

TURFEY - TURF BOOKING SYSTEM



A DESIGN PROJECT REPORT

Submitted by

M.AATHIRAJU

S.GOWRI SHANKAR

R.KIRUSHIGAN

M.RAJADURAI

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM-621112

JUNE-2024

BONAFIDE CERTIFICATE

Certified that this design project report titled “**TURFEY- A BOOKING SYSTEM FOR TURFS AND MULTI SPORTS ARENA**” is the bonafide work of **M. AATHIRAJU (REG NO: 811721243001) GOWRI SHANKAR.S (REG NO: 811721243017) KIRUSHIGAN.R (REG NO: 811721243025) RAJADURAI.M (REG NO: 811721243303)** who carried out the project under my supervision.

SIGNATURE

Dr.T.Avudaiappan M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Associate Professor

Department of Artificial Intelligence

K.Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

SIGNATURE

Mrs.S.Geetha M.E.,

SUPERVISOR

Assistant Professor

Department of Artificial Intelligence

K.Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

Submitted for the viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We jointly declare that the project report on “**TURFEY- A BOOKING SYSTEM FOR TURFS AND ENTERTAINMENT**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This design project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

SIGNATURE

M.AATHIRAJU

S.GOWRI SHANKAR

R.KIRUSHIGAN

M.RAJADURAI

PLACE : SAMAYAPURAM

DATE :

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in - debt to our institution “**K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**”, for providing us with the opportunity to do this project.

We are glad to credit honourable Chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr.S. KUPPUSAMY, MBA., Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

We would like to thank our Principal **Dr.N. VASUDEVAN, M.E., Ph.D.**, who gave opportunity to frame the project the full satisfaction.

We whole heartily thanks to **Dr.T. AVUDAIAPPAN, M.E., Ph.D.**, HEAD OF THE DEPARTMENT, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep and sincere gratitude to my project guide **Mrs.S. GEETHA., M.E** ASSISTANT PROFESSOR, **ARTIFICIAL INTELLIGENCE** for her incalculable suggestions, creativity, assistance and patience which motivated me to carry out the project successfully.

I render my sincere thanks to my project coordinator **Mrs.G.NALINA KEERTHANA M.E.**, other faculties and non-teaching staff members for providing valuable information during the course. I wish to express my special thanks to the officials & Lab Technicians of our departments who rendered their help during the period of the workprogress.

ABSTRACT

The Turf Booking System is a web application designed to streamline the process of booking sports turf facilities. As urbanization progresses, the demand for recreational spaces, particularly for sports activities, has increased significantly. However, managing and booking these facilities efficiently has remained a challenge. TBS addresses this challenge by providing a user-friendly platform where users can easily search for available turf facilities, make reservations, and manage bookings seamlessly. Key features of the TBS include a comprehensive database of turf facilities, allowing users to search based on location, availability, and amenities. The system also incorporates a user authentication system to ensure secure access for both facility administrators and users. Administrators can manage facility details, set availability, and monitor bookings, while users can create accounts, book available slots, and make payments online. Users can easily view available time slots, select preferred dates, and make reservations with just a few clicks. Additionally, the system provides notifications and reminders to users regarding upcoming bookings, ensuring a smooth experience. Furthermore, TBS offers flexibility and scalability to accommodate various types of turf facilities, including soccer fields, cricket, and tennis courts. It can be customized to meet the specific needs of different sports venues, such as leagues, tournaments, and recurring bookings. In summary, the Turf Booking System is a comprehensive solution designed to modernize and optimize the management of sports turf facilities.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	9
	1.1 BACKGROUND	9
	1.2 PROBLEM STATEMENT	10
	1.3 OBJECTIVE	10
2	LITERATURE SURVEY	12
	2.1 CONFIGURING AN APPLICATION WHICH ALLOWS ONLINE BOOKING AND PURCHASE OF TRAVEL TICKETS FOR RAILWAYAND ROAD TRANSPORT - UNIFIED MODELING LANGUAGE	12
	2.2 ENHANCED HOTEL BOOKING APPLICATION USING PEGA	13
	2.3 THE DESIGN AND APPLICATION OF THE PUBLICBOOKING SERVICE SYSTEM	14
	2.4 BOOKAZOR - AN ONLINE APPOINTMENT BOOKING SYSTEM	15
3	SYSTEM ANALYSIS	16
	3.1 EXISTING SYSTEM	16
	3.1.1 Algorithms Used	16
	3.1.2 Drawbacks	17
	3.2 PROPOSED SYSTEM	17
	3.2.1 Algorithms Used	17
	3.2.1 Advantages	18
4	SYSTEM SPECIFICATION	19
	4.1 HARDWARE SPECIFICATIONS	19
	4.2 SOFTWARE SPECIFICATIONS	19
	4.3 SOFTWARE DESCRIPTION	19

CHAPTER NO.	TITLE	PAGE NO.
5	ARCHITECTURAL DESIGN	20
	5.1 ARCHITECTURAL DIAGRAM	20
	5.2 DATA FLOW DIAGRAM	21
	5.3 USER CASE DIAGRAM	22
	5.4 ACTIVITY DIAGRAM	23
	5.5 SEQUENCE DIAGRAM	24
6	MODULE DESCRIPTION	25
	6.1 MODULES	25
	6.1.1 Login/Signup Module	25
	6.1.2 Location Module	25
	6.1.3 Recommendation Module	26
	6.1.4 Booking Module	26
	6.1.5 Payment Module	27
	6.1.6 Confirmation Module	27
7	CONCLUSION AND FUTURE ENHANCEMENT	28
	7.1 CONCLUSION	28
	7.2 FUTURE ENHANCEMENT	28
	APPENDIX 1 SOURCE CODE	30
	APPENDIX 2 SCREENSHOTS	41
	REFERENCES	44

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO
5.1	Architectural Diagram	20
5.2	Data Flow Diagram	21
5.3	Use Case Diagram	22
5.4	Activity Diagram	23
5.5	Sequence Diagram	24
A.2.1	Login Page	41
A.2.2	Recommended Turfs	41
A.2.3	Selecting Turfs	42
A.2.4	Booking	42
A.2.5	Payment	43
A.2.6	Slot Confirmation	43

CHAPTER 1

INTRODUCTION

The Turf Booking System is a cutting-edge web application revolutionizing the way sports turf facilities are managed and booked. Designed to meet the growing demand for recreational spaces, TBS offers users a convenient platform to search for available turf facilities, make reservations, and manage bookings effortlessly. With its intuitive interface and robust features, TBS aims to streamline the booking process, enhance accessibility, and promote active lifestyles within communities.

1.1 BACKGROUND

The need for a Turf Booking System arises from the growing demand for sports facilities amidst urban expansion. As cities become more densely populated, the availability of recreational spaces, particularly for sports activities, becomes increasingly scarce. Traditional methods of managing turf facilities, such as phone bookings or in-person inquiries, are inefficient and often lead to scheduling conflicts and underutilization of resources. Moreover, the lack of centralized booking systems results in frustration for both facility administrators and users. Thus, there is a pressing need for a digital solution that streamlines the booking process, enhances accessibility, and optimizes the utilization of turf facilities, ultimately fostering a more active and engaged community. The demand for turf facilities extends beyond traditional sports leagues to encompass a wide range of activities, including casual matches, training sessions, and community events. However, without a centralized booking system, managing these diverse activities becomes challenging, leading to inefficiencies and missed opportunities for revenue generation. Additionally, the lack of transparency in scheduling often leads to disputes and dissatisfaction among users. A Turf Booking System addresses these issues by providing a user-friendly platform where users can easily browse available slots, make reservations,

and receive instant confirmations. By digitizing the booking process, the system enables facility administrators to efficiently manage resources, optimize utilization rates, and generate insights into usage patterns, thus facilitating informed decision-making and long-term planning.

1.2 PROBLEM STATEMENT

- Current turf booking systems often rely on manual processes, phone calls, or in-person visits, leading to inefficiencies and delays in securing reservations
- Some turf facilities lack an online booking system, limiting accessibility for users who prefer to make reservations digitally or outside of regular business hours
- Manual record-keeping processes may result in inaccuracies, making it difficult for administrators to track usage patterns, analyze trends, and make informed decisions about facility management.
- Manual booking systems are prone to errors and double bookings, leading to scheduling conflicts and dissatisfaction among users.

1.3 OBJECTIVES

Provide users with easy access to information about available turf facilities in their desired locations. Simplify the process of booking turf facilities through an intuitive and user-friendly web application interface. Enable facility administrators to efficiently manage bookings, set availability, and monitor facility usage. Ensure secure online transactions for users making bookings and payments through the system.

- Streamline the process of managing turf facilities by providing administrators with tool to oversee bookings, availability, and facility details.
- Offer a user-friendly interface that allows users to easily search for available turffacilities, select preferred dates and times, and make reservations hassle-free.
- Increase access to sports turf facilities by providing a centralized platform where users

can find and book available slots conveniently.

- Maximize the utilization of turf facilities by minimizing idle time through efficient booking management and scheduling.
- Ensure secure online transactions for booking payments, providing users with confidence in the system's reliability and safety.
- Provide flexibility to accommodate various types of turf facilities and customizable options to meet specific needs, such as league bookings, recurring reservations, and special events.

CHAPTER 2

LITERATURE SURVEY

2.1 CONFIGURING AN APPLICATION WHICH ALLOWS ONLINE BOOKING AND PURCHASE OF TRAVEL TICKETS FOR RAILWAY AND ROAD TRANSPORT - UNIFIED MODELING LANGUAGE

Author: Desdemona Isabela, Scărișoreanu

Year Of Publication: IEEE APR 2020

Algorithm Used: Recommendation Systems, Predictive Analytics, Fraud Detection, Natural Language Processing (NLP)

Abstract : The purpose of this paper is to configure an application which will allow the online reservation and purchase of travel tickets for rail transport and road system. This application aims to connect the reservation and payment systems of the Romanian rail transport operators and those that ensure the transport of passengers by coaches, on the internal routes. The purpose of the application is to create a software product which will help as many users as possible. They will thus be able to plan their trip, including purchasing online a single ticket to travel by different means of transport, at an early stage train and bus, using the mobile phone. The online system for booking and purchasing tickets responds to the current needs of the society, one of them being the need to travel and encourages the use of public transport which leads to the reduction of pollution.

Merits: Efficiency, security, personalization, fraud prevention, optimization.

Demerits: Complexity may lead to system instability.

2.2 ENHANCED HOTEL BOOKING APPLICATION USING PEGA

Author: R Dhanagopal, N Archana, R Menaka

Year Of Publication: IEEE JUL 2020

Algorithm Used: Collaborative Filtering, Content-Based Filtering, Random Forest.

Abstract: Flying colors is a Hotel Booking Application; this application provides different staying solutions to customers. It is to book customers a hotel room online based on date and location along with the type of hotel. This architecture is based on agile development which is specially designed to work on multiple regions, channels, and quick future enhancements. PEGA7 Business Process Management (BPM) Tool which enables significant enhancement in building the application and data management experiences mobile functionality, user experience, and analytics. The goal of developing this application using PEGA is to improve the efficiency and performance, the liveliness of usual processes in the business. An automated system is developed to deliver high security and end to end delivery of works and speeds up the process drastically.

Merits: Improved user experience, personalized recommendations

Demerits: Complexity, training data availability.

2.3 THE DESIGN AND APPLICATION OF THE PUBLIC BOOKING SERVICE SYSTEM

Author: Liu Dan, Xi'an Xi'an, China Yue.

Year Of Publication: IEEE NOV-2020

Algorithm Used: Collaborative Filtering, Content-Based Filtering, DecisionTrees, Neural Networks

Abstract: The target of the public booking service system is to relieve the pressure of queue congestion due to the contradictions of supply and demand between people and social resources, achieve a well-ordered management of social resources, and save time spent waiting in line. First, through the analysis of booking resources, this paper proposes a unified booking resources model and achieves the consistency management of all kinds of booking resources. On this basis, this article introduces the basic structure and components of Public Service Booking System; terminal queuing calling system, network application server and Web site. In the end, the paper analyzes the application in Real Estate Trade Center in Sichuan province; the application proves this system is very effective.

Merits: Efficiency, accessibility, convenience, transparency, user-friendly.

Demerits: Complexity, scalability issues, potential errors.

2.4 BOOKAZOR - AN ONLINE APPOINTMENT BOOKING SYSTEM

Author: V. Akshay, Anish Kumar S, R.M. Alagappan, Gnanavel

Year Of Publication: IEE FEB 2022

Algorithm Used: CNN , Safy Model , Uav Hyperspectral Imagery

Abstract: Bookazor is an appointment booking and scheduling web-based application which is used for booking appointments in the streams of parlor, hospitals and architects within a defined geographic area. This application is streamlined in an ionic basis. It is an open source SDK for hybrid mobile application development. It uses technologies like CSS, HTML, and JavaScript. Firebase plays a vital role in fetching data for appointment scheduling that helps to enhance application development effectively. The system includes NodeJS for storing the number of requests, each of which specifies a sequence of regions to be visited by a particular user. NodeJS server is used for offering appointments at specified times, using the table to check for availability of operatives in specified regions at specified times, and for inserting new jobs in the routes to reflect booked appointments. A scheduler periodically updates the routes e.g. by means of a simulated annealing process, to generate a new set of appointments and prevents duplication.

Merits: Convenience for scheduling appointments, Efficient management of bookings, Accessible from anywhere.

Demerits: Scalability may be challenging, Security vulnerabilities may arise, experience might suffer.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

It appears there might be a misunderstanding. "Book My Show" is a popular online ticketing platform primarily used for booking tickets to movies, events, concerts, and other entertainment shows. It doesn't directly relate to turf booking systems. However, if you're looking for information about an existing turf booking system similar to Book My Show but for sports fields, golf courses, or other turf-related facilities.

3.1.1 Algorithm Used

- **First-Come, First-Served (FCFS)**

This is one of the simplest algorithms where bookings are processed in the order they are received. It's straightforward and ensures fairness, but it may not optimize resource utilization.

- **Priority Queues**

In priority queue-based algorithms, bookings are assigned priority levels based on certain criteria such as membership status, booking history, or special requests. This ensures that high-priority bookings are processed first.

- **Time Slot Allocation**

This algorithm divides the available time slots into fixed intervals (e.g., hourly or half-hourly) and allocates bookings based on these time slots. It ensures efficient utilization of time but may lead to underutilization of resources if there are gaps between bookings.

- **Optimization Algorithms**

Various optimization techniques such as greedy algorithms, genetic algorithms, or dynamic programming can be used to optimize resource allocation and scheduling. These algorithms aim to maximize resource utilization, minimize waiting times, or satisfy certain constraints.

3.1.2 Drawbacks

- Limited availability
- Overbooking risk
- 3. Technical glitches
- Complexity for administrators
- User dependency on technology

3.2 PROPOSED SYSTEM

The introduction of a new app, finding and book the perfect venue is just a few clicks away. The app is designed with convenience in mind. It features a user-friendly interface and an intuitive search function to help users quickly find the perfect venue. The app also provides detailed information about each venue including photos, maps, and reviews. Additionally, users can easily book and pay for their reservation in the app.

3.2.1 Algorithms Used

- **Booking Algorithm**

An algorithm to manage booking slots efficiently, considering factors like

availability, user preferences, and facility constraints.

- **Search Algorithm**

A search algorithm to help users find available sports arenas based on location, availability, facilities, and other criteria.

- **Scheduling Algorithm**

This algorithm helps in optimizing the scheduling of bookings to maximize facility usage while minimizing conflicts and ensuring fair allocation of time slots.

- **Payment Processing Algorithm**

Secure algorithms for handling payment transactions, including encryption, decryption, and verification processes.

- **User Recommendation Algorithm**

If you want to provide personalized recommendations for users, you might need algorithms to analyze user preferences, booking history, and other relevant data to suggest suitable sports arenas or time slots.

3.2.2 Advantages

- Efficient resource utilization
- Convenient booking process
- Flexible scheduling options
- Improved user experience.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE SPECIFICATIONS

- Computer
 - Processor - at least 2 GHz clock speed
 - RAM - Minimum 8 GB RAM
 - Storage - SSD (Solid State Drive) is preferable over HDD (Hard Disk Drive)
 - Devices – iOS or Android.

4.2 SOFTWARE SPECIFICATION

- Operating System – Windows or macOS or Linux
 - Integrated Development Environment (IDE) - Android Studio, Xcode, VisualStudio/Visual Studio Code
 - Version Control System - Visual Studio/Visual Studio Code
- Database Management System (DBMS) - MySQL, PostgreSQL, SQLite
- Testing Tools
 - Android - JUnit, Espresso
 - iOS - XCTest, XCUITest
 - Deployment Tools - Google Play Console, Apple App Store.

4.3 SOFTWARE DESCRIPTION

Android Studio is a software tool created by Google for building apps for Android devices. It's like a toolbox for developers, providing everything they need to write, test, and debug their apps. It includes features like a code editor, layout designer, and testing tools to make the app development process easier.

CHAPTER 5

ARCHITECTURAL DESIGN

The architecture of a turf booking system encompasses several components working together seamlessly to facilitate efficient booking processes and user interactions. At its core lies the client application, typically a web or mobile interface through which users access the system to browse available turf slots, make bookings, and manage reservations.

This client application communicates with the web server, which hosts and serves the application's content, handling client requests and interfacing with the application server. The application server, where the business logic of the system resides, processes these requests, managing authentication, validating inputs, and interacting with the database to retrieve or store data. Speaking of which, the database serves as the repository for all pertinent data related to turf facilities, available time slots, user accounts, bookings, and transaction records, ensuring data integrity and persistence.

5.1 ARCHITECTURAL DIAGRAM

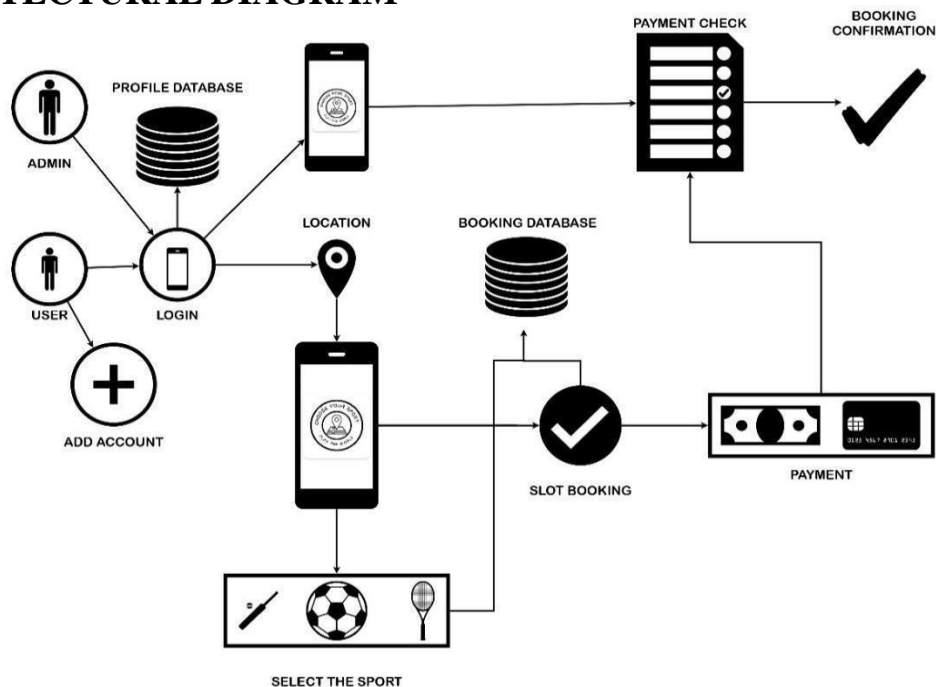


Figure No. 5.1. Architectural Diagram

5.2 DATA FLOW DIAGRAM

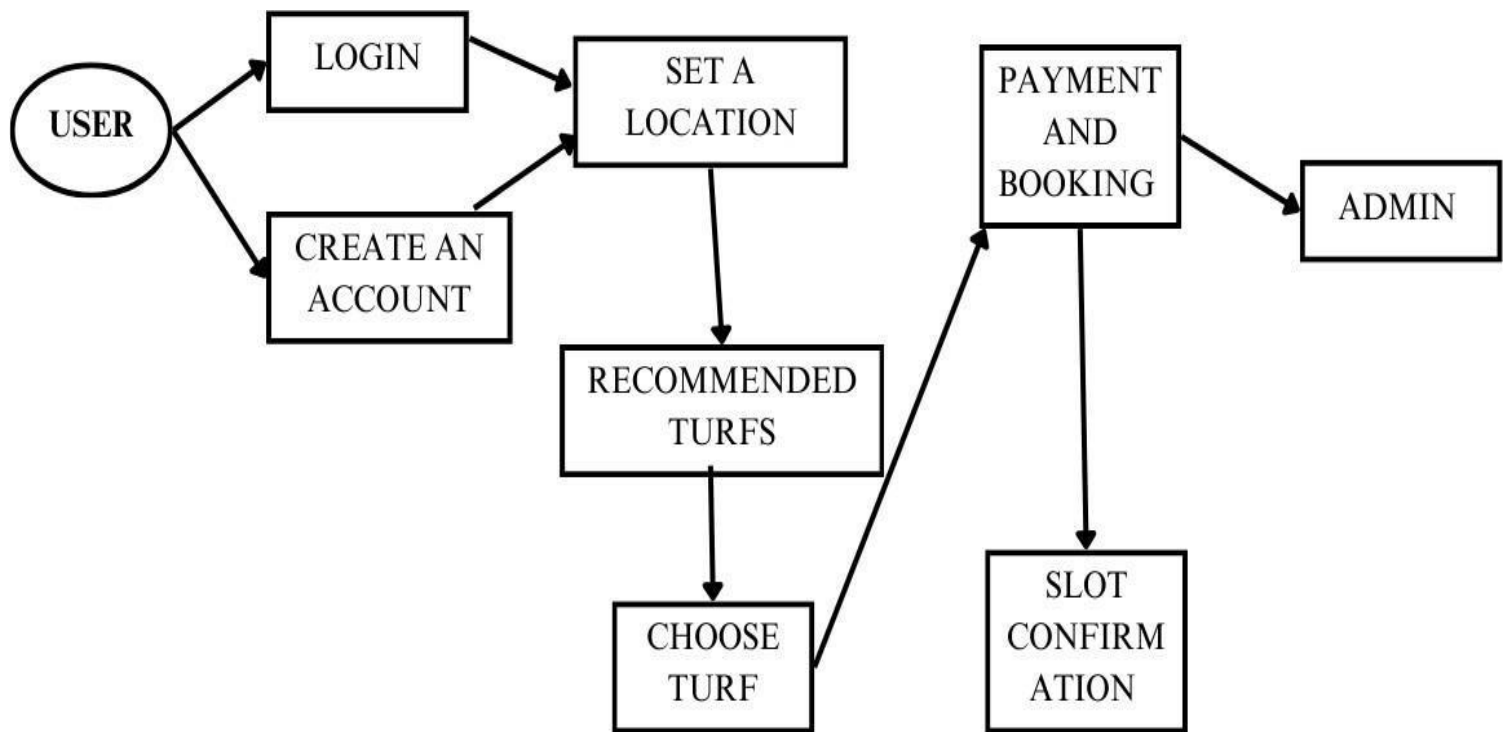


Figure No.5.2. Data Flow Diagram

5.3 USECASE DIAGRAM

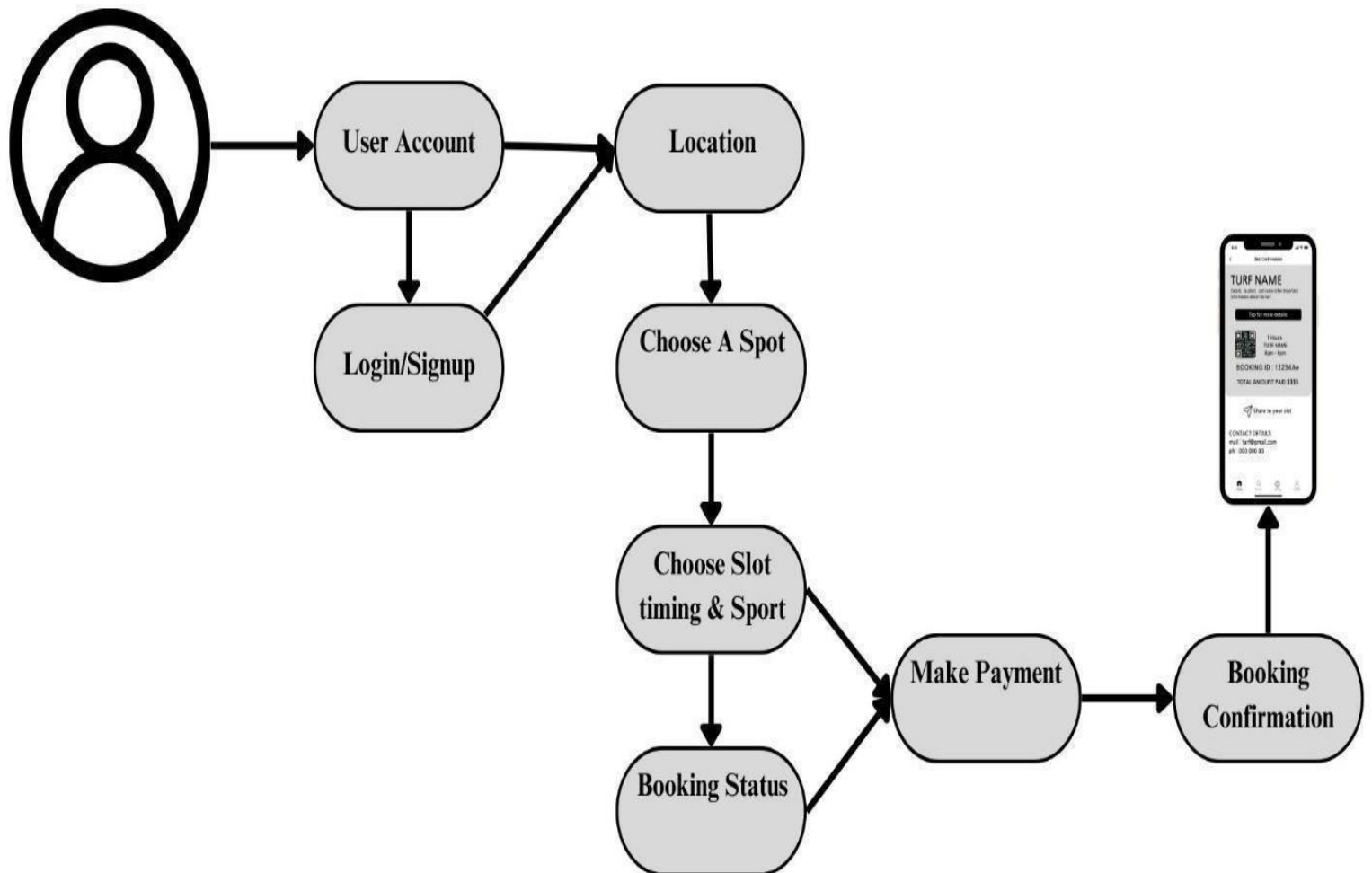


Figure No.5.3. Usecase Diagram

5.4 ACTIVITY DIAGRAM

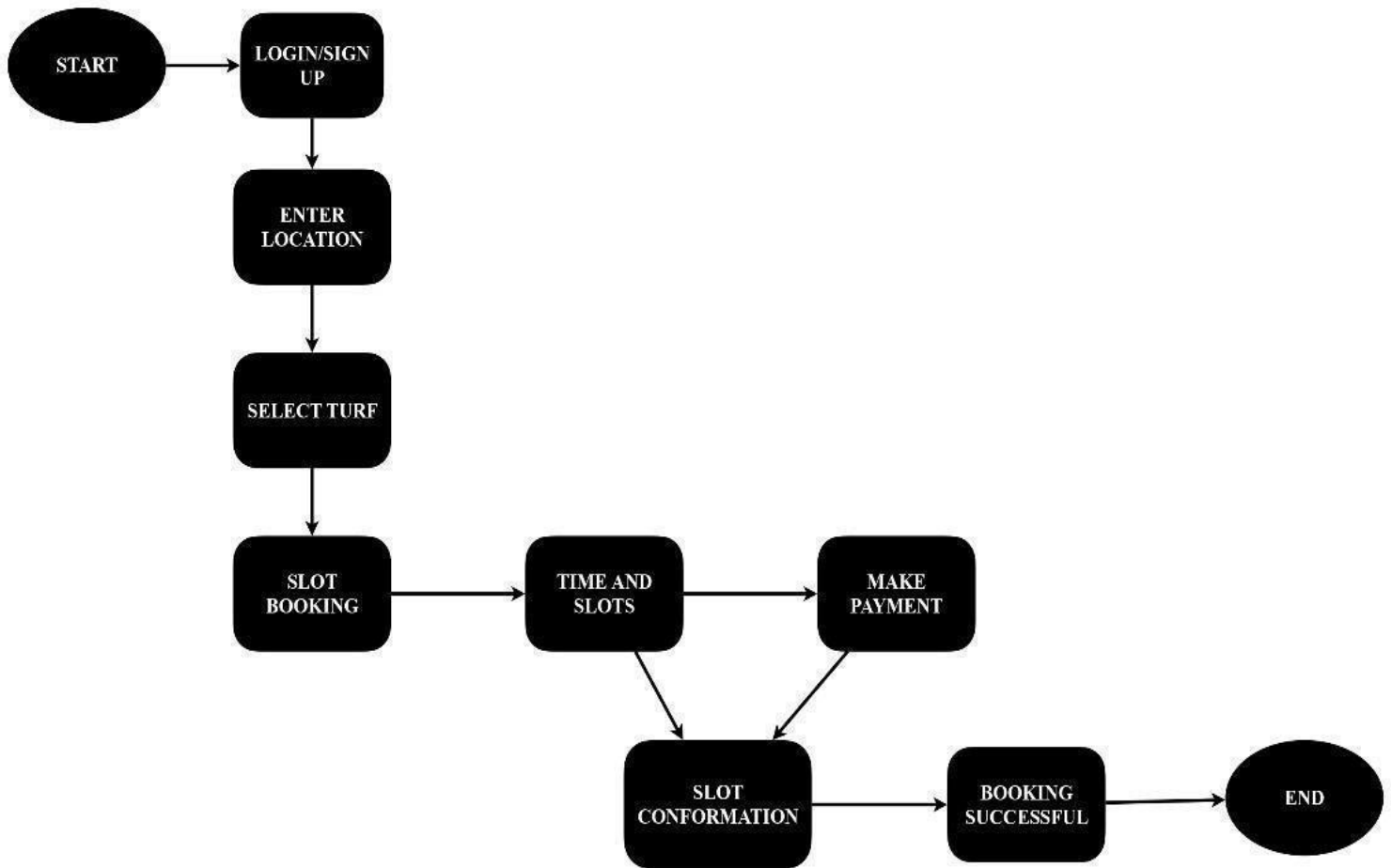


Figure No.5.4. Activity Diagram

5.5 SEQUENTIAL DIAGRAM

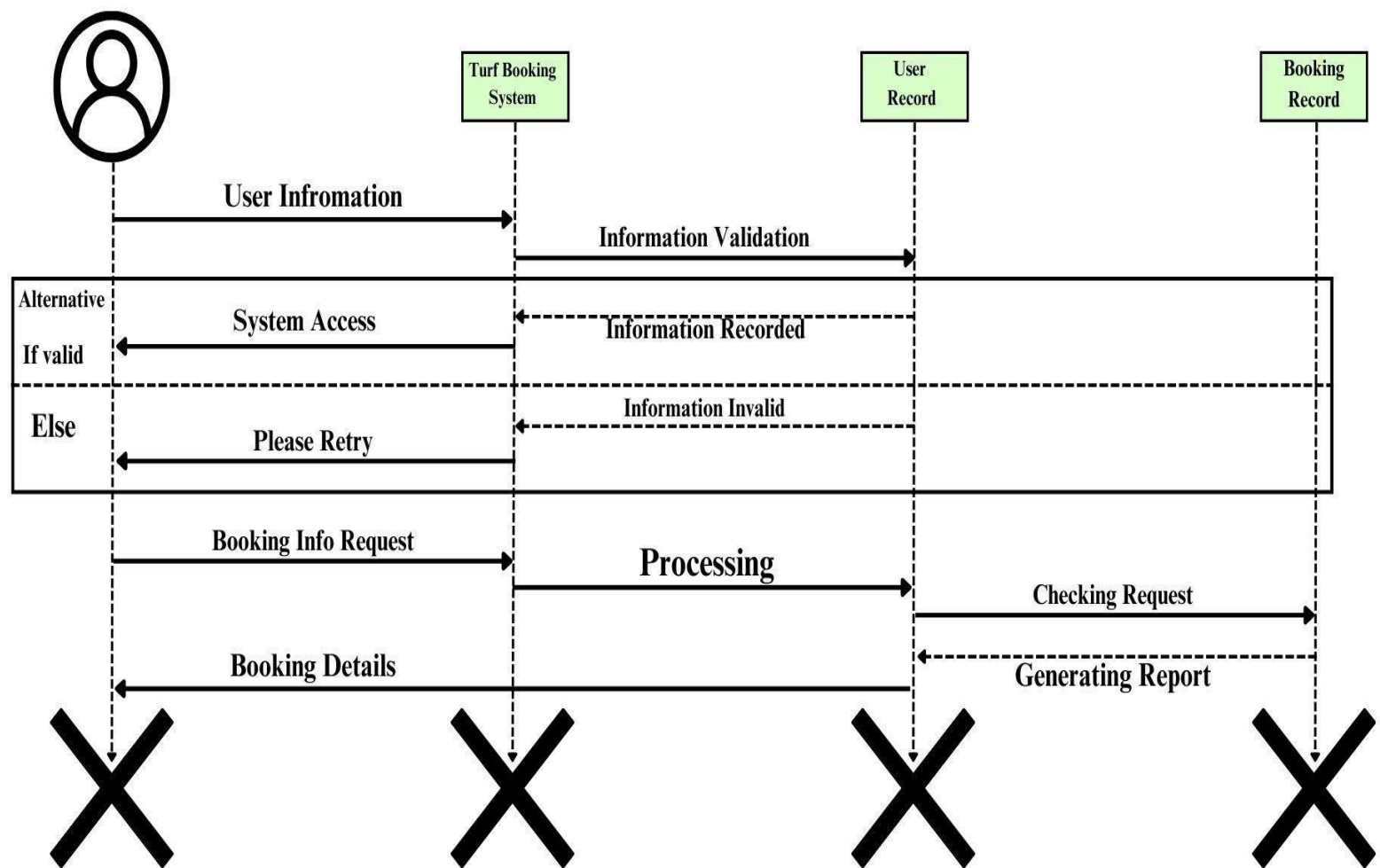


Figure No.5.5. Sequential Diagram

CHAPTER 6

MODULE DESCRIPTION

6.1 MODULES

- Login / Sign-Up Module
- Location Module:
- Recommendation Module
- Booking Module
- Payment Module
- Confirmation Module

6.1.1 Login / Sign-Up Module

The login/sign-up module is the phase where the end users sign-up and log on to Turfey(app)to explore the features and book turf.

- **Login Process**

The login phase is the process where the end users are allowed to enter their credentials to enter the app.

- **Sign-up Process**

The sign-up phase is the process when the end user does not have anaccount to login se he/she is forced to sign-up to the app first.

6.1.2 Location Module

The location module is the phase where the end users enter their location via GPS or manually for smooth search of near-by turfs.

- **GPS**

GPS stands for Global Positioning System. It is a radio navigation system used onland, sea, and air to determine the exact location, time so that the users can enter their location.

- **Manual Location**

Once the user entered the location module he/she should enter their location manually by searching it or by entering the pincode.

6.1.3 Recommendation Module

This app recommend a user-friendly online platform that offers easy navigation and efficient booking processes. Incorporating features such as secure payment options and automated confirmation emails will enhance user experience and reliability. Additionally, providing customizable booking options for recurring bookings or special events can cater to various user needs. With a responsive interface and robust backend management, this turf booking system can efficiently serve both facility managers and customers, ensuring smooth operations and maximum satisfaction.

- **Database Design:** Develop a database schema to store information about turf availability, bookings, user profiles, and administrative data.
- **Booking Management:** Implement features for users to search for available turf slots, select desired dates and times, and proceed with booking. Include options for recurring bookings, multiple turf selections, and flexible booking durations.
- **Requirement Analysis:** Understand the specific needs and requirements of the turf facility, including booking policies, user preferences, and administrative features.

6.1.4 Booking Module

Booking turf with the user-friendly system is as easy as a breeze. Simply navigate to our platform, select your desired date and time, and with just a few clicks, secure your spot on the turf. Our streamlined booking process ensures efficiency and convenience, allowing you to focus on what truly matters – enjoying your game. Whether it's a friendly match or a rigorous training session, booking turf has never been smoother. Join us and experience hassle-free turf booking today.

6.1.5 Payment Module

This project offers a seamless payment process, ensuring convenience and efficiency for Turfey users. With secure payment gateways integrated directly into the platform, users can effortlessly complete transactions for their turf reservations. Whether it's for a one-time booking or a recurring schedule, our system accepts various payment methods, providing flexibility to our customers. Say goodbye to hassle and delays – with the streamlined payment system, booking your turf has never been easier.

- **Booking Confirmation and Payment Authorization:** Implement a system to confirm bookings and authorize payments, ensuring that payment is processed only when a booking is confirmed.
- **Refund Processing:** Develop a mechanism to handle refunds about 80% in case of cancellations or disputes, ensuring that refunds are processed promptly and accurately.
- **Security Measures:** Implement robust security measures to protect users' payment information, including encryption, PCI compliance, and fraud detection mechanisms.
- **Payment Verification:** Implement verification processes to ensure the authenticity of payment transactions and prevent fraudulent activities.

6.1.6 Confirmation Module

Confirmation for turf booking system ensures that your reservation is secured and acknowledged. Once you've made your booking, you'll receive a confirmation email or message containing all the pertinent details of your reservation, such as the date, time, and location of your turf booking. This serves as a reassurance that your desired slot has been successfully reserved, giving you peace of mind and eliminating any uncertainties regarding your upcoming sports activity. With confirmation in place, you can confidently plan and prepare for your time on the turf, knowing that your booking is confirmed.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In conclusion, the implementation of a turf booking system presents a transformative solution to streamline and optimize the process of reserving sports facilities. Through this project, we have successfully addressed the challenges of manual booking systems by introducing an efficient and user-friendly digital platform. By leveraging advanced technologies, such as online reservation portals and automated scheduling algorithms, we have enhanced accessibility and convenience for both users and facility managers alike. The turf booking system not only simplifies the booking process but also improves resource utilization and revenue generation for sports venues. With its intuitive interface and robust functionality, this system not only meets the immediate needs of users but also lays a foundation for future scalability and innovation in sports facility management. Overall, the turf booking system stands as a testament to the power of technology in revolutionizing traditional processes and enhancing the sports experience for enthusiasts of all levels.

7.2 FUTURE ENHANCEMENT

The future scope for a turf booking system for both turfs and entertainment is vast and promising as society increasingly values convenience and experiences, such a platform holds immense potential. Partnerships with local businesses and event organizers could expand the platform's offerings beyond just turf bookings, encompassing a wide range of entertainment options such as concerts, sports events, and cultural festivals. The future of such a system lies in its adaptability, innovation, and ability to enhance the way people discover, book, and enjoy entertainment .

Here are some potential trends and developments in the future scope of turf booking systems for turfs and entertainment.

- Integration of Augmented Reality (AR) and Virtual Reality (VR)
- Smart Turf Management
- Personalized Recommendations
- Sustainability Initiatives
- Flexible Booking Options
- Mobile-First Approach
- Data Analytics for Insights
- Global Expansion

The future of such a system lies in its adaptability, innovation, and ability to enhance the way people discover, book, and enjoy entertainment experiences.

Additionally, ensuring scalability, security, and user privacy will be crucial for sustaining long-term success in this competitive landscape

APPENDIX 1 SOURCE CODE

LOGIN

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<EditText
    android:id="@+id/editTextMobile"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter Mobile Number"
    android:inputType="phone" />
```

```
<EditText
    android:id="@+id/editTextOTP"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter OTP"
    android:inputType="number" />
```

```
<Button
    android:id="@+id/buttonSendOTP"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
android:layout_below="@id/editTextMobile"
android:text="Send OTP" />
```

```
<Button
    android:id="@+id/buttonVerifyOTP"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextOTP"
    android:text="Verify OTP" />
```

```
</RelativeLayout>
```

LOCATION

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
```

```
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LocationActivity">
```

```
<EditText
```

```
android:id="@+id/locationEditText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Enter location"/>
```

```
<Button
android:id="@+id/getLocationButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/locationEditText"
android:text="Get Current Location"/>
```

```
</RelativeLayout>
```

```
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
```



```

public class LocationActivity extends AppCompatActivity {

    private static final int LOCATION_PERMISSION_REQUEST_CODE = 100;
    private EditText locationEditText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_location);

        locationEditText = findViewById(R.id.locationEditText);
        Button getLocationButton = findViewById(R.id.getLocationButton);

        getLocationButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(LocationActivity.this,
                    Manifest.permission.ACCESS_FINE_LOCATION) !=
                    PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(LocationActivity.this, new
                        String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                        LOCATION_PERMISSION_REQUEST_CODE);
                } else {
                    // Permission already granted
                    getLocation();
                }
            }
        });
    }
}

```

```
}  
});  
}
```

```
private void getLocation() {  
    // Code to get location  
    // You can use LocationManager or FusedLocationProviderClient based on your needs  
    // For simplicity, I'll just show a toast message with a dummy location  
    Location location = new Location("");  
    location.setLatitude(37.7749);  
    location.setLongitude(-122.4194);  
  
    Toast.makeText(this, "Latitude: " + location.getLatitude() + ", Longitude: " +  
        location.getLongitude(), Toast.LENGTH_SHORT).show();  
}
```

```
@Override
```

```
public void onRequestPermissionsResult(int requestCode, @NonNull String[]  
    permissions, @NonNull int[] grantResults) {  
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
    if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {  
        if (grantResults.length > 0 && grantResults[0] ==  
            PackageManager.PERMISSION_GRANTED) {  
            getLocation();  
        } else {  
            Toast.makeText(this, "Location permission denied",  
                Toast.LENGTH_SHORT).show();  
        }  
    }  
}
```

```
}  
}  
}  
}
```

LIST:

```
// MainActivity.java
```

```
public class MainActivity extends AppCompatActivity {
```

```
    private GoogleApiClient googleApiClient;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        // Initialize Google Places API client
```

```
        googleApiClient = new GoogleApiClient
```

```
            .Builder(this)
```

```
            .addApi(Places.GEO_DATA_API)
```

```
            .addApi(Places.PLACE_DETECTION_API)
```

```
            .enableAutoManage(this, /* your connection failed listener */)
```

```
            .build();
```

```
        Button searchButton = findViewById(R.id.search_button);
```

```
        EditText locationEditText = findViewById(R.id.location_edit_text);
```

```

searchButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String location = locationEditText.getText().toString();
        if (!TextUtils.isEmpty(location)) {
            findNearbyPlaces(location);
        } else {
            Toast.makeText(MainActivity.this, "Please enter a location",
            Toast.LENGTH_SHORT).show();
        }
    }
});
}

```

```

private void findNearbyPlaces(String location) {
    // Use Google Places API to find nearby places based on the location
    // Example: Use Places API's Nearby Search or Place Autocomplete
    // Handle the response and update UI with the list of places
}
}

```

CHOOSE SPOT

```
// MainActivity.java
```

```

public class MainActivity extends AppCompatActivity {
    private static final int PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION = 1;

```

```
private boolean locationPermissionGranted;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    // Check and request location permissions  
    if (ContextCompat.checkSelfPermission(this,  
        Manifest.permission.ACCESS_FINE_LOCATION)  
        == PackageManager.PERMISSION_GRANTED) {  
        locationPermissionGranted = true;  
        getLocation();  
    } else {  
        ActivityCompat.requestPermissions(this,  
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},  
            PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION);  
    }  
}
```

```
@Override
```

```
public void onRequestPermissionsResult(int requestCode,  
    @NonNull String[] permissions,  
    @NonNull int[] grantResults) {  
    if (requestCode == PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION) {  
        if (grantResults.length > 0 &&  
            grantResults[0] == PackageManager.PERMISSION_GRANTED) {
```

```
locationPermissionGranted = true;
```

```
getLocation();
```

```
}
```

```
}
```

```
}
```

```
private void getLocation() {
```

```
// Get the user's location using location services
```

```
// This code is omitted for brevity, you'll need to implement it
```

```
// Example: LocationManager or FusedLocationProviderClient
```

```
}
```

```
private void fetchNearbyPlaces(Location location) {
```

```
// Use Google Places API to fetch nearby places based on the user's location
```

```
// This code is omitted for brevity
```

```
}
```

```
// Method to handle place selection
```

```
private void onPlaceSelected(Place selectedPlace) {
```

```
// Handle the selected place, e.g., navigate to another activity or perform an action
```

```
}
```

```
}
```

PAYMENT

```
// Inside your Activity or Fragment class
```

```

// Define variables for payment methods
private static final int PAYMENT_METHOD_UPI = 1;
private static final int PAYMENT_METHOD_CARD = 2;
private static final int PAYMENT_METHOD_NET_BANKING = 3;

// Implement method to initiate payment based on selected payment method
private void initiatePayment(int paymentMethod) {
    switch (paymentMethod) {
        case PAYMENT_METHOD_UPI:
            // Handle UPI payment
            initiateUpiPayment();
            break;
        case PAYMENT_METHOD_CARD:
            // Handle credit/debit card payment
            initiateCardPayment();
            break;
        case PAYMENT_METHOD_NET_BANKING:
            // Handle online banking payment
            initiateNetBankingPayment();
            break;
    }
}

// Method to initiate UPI payment
private void initiateUpiPayment() {
    // Call UPI payment gateway SDK methods
}

```

```
// Method to initiate credit/debit card payment  
private void initiateCardPayment() {  
    // Call credit/debit card payment gateway SDK methods  
}
```

```
// Method to initiate online banking payment  
private void initiateNetBankingPayment() {  
    // Call online banking payment gateway SDK methods  
}
```


APPENDIX 2 SCREENSHOTS

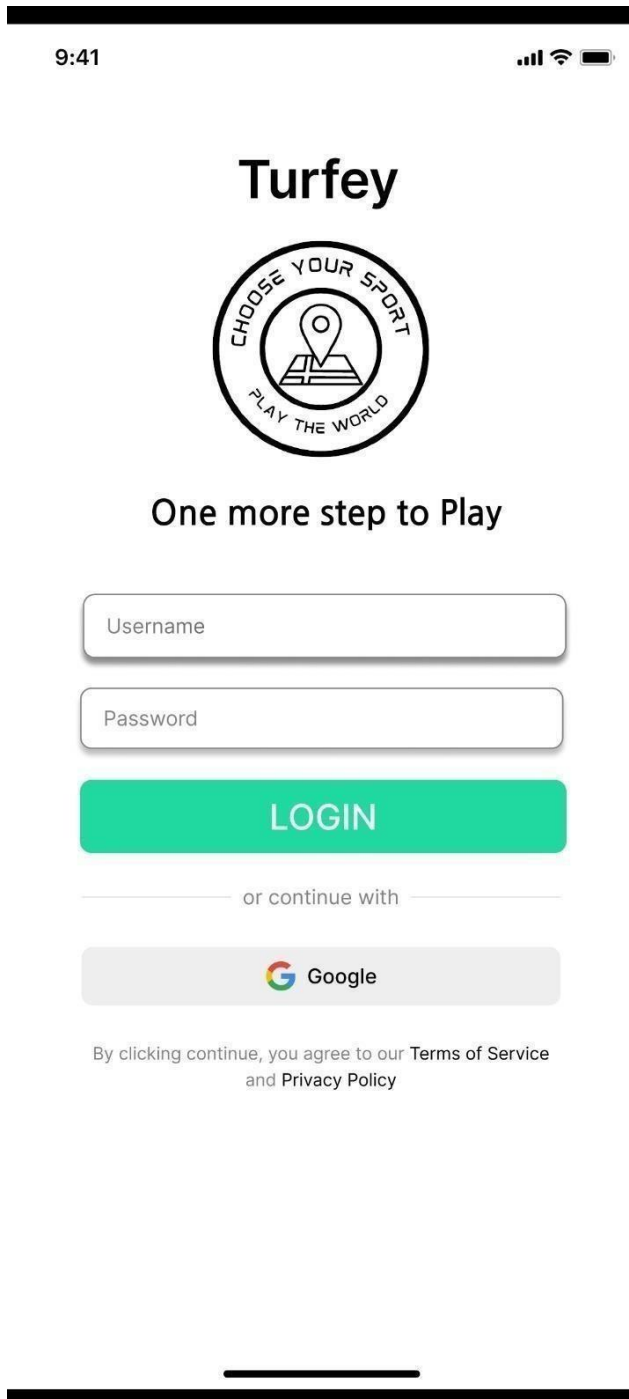
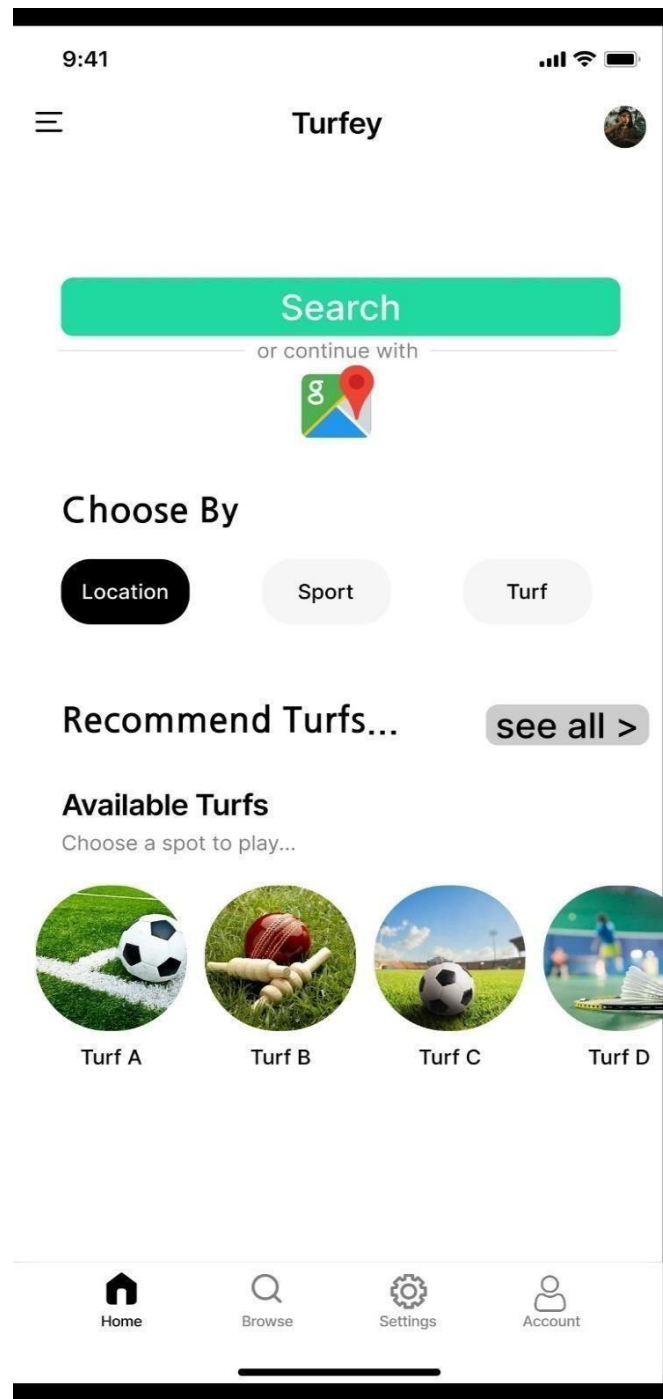


Figure No.A.2.1. Login Page



**Figure No.A.2.2. Recommended Turfs
(nearby turfs)**

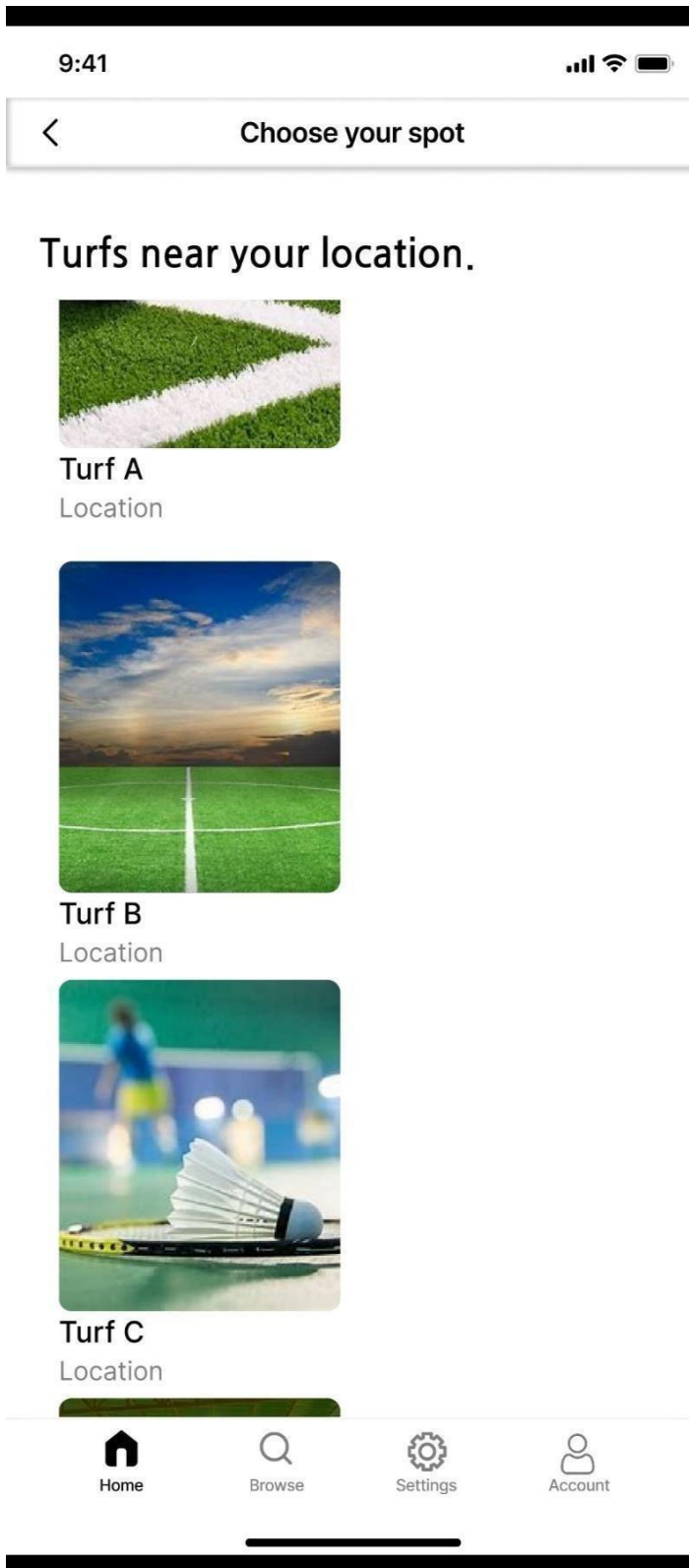


Figure No.A.2.3.Selecting Turfs

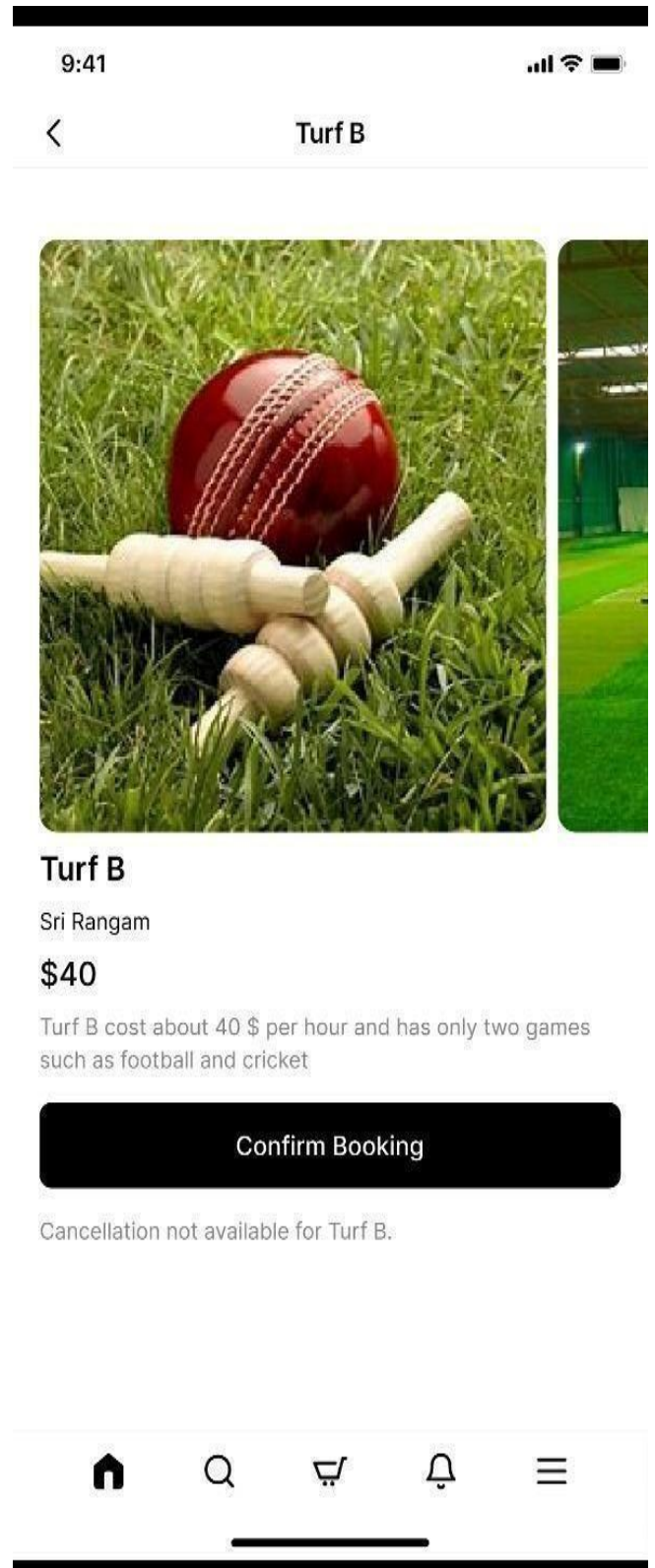


Figure No.A.2.3.Booking

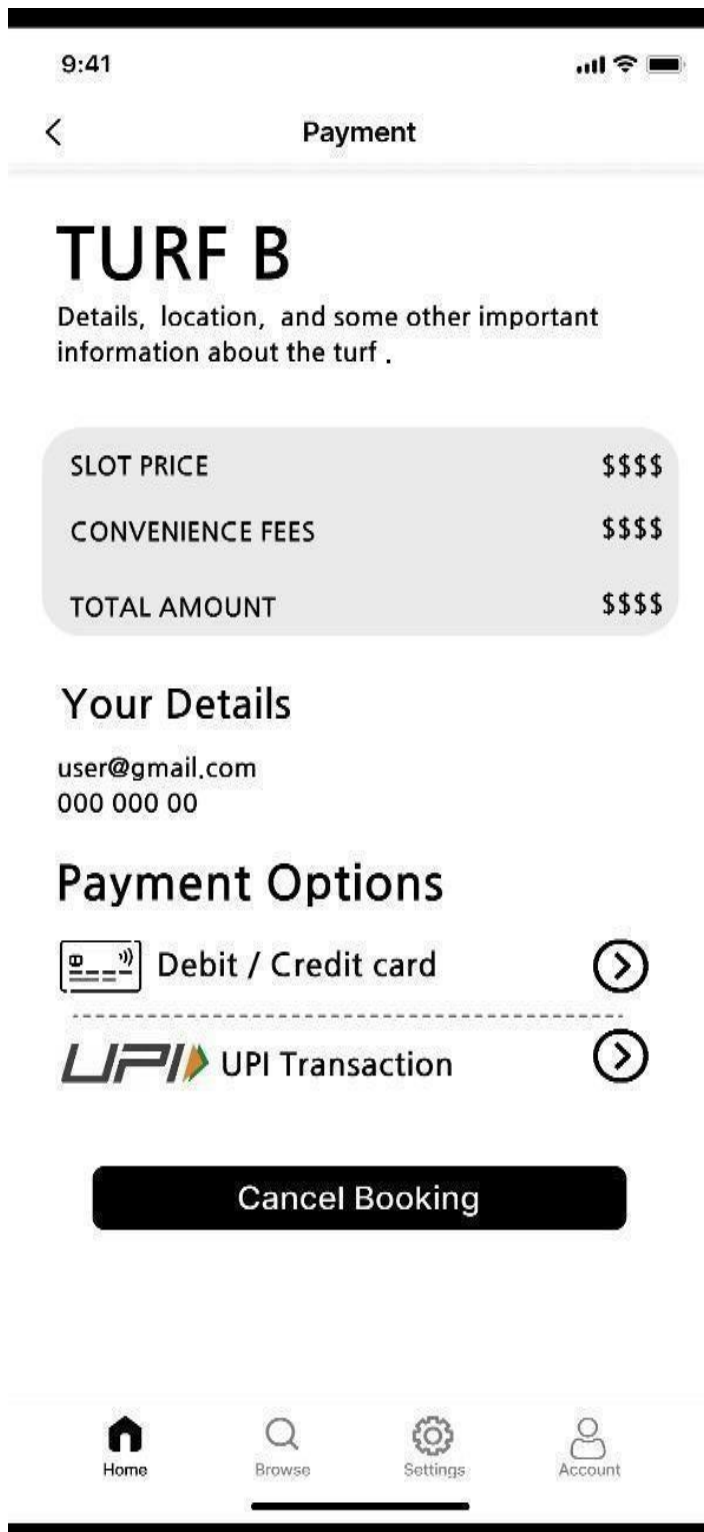


Figure No.A.2.5. Payment

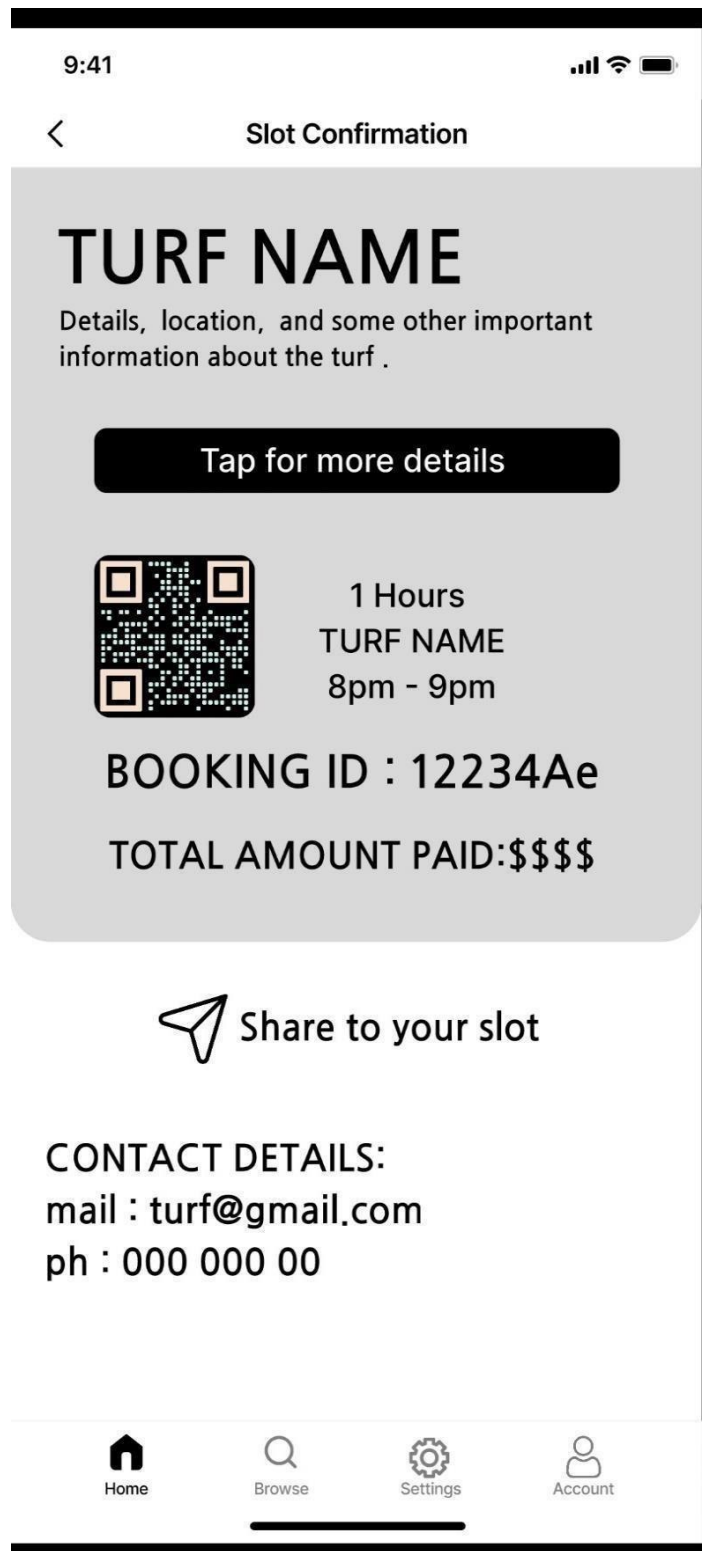


Figure No.A.2.6. Slot Confirmation

REFERENCES

1. Akshay.V, Feb 2020, Department of Computer Science and Engineering, Rajalakshmi EngineeringCollege, Chennai, Tamil Nadu.
2. Alagappan.A.M, Feb 2020, Department of Computer Science and Engineering, Rajalakshmi EngineeringCollege, Chennai, Tamil Nadu.
3. Anish Kumar.S, Feb 2020, Department of Computer Science and Engineering, Rajalakshmi EngineeringCollege, Chennai, Tamil Nadu.
4. Archana.N, Jul 2020, Department of Electronics and Communication Engineering, Sarathy NagarKundrathur, Chennai Institute of Technology, Chennai, India.
5. Desdemona Isabela Scărișoreanu, Apr 2020, Faculty of Industrial and Robotic EngineeringUniversity Politehnica of Bucharest Bucharest, Romania.
6. Dhanagopal.R, Jul 2020 Department of Electronics and Communication Engineering, Sarathy NagarKundrathur, Chennai Institute of Technology, Chennai, India.
7. Gnanavel.S, Feb 2022 Department of Computer Science and Engineering, Rajalakshmi EngineeringCollege, Chennai, Tamil Nadu.
8. Liu Dan, Nov 2020, School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China.
9. Menaka.R, Jul 2020 Department of Electronics and Communication Engineering, Sarathy NagarKundrathur, Chennai Institute of Technology, Chennai, India.
10. Yue KaiDuan,Nov 2020, School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China.