in, users can access the app's features, such as booking soccer fields, managing reservations, and viewing their profile.

The logout function of the app allows users to log out of the app and end their current session. This is important for security reasons and to ensure that users' accounts are not accessed by unauthorized individuals. When the user logs out, they are redirected to the login page where they can log in again if they want to use the app.

+ Register account

Using the Register account Firebase Authentication feature with the requirement of a minimum password of 8 characters including lowercase, uppercase, number, and special characters helps make the soccer field booking app more secure and user-friendly.

With Firebase Authentication, users can easily create an account with a valid email address and password that meets the specified requirements. This ensures that user passwords are strong and not easily guessed, making it difficult for attackers to gain unauthorized access to their accounts.

In addition to providing secure account creation, Firebase Authentication also allows for easy integration with other authentication providers such as Google and Facebook. This makes it convenient for users to sign up and log in to the app using their existing social media accounts, reducing the friction of creating and remembering another set of login credentials.

Furthermore, with Firebase Authentication, the soccer field booking app can offer additional features such as email verification and two-factor authentication, which can further increase the security of user accounts.

Overall, the Register account Firebase Authentication feature with the requirement of a minimum password of 8 characters including lowercase, uppercase, number, and special characters adds an extra layer of security and convenience to the soccer field booking app, making it a more robust and user-friendly platform.

3.2.2 Booking

+ Create/edit/delete Field

The Create/Edit/Delete Field feature is an important functionality in a soccer field booking app that allows field owners to easily manage their fields using an Expo SQLite database.

Using this feature, field owners can create new fields by providing basic information such as the name of the field, type of field, price, and operating hours. The information is then stored in the Expo SQLite database, allowing customers to easily search for and book fields on the app.

The Edit Field feature allows field owners to modify the information of their previously created fields. They can easily change the name of the field, type of field, price, or operating hours, and the updated information will be stored in the Expo SQLite database in real-time. This ensures that field information is always up-to-date and accurate.

Furthermore, the Delete Field feature allows field owners to remove a field from the system. This is useful in cases where field owners want to stop operating a field or change their business operations. The

information about the field is completely removed from the Expo SQLite database, ensuring that no unnecessary information is stored in the system.

In summary, the Create/Edit/Delete Field feature in a soccer field booking app using an Expo SQLite database is an important functionality that helps field owners manage their fields in an easy and convenient way. It increases flexibility in business management and ensures that field information is always up-to-date.

+ Booking/Cancel Field

The Booking/Cancel Field feature is an important function in a soccer field booking app, allowing customers to book a field and field owners to manage their fields using Expo SQLite.

Customers can easily book a field by selecting a field and the desired time slot on the app. This booking information will be stored in the app's SQLite database and a notification will be sent to the field owner to confirm the booking.

When a customer cancels a booking, the field owner can cancel the booking and the time slot will become available again. Field owners can use this feature to manage their fields and switch between customers more easily.

Additionally, if a customer does not show up or arrives late, the field owner can cancel the booking and use the time slot for another customer. This feature helps field owners manage their fields more efficiently, reduces the number of empty fields, and increases revenue.

In summary, the Booking/Cancel Field feature is an important function in a soccer field booking app, allowing customers to book a field and field owners to manage their fields efficiently using Expo SQLite. It also helps increase revenue for field owners and improves the user experience.

+ View field available

The View Field Available feature is an important component of a football field booking app, allowing field managers to easily see which fields are available for booking. This feature is typically implemented using a database such as SQLite to store field information and is designed to be user-friendly and intuitive.

The View Field Available feature displays all fields that are currently available for booking, with filters to allow the user to search by name, type of field, price, and time. This allows the user to find the field quickly and easily they are looking for, without having to scroll through a long list of options.

In addition, the View Field Available feature should also display important information about each field, such as the field name, type, price, and availability. This information should be updated in real-time, so that the user can see which fields are available for booking at any given moment.

To ensure that the View Field Available feature is as user-friendly as possible, it should be designed with a simple and intuitive interface. This may include features such as search filters, sorting options, and the ability to save searches for future use.

Overall, the View Field Available feature is an important part of any football field booking app and is essential for ensuring that the user can quickly and easily find the field they need. With the right design and functionality, this feature can help to streamline the booking process and improve the user experience.

3.2.3 Statistical

+ View detail revenue

The Statistical View Detail Revenue feature is an important functionality of a football field booking app that allows field managers to view and analyze revenue data through intuitive charts and graphs. This feature is designed to help field managers make informed business decisions by providing them with a comprehensive overview of their revenue streams.

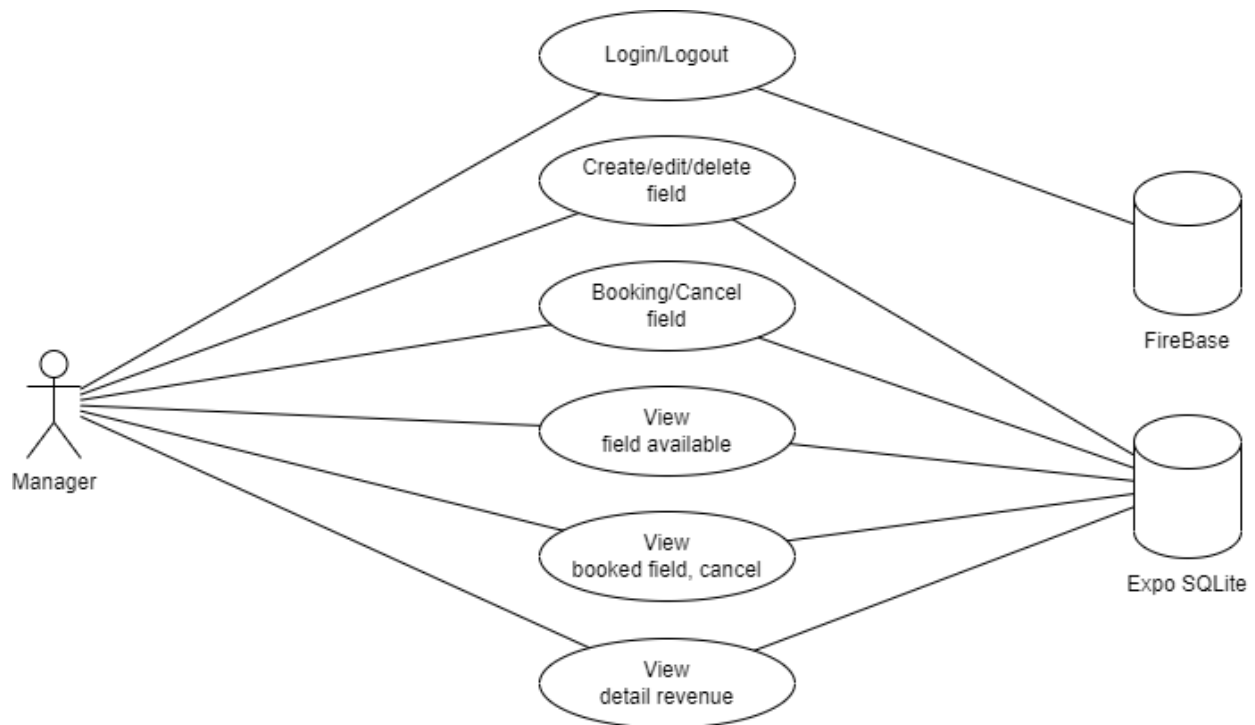
With this feature, the field manager can view revenue data for a particular time period, such as a week, a month, or a year. The data can be displayed in various formats, including line charts, bar graphs, and pie charts. The charts and graphs should be interactive, allowing the field manager to zoom in on specific data points and explore the data in greater detail.

In addition to revenue data, this feature may also provide information on other metrics such as booking frequency, customer demographics, and peak booking times. This information can help the field manager identify trends and patterns in customer behavior, which can inform future business strategies and marketing efforts.

Overall, the Statistical View Detail Revenue feature provides field managers with a valuable tool for understanding their business performance and making data-driven decisions. By providing clear and concise visualizations of revenue data, this feature can help field managers optimize their revenue streams and grow their business over time.

Chapter 4. Use Case Diagram

4.1 General use case



Picture 1: General use case

4.2 Use Case Description

4.2.1 Login/Logout

Nr.	Section	Content/Explanation
1	Name	Login/Logout
2	Authors	Thanh Phuc
3	Priority	
4	Person responsible	
5	Description	The login/logout use case is used to authenticate a user and allow them to access a system. It involves verifying the user's credentials, such as their username and password, and granting them access to the system. Once the user has finished using the system, they can choose to log out, which will terminate their session and revoke their access.
6	Actors	Manager.

7	Pre-conditions	The user must have a valid account in the system. The system must be operational.
8	Post conditions	
9	Main scenario	<ol style="list-style-type: none"> 1. The user enters their username and password into the login form and clicks "Login". 2. The system verifies the user's credentials and grants them access to the system. 3. The user is presented with the system interface and can perform actions within the system. 4. When the user is finished using the system, they click the "logout" button in Setting Screen. 5. The system terminates the user's session and revokes their access to the system, return Login Screen.
10	Exception scenarios	If the user enters incorrect login credentials, the system displays an error message and prompts the user to try again.

4.2.2 Create/edit/delete Field

Nr.	Section	Content/Explanation
1	Name	Create/edit/delete Field
2	Authors	Thanh Phuc
3	Priority	High
4	Person responsible	
5	Description	This use case involves creating, editing, or deleting a field in a system or application.
6	Actors	Manager.
7	Pre-conditions	<p>The user is logged in to the system or application.</p> <p>The user has the necessary permissions to create, edit, or delete fields.</p> <p>The system or application is running and available.</p>

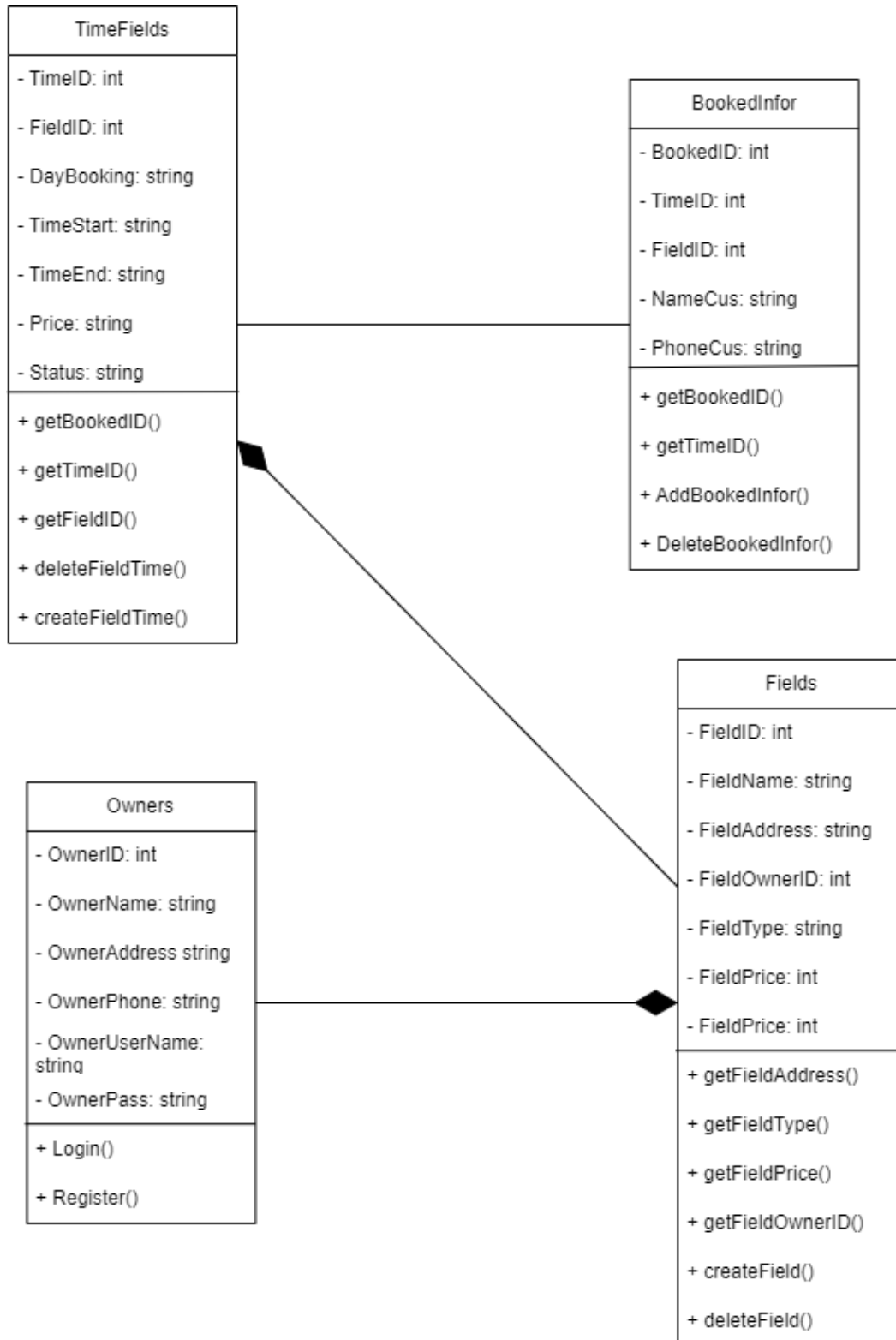
8	Post conditions	<p>A new field is created, edited, or deleted from the system or application.</p> <p>The data in the system or application is updated to reflect the changes made to the field.</p>
9	Main scenario	<ol style="list-style-type: none"> 1. The Manager navigates to the field management section of the system or application. 2. The Manager clicks on the "Create Field" button. 3. The system displays a dialog box to gather the necessary details for the new field, such as name, size, type, and location. 4. The Manager enters the required information in the dialog box. 5. The Manager reviews the entered information and clicks the "Save" button. 6. The system validates the input and saves the new field information to the database. 7. The system confirms the successful creation of the field and updates the list of available fields to include the newly created field.
10	Exception scenarios	<ul style="list-style-type: none"> - Invalid Information: If the Manager provides invalid or incomplete information in the dialog box, the system displays an error message and prompts the Manager to correct the information before saving. - Database Error: If there is an issue with saving the field information to the database, the system displays an error message and informs the Manager about the problem. The Manager may need to retry or contact the system administrator for assistance.

4.2.3 View detail revenue

Nr.	Section	Content/Explanation
-----	---------	---------------------

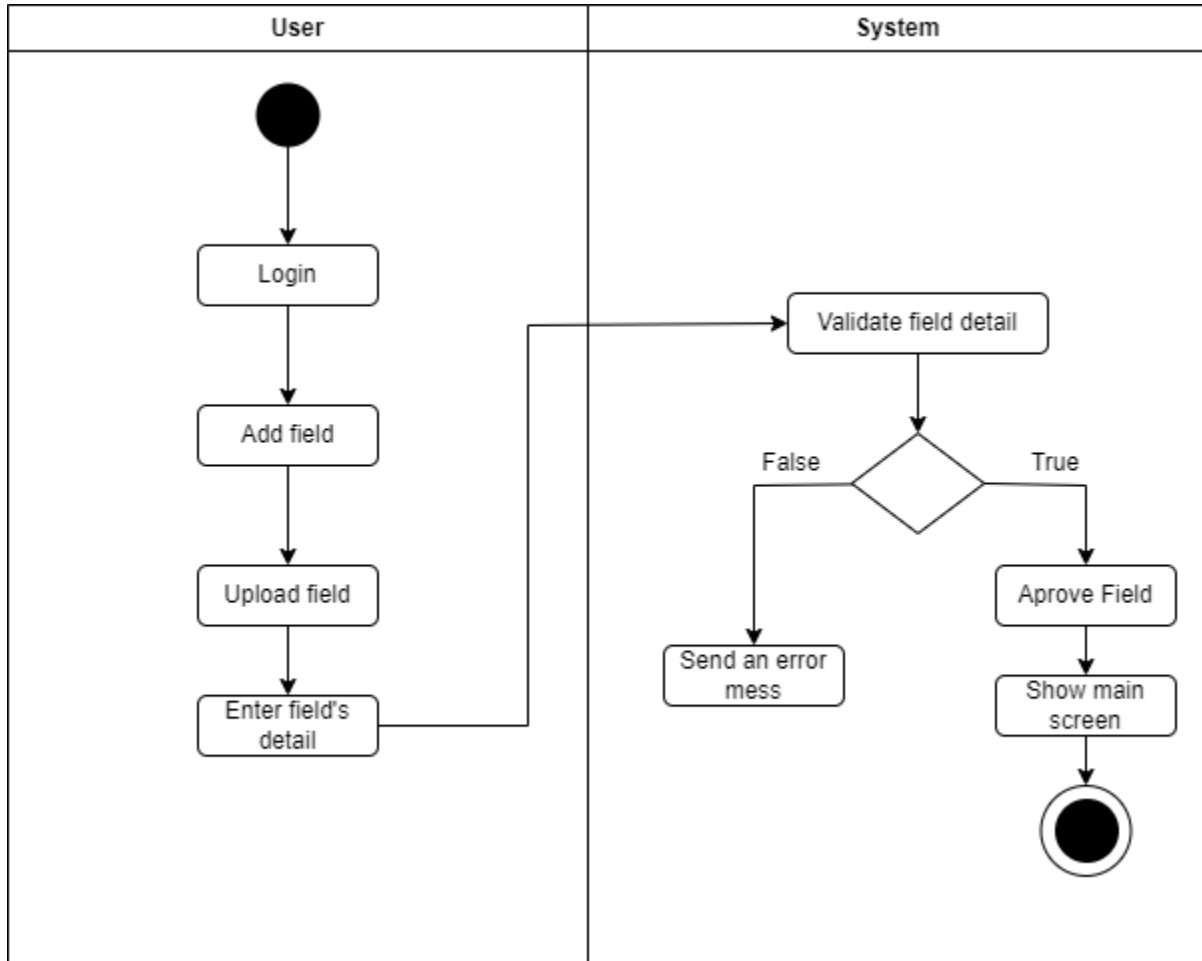
1	Name	View detail revenue
2	Authors	Thanh Phuc
3	Priority	High
4	Person responsible	
5	Description	This use case involves viewing detailed revenue information in the system or application. It allows the user to access specific revenue details, such as earnings, expenses, and profit, for a given period or category.
6	Actors	Manager
7	Pre-conditions	The user is logged in to the system or application. The user has the necessary permission to view revenue details. The system or application is running and available. Revenue data is available in the system or database.
8	Post conditions	The user can view the detailed revenue information.
9	Main scenario	User use navigates to change revenue screen.
10	Exception scenarios	No

Chapter 5. Class diagram

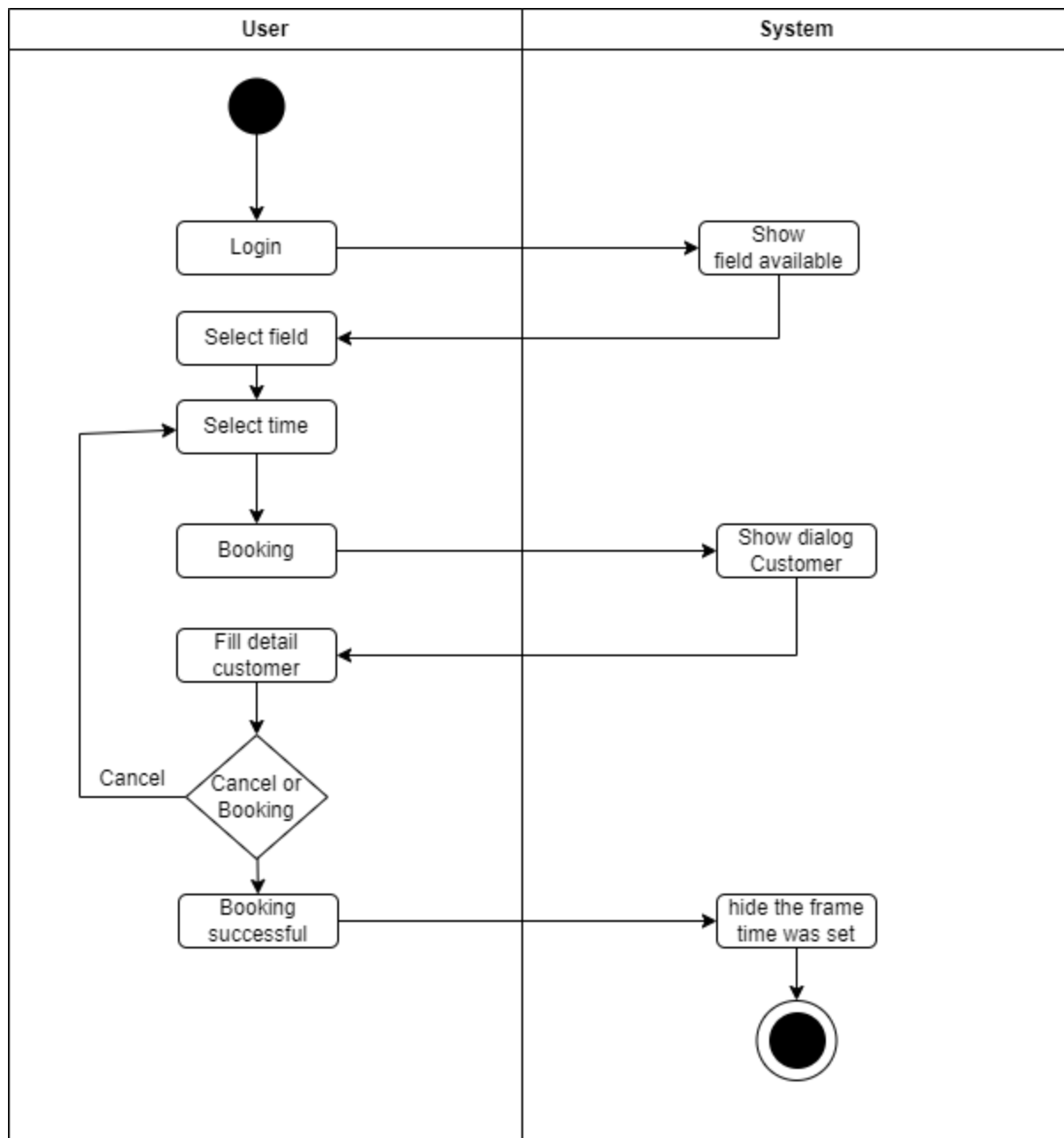


Picture 2: Class diagram

Chapter 6. Activity diagrams

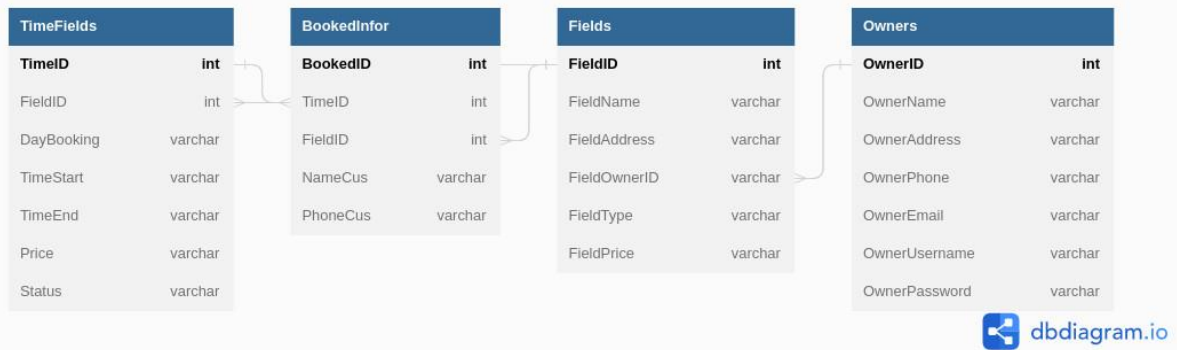


Picture 3: Add field







Picture 4: Booking field

Chapter 7. Database




Picture 5: Database

Chapter 8. Interface

12:59   16.0 KB/S  4G  43

Already have an Account?





Email:
nguyenphuc2002ok@gmail.com


Password:
.....

Login

Register

Use other methods ?

2 AsyncStorage has been extracted from... 

Picture 6: Login screen

Already have an Account?



Email:

nguyenphuc2002ok@gmail.com

Password:

.....

Retype password:

.....

Register

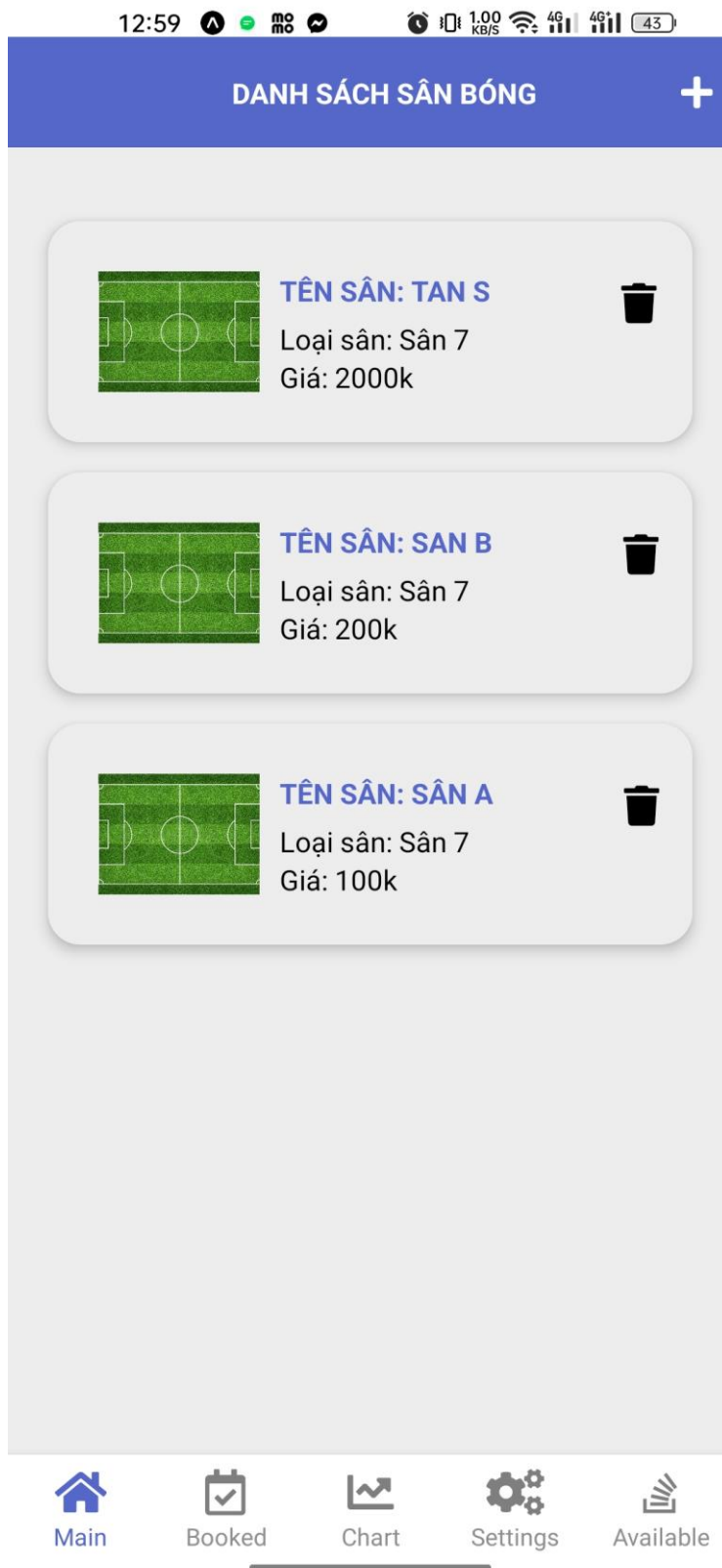
Use other methods ?



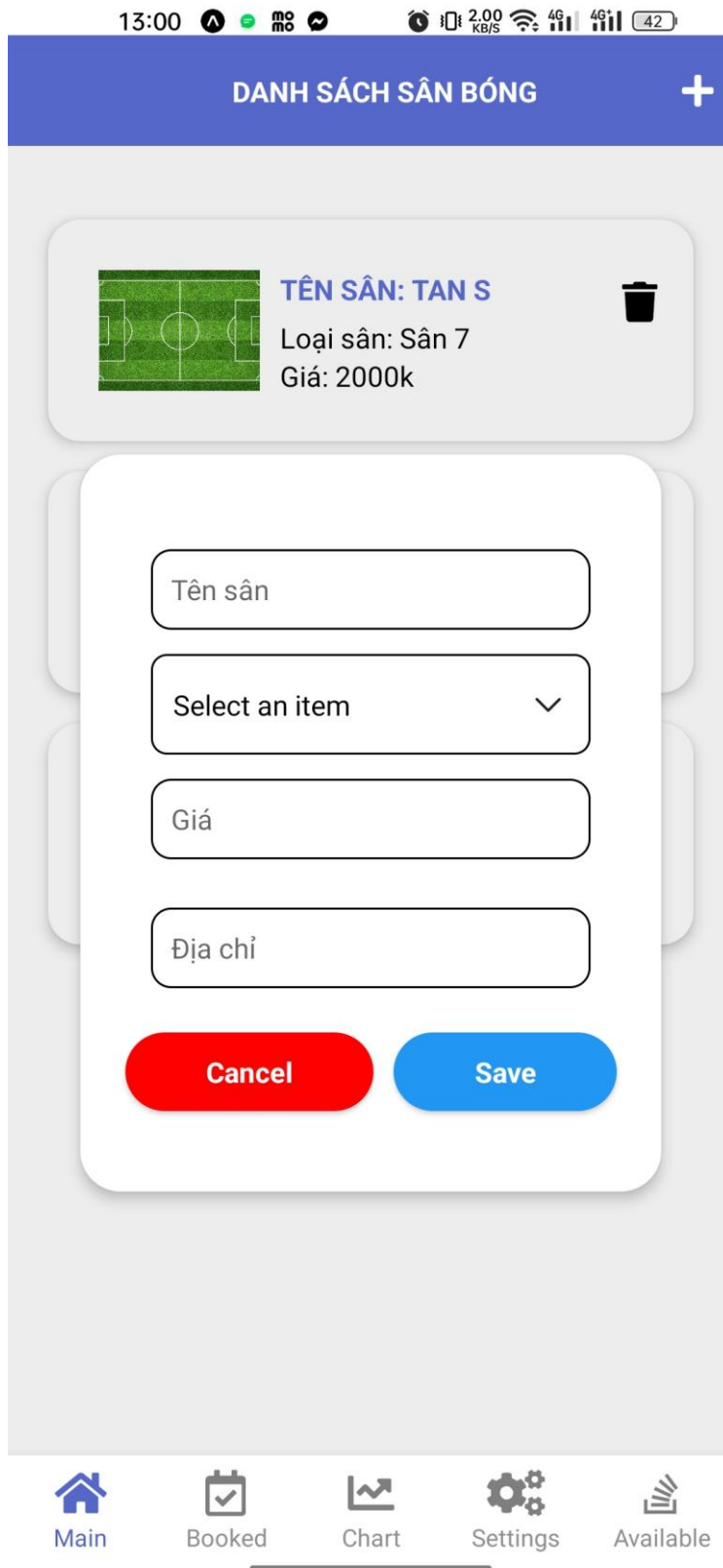
AsyncStorage has been extracted from...



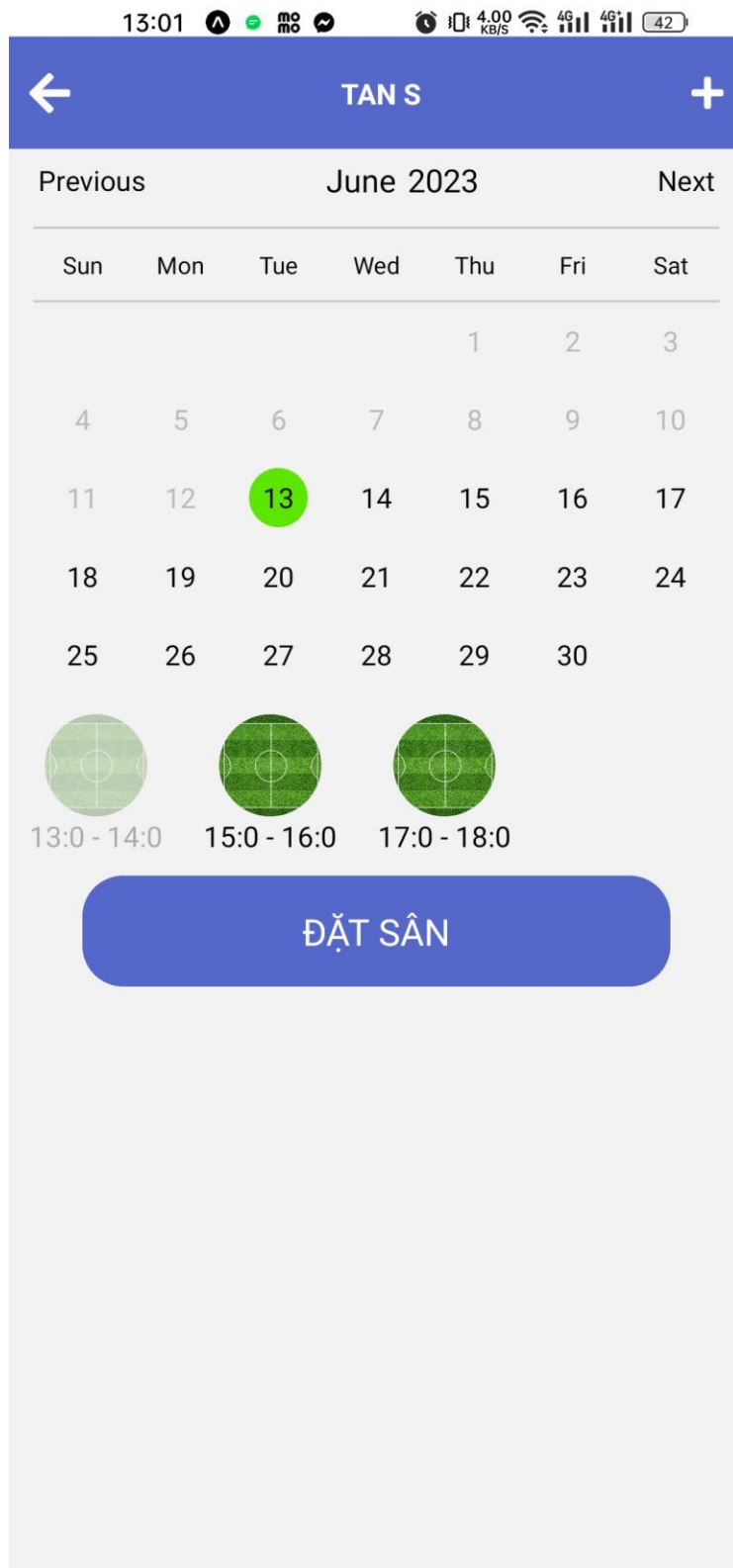
Picture 7: Register screen



Picture 8: Main screen



Picture 9: Add field



Picture 10: Booking screen

DANH SÁCH SÂN ĐÃ ĐẶT

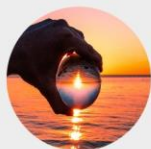


Tên: Phuc

Tên sân: Tan s

Số điện thoại: 046464991

Giá: 2000

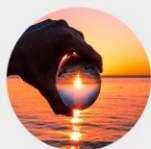


Tên: Thanh phuc

Tên sân: Tan s

Số điện thoại: 094346464

Giá: 2000



Tên: Nguyen van b

Tên sân: Tan s

Số điện thoại: 0946494319

Giá: 2000



Tên: Khang

Tên sân: San b

Số điện thoại: 0464843194

Giá: 200



Main



Booked



Chart

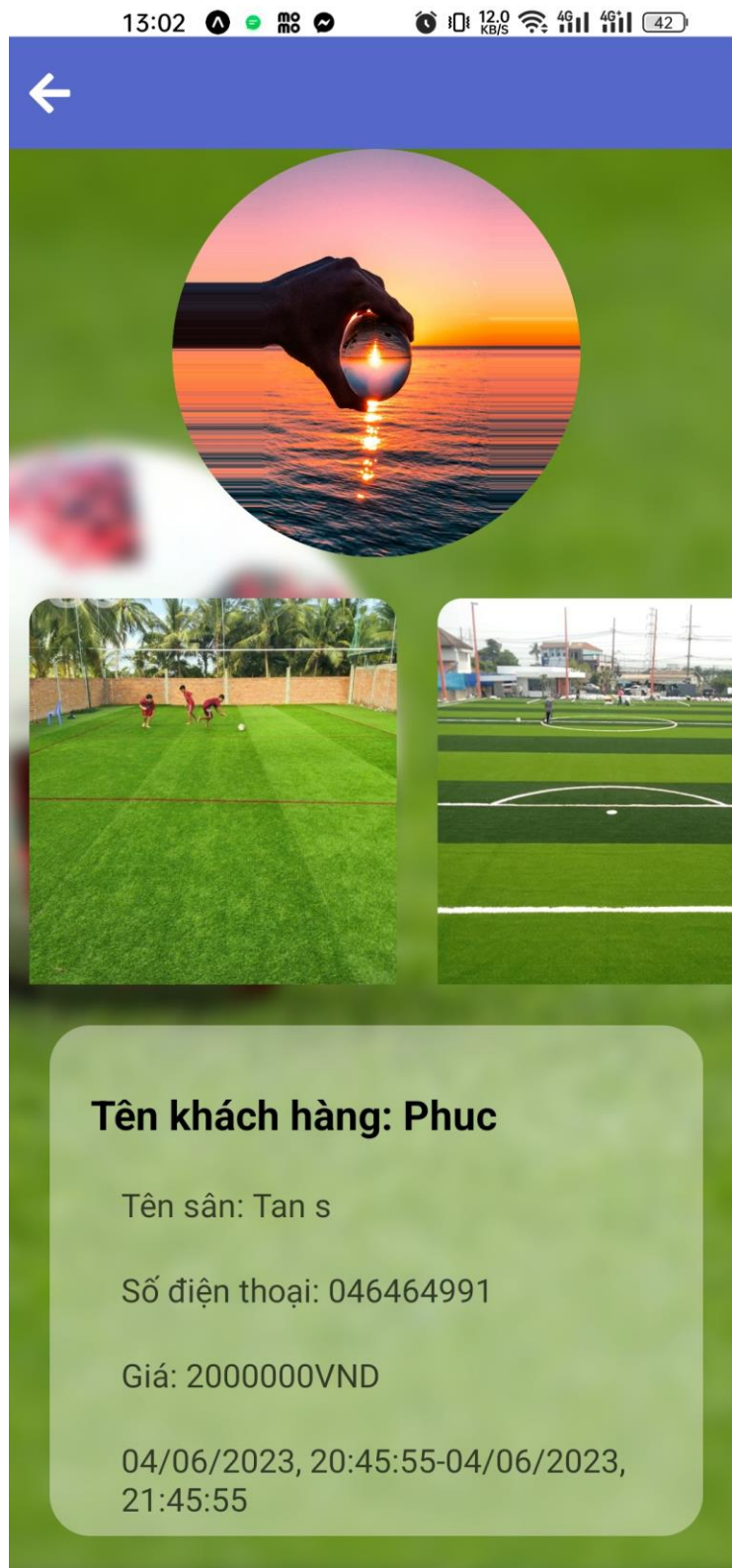


Settings

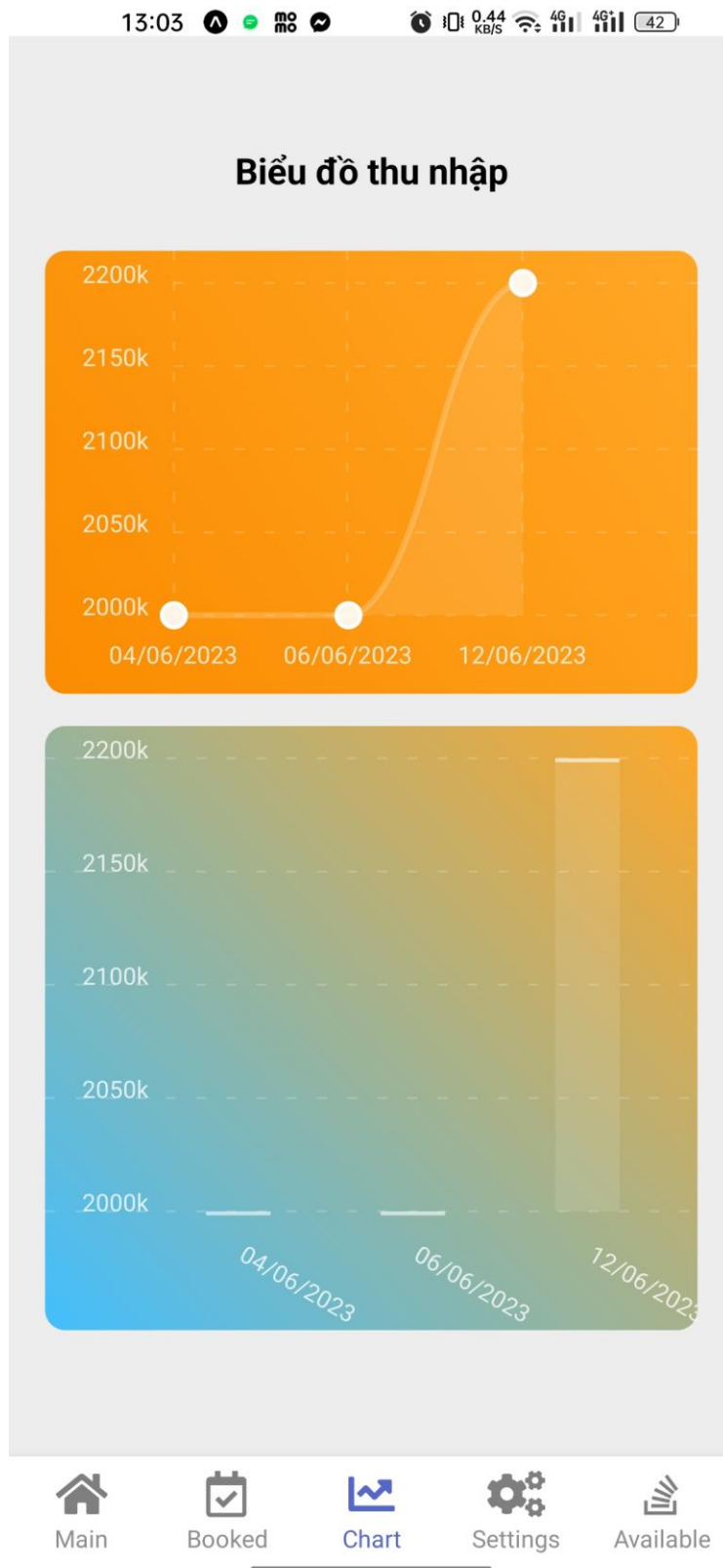


Available

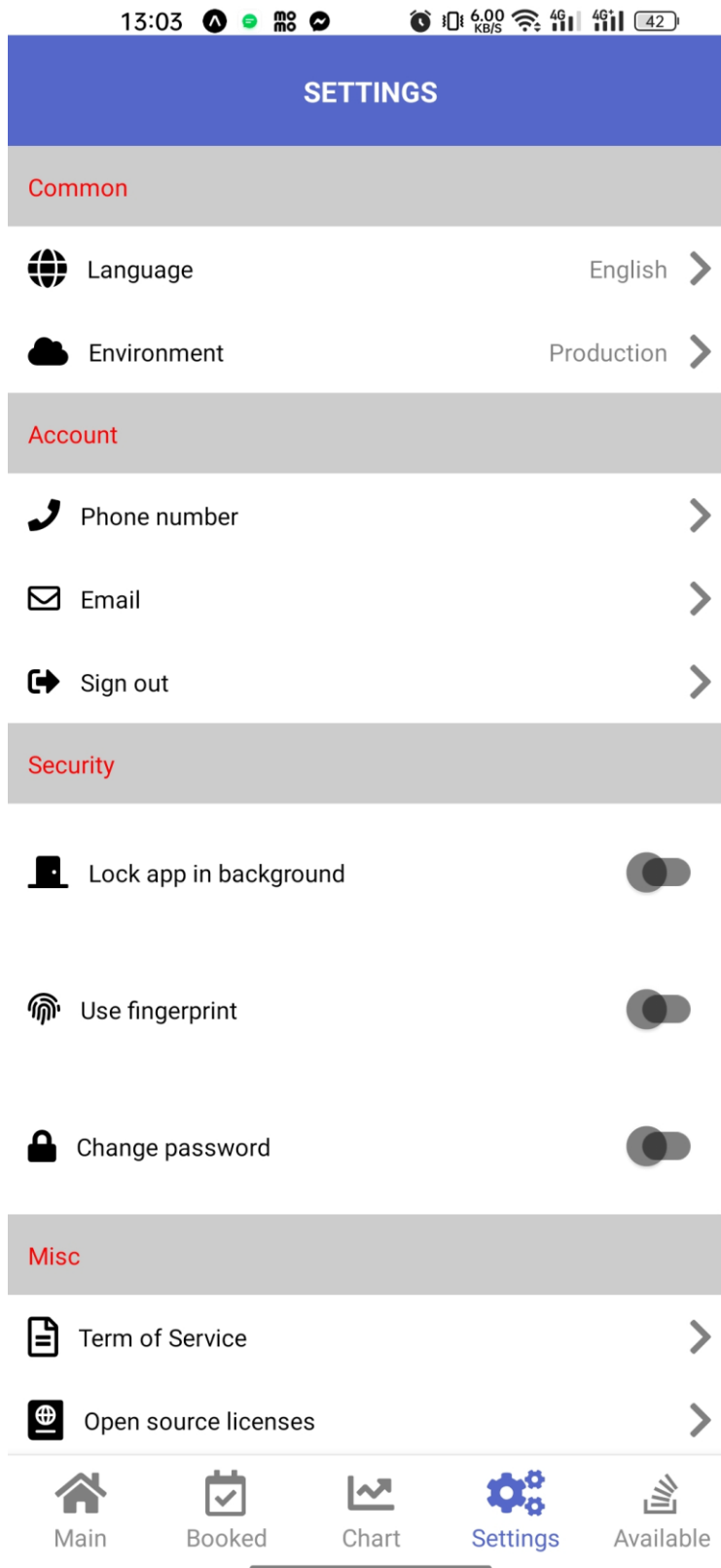
Picture 11: Booked screen



Picture 12: Detail booked



Picture 13: Chart screen



Picture 14: Setting screen



Picture 15: Available field

Chapter 9. Installation

Language and Environment:

The application is developed using React Native, a popular framework for building cross-platform mobile applications. Therefore, the installation process requires the following:

- Programming Language: JavaScript
- Development Framework: React Native
- Required Development Tools: Node.js and npm (Node Package Manager)

Installation Notes:

- Before proceeding with the installation, please consider the following notes:
1. Operating System Compatibility: Ensure that your development environment supports the operating systems for which the application is intended (e.g., iOS, Android). React Native provides support for multiple platforms, but there may be specific requirements or considerations for each.
 2. Node.js and npm: Make sure you have the latest version of Node.js and npm installed on your machine. You can visit the official Node.js website (<https://nodejs.org>) to download and install the latest stable version.
 3. Development Environment Setup: Set up your development environment by installing the necessary tools and dependencies. This may include a code editor (e.g., Visual Studio Code) and a mobile emulator or device for testing the application.
 4. Dependency Management: React Native relies on various third-party libraries and packages. It is essential to manage and install these dependencies correctly to ensure the application runs smoothly. The package.json file included with the application source code should contain a list of required dependencies. Run the following command in the project directory to install these dependencies:

```
npm install
```

5. This command will fetch and install the required packages listed in the package.json file.
6. Building and Running the Application: Once the dependencies are installed, you can build and run the application using the following commands:

```
npx react-native run-ios
```

or

```
npx react-native run-android
```

7. These commands will launch the application on the respective platforms (iOS or Android) using a connected emulator or device.
8. Note: Make sure to have the appropriate emulator or device set up and connected to your development environment.
9. Troubleshooting: If you encounter any issues during the installation process, please refer to the official React Native documentation (<https://reactnative.dev/docs/environment-setup>) or community resources for troubleshooting steps and solutions.

Chapter 10. Testing

All the testing process was conducted manually.

10.1 Testing for the Login/Logout Application Function:

Test Approach: Unit Testing

Test Cases:

Test Case 1: Verify that users can successfully login to the system with valid credentials.

- Description: This test case verifies that users can log in to the soccer field booking app using valid credentials.
- Steps:
 - + Enter valid login credentials (email/username and password).
 - + Click on the login button.
- Expected Result: The user should be logged in successfully and granted access to the app's features.

Test Case 2: Verify that users are properly logged out from the system when they choose to logout.

- Description: This test case verifies that users can log out from the soccer field booking app and end their current session.
- Steps:
 - + Locate the logout button within the app.
 - + Click on the logout button.
- Expected Result: The user should be logged out and redirected to the login page.

10.2 Create/Edit/Delete Field feature:

Test Approach: Unit Testing

- Test Case: Verify that field owners can create a new field with valid information.
- Description: This test case ensures that field owners can successfully create a new field by providing valid information.
- Steps:

- + Log in to the app as a field owner.
- + Navigate to the "Create Field" section.
- + Enter all the required information for the new field (e.g., field name, type, price, operating hours).
- + Click on the "Create" or "Save" button.
- Expected Result: The new field should be created and added to the Expo SQLite database. The field owner should be able to view and manage the newly created field.

10.3 View Detail Revenue functionality:

Test Approach: Unit Testing

- Test Case: Verify that field managers can view revenue data.
- Description: This test case ensures that field managers can access the "View Detail Revenue" feature,
- Steps:
 - + Log in to the app as a field manager.
 - + Navigate to the "View Detail Revenue" section.
 - + Analyze the revenue data and look for trends, patterns, or any other relevant insights.
- Expected Result: The field manager should be able to access the "View Detail Revenue" section and view the revenue data for the selected time period.

Chapter 11. Conclusion

11.1 Final result

In conclusion, the development of a mobile application for managing football pitches using React Native is a significant step towards revolutionizing sports management. The application provides efficient organization and administration of football pitches, catering to the growing demand for digital solutions in the sports landscape. The use of React Native as the framework for development offers numerous benefits, including cross-platform compatibility, code reusability, native-like performance, and a vibrant community of developers.

The implemented features of the application, such as account management, field booking and cancellation, and revenue statistics, address the key requirements for efficient pitch management. The integration of Firebase Authentication ensures secure and convenient user account management, while the Expo SQLite database facilitates the creation, editing, and deletion of fields. The application also provides field availability information and detailed revenue insights, empowering both users and field managers with valuable data.

11.2 Advantages:

Efficient management: The mobile application streamlines the management of football pitches, providing a convenient and accessible platform for users and field owners.

Enhanced user experience: Users can easily book and cancel fields, view availability, and access detailed revenue information, improving their overall experience.

Cross-platform compatibility: The use of React Native enables the application to run on both iOS and Android platforms, reaching a wider user base.

Code reusability: With React Native, the development process is more efficient as the codebase can be reused for multiple platforms, reducing development time and effort.

Native-like performance: The application delivers a seamless user experience with fast loading times, smooth animations, and optimal performance.

11.3 Disadvantages:

Learning curve: Developers may require some time to learn React Native and its ecosystem, especially if they are not familiar with JavaScript or native mobile app development.

Dependency on third-party libraries: While React Native has a vast library ecosystem, relying heavily on external libraries can introduce potential compatibility issues and maintenance challenges.

Limited access to native features: Although React Native provides access to many native device features, there may be certain advanced functionalities that require native development or custom bridging.

11.4 Future Developments:

There are several potential future developments for the mobile application for managing football pitches:

Advanced booking features: Implementing additional functionalities such as recurring bookings, booking reminders, and in-app payments can enhance the convenience for users and improve the overall booking process.

Integration with external APIs: Integrating the application with external APIs, such as weather forecasts or GPS services, can provide additional information to users and optimize their experience.

Real-time notifications: Implementing real-time notifications for field availability, booking confirmations, and updates can keep users and field owners informed and engaged.

Enhanced analytics: Expanding the statistical features to provide more comprehensive analytics, customer behavior insights, and performance metrics can help field owners make data-driven decisions and improve their business strategies.

Social features: Incorporating social features such as user reviews, ratings, and social media sharing can foster community engagement and generate more visibility for the app and its associated fields.

Overall, the mobile application for managing football pitches using React Native offers a promising solution for efficient sports management. With further development and enhancements, it has the potential to transform the way football pitches are organized, booked, and managed, providing a seamless experience for both users and facility owners.

Bibliography

Google. (n.d.). Retrieved from <https://firebase.google.com/docs>

Platforms, M. (n.d.). *React Native*. Retrieved from <https://reactnative.dev/docs/getting-started>

W3Schools. (n.d.). Retrieved from https://www.w3schools.com/react/react_getstarted.asp