
Software Requirements and Design Document

For

<Pitch Planner>

**Prepared by < Muhammad Moud Ali (22I-1026),
Muhammad Salahuddin (22I-0969), Muhammad Omer (22I-0921) >**

<FAST NUCES>

<27 November 2024 >

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Product Scope.....	1
1.3 Title	1
1.4 Objectives	1
1.5 Problem Statement.....	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions.....	2
2.3 List of Use Cases	3
2.4 Extended Use Cases	3
2.5 Use Case Diagram	16
3. Other Nonfunctional Requirements	17
3.1 Performance Requirements	17
3.2 Safety Requirements.....	17
3.3 Security Requirements	17
3.4 Software Quality Attributes	17
3.5 Business Rules.....	17
3.6 Operating Environment	18
3.7 User Interfaces.....	18
4. Domain Model	27
5. System Sequence Diagram	28
6. Sequence Diagram	35
7. Class Diagram	Error! Bookmark not defined.
8. Package Diagram	Error! Bookmark not defined.
9. Deployment Diagram.....	Error! Bookmark not defined.

1. Introduction

1.1 Purpose

The purpose of this document is to outline the functionalities and requirements of the **Pitch Planner**, focusing on ground booking and management for cricket and football events. This document aims to identify the primary needs of stakeholders, including **Player**, **Ground Owner**, and **Admin**, ensuring seamless organization, scheduling, and conflict resolution. It specifies the high-level features and capabilities required for version **1.0**, covering the entire system to support efficient operations and interactions among stakeholders.

1.2 Product Scope

The **Pitch Planner** is designed to facilitate efficient booking and management of sports grounds, specifically for cricket and football events. This software simplifies the coordination process between players, ground owners, and administrators. Players can easily search for and book grounds, ground owners can manage schedules and pricing, and administrators ensure compliance by handling complaints and verifying ground details. The system enhances operational efficiency and user satisfaction by automating manual processes, promoting transparency, and fostering better communication. By aligning with modern digital transformation goals, the product supports accessible and well-organized sports event management for communities.

1.3 Title

Pitch Planner: A Comprehensive Sports Ground Booking and Management System

1.4 Objectives

The primary objective of **Pitch Planner** is to create an efficient and reliable system for booking and managing sports grounds, addressing the needs of players, ground owners, and administrators. The project aims to:

- Simplify the process of ground reservations by offering a user-friendly interface for scheduling and managing bookings.
- Enable ground owners to easily add, update, or remove grounds while maintaining detailed booking records.
- Provide players with features such as match-up facilitation, feedback submission, and complaint filing to enhance their experience.
- Equip administrators with tools to verify grounds, resolve complaints, and maintain system integrity.
- Ensure smooth communication between all stakeholders while improving overall operational efficiency and transparency.

1.5 Problem Statement

Managing sports ground bookings and match coordination has traditionally been a time-consuming and inefficient process, relying heavily on manual record-keeping, physical interactions, and limited communication between players and ground owners. This lack of automation often results in scheduling conflicts, mismanagement, and reduced satisfaction for both ground owners and players. Additionally, there is no streamlined mechanism for handling complaints or verifying grounds, which compromises the quality and integrity of the services offered.

Pitch Planner addresses these challenges by providing a fully automated, digital solution for ground booking, management, and communication. By integrating functionalities such as match-up facilitation, complaint resolution, and ground verification, the platform not only mitigates manual effort but also enhances transparency and efficiency for all stakeholders. This approach ensures feasibility by leveraging modern technologies, enabling scalable and adaptable solutions that align with user needs and expectations.

2. Overall Description

2.1 Product Perspective

Pitch Planner is a standalone platform developed to address the specific needs of players, ground owners, and administrators in managing sports grounds effectively. This is a self-contained system that integrates booking, payments, complaint resolution, ground management, and user role-based functionality. The system provides a streamlined user experience by enabling players, ground owners, and administrators to manage tasks through distinct interfaces tailored to their roles.

As depicted in the use case diagram, the product caters to three primary actors:

1. **Player:** Responsible for booking grounds, making payments, searching for opponents, giving feedback, and filing complaints.
2. **Ground Owner:** Manages ground-related activities such as adding, updating, or removing grounds and handling bookings.
3. **Admin:** Handles the verification of grounds, resolving complaints, and blacklisting users or grounds when required.

2.2 Product Functions

❖ For Player:

- Book sports grounds and search for match-up opponents.
- Make payments using Easypaisa integration.
- Provide feedback on completed bookings.

- Cancel upcoming bookings if necessary.
- File complaints against any issues with bookings or grounds.

❖ **For Ground Owner:**

- Add, update, or remove grounds from the system.
- View and manage scheduled bookings.
- Register new grounds for verification by the admin.

❖ **For Admin:**

- Verify new ground registrations submitted by ground owners.
- Resolve complaints filed by players or ground owners.
- Blacklist problematic grounds or users to maintain system integrity.

2.3 List of Use Cases

- 1) Book the Ground
- 2) Make Payment
- 3) Look for Opponent
- 4) View Ground Detail
- 5) Give Feedback
- 6) Cancel Booking
- 7) Register/Login
- 8) File Complaint
- 9) Manage Ground
- 10) Verify Ground Registration
- 11) Blacklist User
- 12) Blacklist Ground
- 13) Resolve Complaint

2.4 Extended Use Cases

1) Use Case: Book the Ground

Use Case Name: Book the Ground

Scope: Pitch Planner

Level: User Goal

Primary Actor: Player

Stakeholders and Interests:

- **Player:** Wants to book a ground for a specific date and time for their event and has the option to look for an opponent team if needed.
- **Pitch Planner:** Manages ground bookings, ensures availability, and facilitates connections between teams.
- **Ground Owner:** Oversees scheduling and ensures bookings meet the required standards.

Preconditions:

- The player must be logged into the system.
- The player must have a valid payment method.
- The ground must be available for booking on the requested date and time.

Postcondition:

- The ground is successfully booked for the player.
- If the player is looking for an opponent, the booking is flagged in the system as “Looking for Opponent.”
- The booking details are stored in the system.
- A confirmation message is sent to both the player and the ground owner.

Main Success Scenario:

User Action	System Response
1. The player selects a ground, desired date and time.	2. The system checks availability for the selected slot.
3. The player confirms the booking.	4. The system confirms the booking.

Extensions:

- **2a:** The player selects a date and time that is already booked.
The system shows an error message and suggests to select alternative available slot.
- **3a:** The player request Match-Up.
Six digit Match-Up code is generated and the booking is confirmed.

Technology and Data Variations:

Multiple times a day, depending on the number of players and available grounds.

2) Use Case: Make Payment

Use Case Name: Make Payment

Scope: Pitch Planner

Level: User Goal

Primary Actor: Player

Stakeholders and Interests:

- **Player:** Wants to successfully complete the booking by paying for the ground.
- **Ground Owner:** Wants to receive payment for the booking.
- **Easypaisa:** Ensures the transaction is processed securely.

Preconditions:

- The player has confirmed booking.
- The player must be logged into the system.
- The player must have a valid payment method.

Postcondition:

- Payment is successfully processed and payment details are saved in the system.
- The player receives a confirmation of the payment.

Main Success Scenario:

User Action	System Response
1. The player proceeds to Make Payment	2. The system displays all the pending payments.
3. The player selects a payment.	
4. The player enters easypaisa number and confirms payment.	5. The system connects to the Easypaisa and processes the payment.
	6. The payment is successfully processed and payment status is updated.

Extensions:

- **2a:** No pending payments.
The player returns to player dashboard.
- **4a:** Player enters incorrect payment information.
The system displays an error message prompting the player to re-enter correct details.
- **6a:** Payment fails (due to insufficient funds, network issues, etc.).
The system informs the player of the failure and suggests trying a different payment method. The player retries the payment or cancels the booking process.

Special Requirements:

- The payment system must be integrated with secure payment gateways (Easypaisa) to ensure encrypted transactions.
- The player should be able to save payment details for future use.
- The system must comply with relevant regulations (e.g., PCI-DSS) to protect financial information.

Technology and Data Variations:

- **Payment Methods:** Easypaisa

3) Use Case: Match-Up**Use Case Name:** Match-Up**Scope:** Pitch Planner**Level:** User Goal**Primary Actor:** Player**Stakeholders and Interests:**

- **Player:** Wants to set match with another team

- **Pitch Planner:** Ensures ground bookings are used efficiently and notifies users of any scheduling issues in advance.
- **Opponent Team:** Looking for team to play against.

Preconditions:

- The opponent team must have successfully booked a ground and chosen to mark the booking as "Looking for Opponent".
- Both teams (booking team and opponent-seeking team) must be registered and logged into the system.

Postcondition:

- A match is successfully scheduled between the two teams on the booked ground.
- The system splits the booking cost between the two teams upon successful matchup.
- The system notifies both teams of the scheduled match.

Main Success Scenario:

User Action	System Response
1. The player enters a 6-digit Match-up code.	2. The system shows the booking details, including location, date, time.
5. The player confirms match-up.	4. The system confirms the match-up between the two teams, updates the booking status to "Match-Up" and splits the booking cost evenly between both teams.

Extensions:

- ☐ 1a: Invalid code.
 - Prompts the player to enter valid code.
- ☐ 1b: No match-up request for the entered code.
 - The system displays a message that no team is seeking opponents for this code.

Special Requirements:

- The system should update match-up status in real time to prevent double bookings.
- **Cost Split Calculation:** If a match-up is successfully set, the system should divide the booking cost evenly between both teams, displaying each team's share in the confirmation.

Technology and Data Variations:

- **Notifications:** Notifications can be sent via email, SMS, or in-app alerts, depending on user preferences.
- **Match Status Display:** Match status should be visible on the booking calendar with distinctions between "Looking for Opponent", "Match-up", and "Confirmed".

4) Use Case: View Ground Detail

Use Case Name: View Ground Details

Scope: Pitch Planner

Level: User Goal

Primary Actor: Player

Stakeholders and Interests:

- **Player:** Wants to view detailed information about a specific ground to make an informed booking decision.
- **Pitch Planner:** Ensures accurate and up-to-date information about each ground is displayed.
- **Ground Owner:** Needs to ensure the displayed details about the ground are accurate and current.

Preconditions:

- The player must be logged into the system.
- Ground information must be updated and available in the system.

Postcondition:

- The player views detailed information about the selected ground.
- The player can proceed with booking or exit after viewing the details.

Main Success Scenario:

User Action	System Response
1. The player filters the ground based on type, location and price.	2. The system displays a list of grounds with details.

Extensions:

- **2a:** No grounds available: The system displays a message indicating no grounds are available and suggests alternatives or asks the player to check back later.

Special Requirements:

- The system must update ground details regularly to reflect any changes in availability, pricing, or conditions.
- The player should be able to filter or search for specific types of grounds.

5) Use Case: Give Feedback

Use Case Name: Give feedback

Scope: Pitch Planner

Level: User Goal

Primary Actor: Player

Stakeholders and Interests:

- **Player:** Wants to provide feedback on the ground they booked and used, sharing their experience for other potential users.

- **Pitch Planner:** Ensures that reviews are managed and moderated properly, providing users with helpful and accurate feedback while maintaining quality standards.

Preconditions:

- The player must have successfully completed a booking and used the ground.
- The player must be logged into their account.

Postcondition:

- The system updates the average rating based on new reviews.

Main Success Scenario:

User Action	System Response
1. The player access to the "My Booking".	2. The system displays a list of bookings.
3. The player selects the booking they want to review and provide rating and comment.	4. The system saves the review and updates the overall rating for the ground.

Extensions:

- **2a:** No completed bookings available: The system displays a message informing the player that reviews can only be given for completed bookings.
- **3a:** submits a review without a rating or comment: The system prompts the player or ground owner to provide both a rating and comment before submission.

Special Requirements:

- The system must prevent players from reviewing the same booking more than once.
- Reviews should be moderated for inappropriate content.

Technology and Data Validation List:

- **Rating System:** Reviews can be based on a 1-5 star rating system with optional written feedback.

6) Use Case: Cancel Booking**Use Case Name:** Cancel Booking**Scope:** Pitch Planner**Level:** User Goal**Primary Actor:** Player**Stakeholders and Interests:**

- **Player:** Wants to cancel a booking smoothly without incurring unnecessary penalties.
- **Pitch Planner:** Ensures that cancellations are processed efficiently, updates availability, and handles penalties/refunds as per policy.
- **Ground Owner:** Needs to be informed about cancellations to free up the slot and manage any financial impact.

Preconditions:

- The player must be logged into the system.
- The player must have an active booking.
- Cancellation policies (penalties, time limits) must be defined and visible.

Postcondition:

- The booking is successfully canceled, and the player receives confirmation.
- The system updates the ground's availability.
- The player receives a partial or full refund (if applicable) based on the policy.
- The ground owner is notified of the cancellation.

Main Success Scenario:

User Action	System Response
1. The player access to the "My Bookings" section.	2. The system displays the player's current bookings.
3. The player selects the booking they wish to cancel.	4. The system prompts for cancellation confirmation and cancels the booking.

Extensions:

- **3a:** Booking not eligible for cancellation (e.g., past allowed window): The system informs the player that the booking cannot be canceled and provides reasons.

Special Requirements:

- Real-time updates to ground availability after cancellation must be ensured.
- The system must handle and process penalties and refunds securely and accurately.

Technology and Data Validation List:

- **Refund Process:** Integration with the payment gateway for handling partial and full refunds as applicable.

7) Use Case: File Complaint

Use Case Name: File Complaint

Scope: Pitch Planner

Level: User Goal

Primary Actor: Player and Ground Owner

Stakeholders and Interests:

- **Player:** Wants to submit complaint of ground or/and ground owner.
- **Player:** Wants to provide complaint on the player.

Preconditions:

- The player must have successfully completed a booking and used the ground.
- The player and ground owner must be logged into their account.

Postcondition:

- The system forwards the complaint to the admin.

Main Success Scenario:

User Action	System Response
1. The user selects an already completed booking to file complaint.	
2. The user enters the issue and submit it.	3. The system saves the complaint and forwards it to admin.

Extensions:

- **1a:** No completed bookings available: The system displays a message informing the user that complaint can only be submitted for completed bookings.

8) Use Case: Register/Login**Use Case Name:** Register / Login**Scope:** Pitch Planner**Level:** User Goal**Primary Actor:** User (Player and Ground Owner)**Stakeholders and Interests:**

- **Player:** Wants to create an account or log in to book grounds and manage bookings.
- **Ground Owner:** Needs to create an account or log in to manage ground availability, view bookings, and handle payments.
- **Pitch Planner:** Manages user authentication and ensures secure access for players and ground owners.

Preconditions:

- The player or ground owner must have access to the internet.
- The registration or login page must be available and operational.

Postcondition:

- The player or ground owner is successfully logged into their account.
- Registered users have access to relevant system features (booking, managing grounds).
- New players or ground owners have an account created and can access system features after verification.

Main Success Scenario:

User Action	System Response
-------------	-----------------

1. The user selects: - Register (if new) or - Login (if returning).	Registration Flow: 2. If the user Register: The system displays the registration form with fields.
3. The user fills in the required details and Submits.	4. The system confirms registration and logs the user in.
	Login Flow: 2. If the user selects Login: - The system prompts for: - Username - Password
3. The player or ground owner enters their credentials and clicks Login.	4. The system validates the credentials and access to their dashboard

Extensions:

- **Login Flow 4a:** Player or ground owner enters incorrect login details: The system displays an error message and prompts to try again or reset the password.

Special Requirements:

- The system must support secure authentication methods, including encryption of sensitive data like passwords.
- User sessions must be managed securely with the ability to log out and time out inactive sessions.
- The system should include a CAPTCHA or similar mechanism to prevent bots from registering.

Technology and Data Validation List:

Role-Based Access: Players and Ground Owners see different dashboards and features upon login.

9) Use Case: Manage Ground

Use Case Name: Manage Ground

Scope: Pitch Planner

Level: User Goal

Primary Actor: Ground Owner

Stakeholders and Interests:

- **Ground Owner:** Wants to add or update a ground's details, such as location, facilities, and pricing, to make it available for bookings by players..
- **Players:** Interested in viewing and booking grounds based on up-to-date details, such as location, pricing, and available facilities.
- **Pitch Planner:** Ensures complete and accurate ground listings to increase player engagement and booking platform utilization.

Preconditions:

- The ground owner must be logged into the system.
- The ground owner should have valid details about the ground (location, pricing, availability).

- For updating a ground, the ground must already exist in the Pitch Planner system.
- The system provides an option for managing grounds (adding or editing ground information).

Postcondition:

- The new or updated ground information is successfully added to the database and is visible for booking by players.
- The ground owner receives a notification confirming the successful addition or update of the ground.
- Players can view and book the ground with accurate and current information.

Main Success Scenario:

User Action	System Response
1. The ground owner navigates to the "Manage Grounds" section.	2. The system displays options to "Add Ground" or view a list of owned grounds for updates.
3. The ground owner enters or updates ground details, including: - Ground Name - Location (address, city, etc.) - Type of ground (e.g., football, cricket, tennis) - Capacity (number of players/teams) - Available facilities (e.g., lights, seating, parking) - Pricing (per hour, per match, etc.) - Availability (times and dates when the ground is available) The ground owner submits the form for a new ground or saves the changes for an update	4. The system saves the new or updated ground information, confirming successful addition or update.

Extensions:

- **3a:** Ground owner submits incomplete or incorrect details: The system highlights the missing or incorrect fields and prompts the owner to correct them before submission.
- **4a:** System error during submission: The system informs the ground owner of the issue and prompts them to try again later.

10) Use Case: Verify Ground Registration**Use Case Name:** Verify Ground Registration**Scope:** Pitch Planner**Level:** User Goal**Primary Actor:** Admin**Stakeholders and Interests:**

- **Admin:** Responsible for verifying the registration of new grounds ensuring they meet the required standards for booking on the platform.
- **Ground Owners:** Want to ensure that their registrations are verified so they can list their grounds for bookings.

Preconditions:

- The admin must be logged into the Pitch Planner system.
- New ground registrations must be submitted and awaiting verification.

Postcondition:

- The registration status of the ground is updated to either "verified" or "rejected".

Main Success Scenario:

User Action	System Response
1. The admin access to the "Ground Verification" section.	2. The system displays a list of new ground registrations awaiting verification.
3. The admin selects a ground registration for review and either verify it or reject it.	4. The system updates the ground's registration status accordingly: - If verified, the system changes the status to "verified" and sends a confirmation email/SMS to the ground owner. - If rejected, the system changes the status to "rejected" and sends a notification to the ground owner with the rejection reason.

Special Requirements:

- The system should maintain a clear audit trail of all verification actions taken by admins for accountability.
- The admin interface should be user-friendly to facilitate efficient processing of multiple registrations.

Technology and Data Validation List:

- **Notification System:** Users receive notifications via email or SMS based on their preference settings when their registration status is updated.

11) Use Case: Blacklist User

Use Case Name: Blacklist User

Scope: Pitch Planner

Level: User Goal

Primary Actor: Admin

Stakeholders and Interests:

- **Admin:** Responsible for banning users who violate platform policies or fail to meet standards, ensuring the integrity of the Pitch Planner community.
- **Players (Customers):** Seek a safe and respectful environment; they benefit from the admin's ability to enforce standards and remove disruptive users.
- **Ground Owners:** Interested in maintaining a positive atmosphere for their grounds, benefiting from the ability to report and have problematic users banned.

Preconditions:

- The admin must be logged into the Pitch Planner system.
- The admin must have the authority to ban users (based on reported behavior or ratings).
- The user to be banned must exist in the system.

Postcondition:

- The user is successfully banned from using the Pitch Planner system on specific grounds.
- The system logs the ban action for accountability and records management.
- The banned user is notified of the ban and the reason for it.

Main Success Scenario:

User Action	System Response
1. The admin selects the user and blacklist it.	2. The system blacklists the selected user and update the info.

Extensions:

- **1a:** Admin finds that the user does not exist: The system informs the admin that no such user exists.

Special Requirements:

- The system should provide clear logging of all ban actions, including timestamps

Technology and Data Validation List:

- **User Feedback:** The system should track user feedback and complaints to help inform banning decisions.

12) Use Case: Blacklist Ground

Use Case Name: Blacklist the Ground

Scope: Pitch Planner

Level: User Goal

Primary Actor: Admin

Stakeholders and Interests:

- **Players:** Benefit from accurate listings; removing a ground should ensure players do not attempt to book a ground that is no longer available.
- **Pitch Planner:** Ensures the platform only lists active and available grounds for booking, maintaining an accurate database.

Preconditions:

- The admin must be logged into the Pitch Planner system.
- No active bookings should be associated with the ground at the time of blacklisting.

Postcondition:

- The ground is successfully removed from the Pitch Planner database and is no longer visible to players.

- The system logs the removal action for accountability.
- Players attempting to access the removed ground receive a notification that the ground is no longer available.

Main Success Scenario:

User Action	System Response
1. The admin selects the ground and blacklist it.	2. The system blacklists the selected ground and update the info.

Extensions:

- **2a:** Active bookings are found: The system informs the ground owner that they cannot remove the ground while there are ongoing bookings and suggests they cancel those first.

Special Requirements:

- User notifications should be timely and informative to ensure players are aware of any changes in availability.

Technology and Data Validation List:

- **Booking System:** Must accurately track active bookings to prevent issues when a ground is being removed.
- **User Notification System:** Should effectively communicate to players about the removal of grounds they may have been interested in.

13) Use Case: Resolve Complaint**Use Case Name:** Resolve Complaint**Scope:** Pitch Planner**Level:** User Goal**Primary Actor:** Admin**Stakeholders and Interests:**

- **Admin:** Wants to resolve complaints for better user experience.
- **User:** Wants their complaint to be resolved.

Preconditions:

- The admin must be logged into the Pitch Planner system.
- There should be unresolved complaints.

Postcondition:

- The complaints resolved successfully.
- Users are notified about the actions taken.

Main Success Scenario:

User Action	System Response
1. The admin views all unresolved complaints.	2. The system displays unresolved complaints.
3. The admin resolves the complaints.	4. The systems updates the complaint status

Extensions:

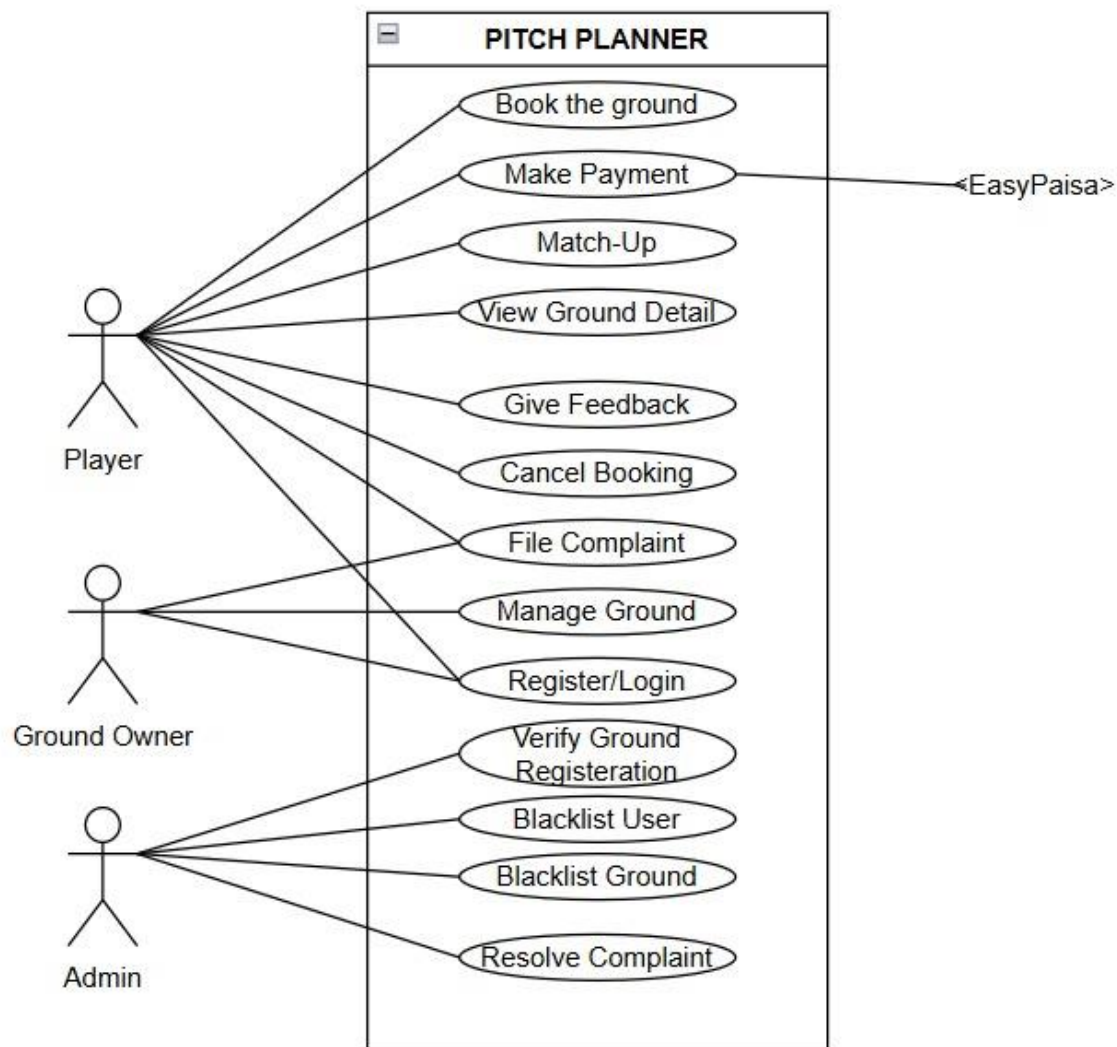
- **2a:** No unresolved complaints are found: admin returns to admin dashboard.

Special Requirements:

- User notifications should be timely and informative to ensure users are aware of any changes in availability.

Technology and Data Validation List:

- **User Notification System:** Should effectively communicate to users about how the complaints are handled.

2.5 Use Case Diagram

3. Other Nonfunctional Requirements

3.1 Performance Requirements

The system must ensure a seamless and responsive experience for all users. For bookings and payments, transactions should be processed in less than **2 seconds** under normal network conditions. The system should handle up to **500 simultaneous users** without any degradation in performance. Real-time updates for match-up codes and ground availability must be refreshed within **1 second** to maintain usability.

3.2 Safety Requirements

The system must ensure data integrity and prevent loss or corruption of critical booking or payment data. Automated backups must be performed daily to avoid any potential data loss. Additionally, users must be restricted from performing illegal operations, such as duplicate bookings or malicious complaints. Compliance with local digital transaction laws (such as Easypaisa regulations) ensures safe monetary transactions.

3.3 Security Requirements

The system must enforce secure login and user authentication protocols, such as encrypted passwords (using hashing algorithms). All sensitive user data, including payment details, should be encrypted. Only authenticated and authorized users should access their respective modules. The admin must have oversight for blacklisting grounds or users. Security against SQL injection, cross-site scripting (XSS), and other vulnerabilities must be implemented.

3.4 Software Quality Attributes

- **Usability:** The UI must be intuitive for all user roles (players, ground owners, and admin). Minimal training should be required.
- **Reliability:** The system must have 99.9% uptime, ensuring availability during high-demand periods.
- **Scalability:** The application should support future enhancements, such as adding more sports or integrating new payment gateways.
- **Maintainability:** The system should use a modular architecture, enabling easy updates and bug fixes without disrupting operations.
- **Testability:** All modules must be unit-tested with at least 90% code coverage to ensure robustness.
- **Adaptability:** The system should function on various devices, including desktops, laptops, and tablets, across modern browsers.

3.5 Business Rules

- Players can only book a ground if they have completed payment via the integrated Easypaisa gateway.
- Admin must verify all grounds before they are available for booking.
- Complaints can only be resolved or grounds/user blacklisted by the admin.
- Ground owners can only manage grounds they have registered.
- Players cannot cancel bookings after the booking time has passed.
- Match-up codes are valid for a limited time to ensure timely matches.

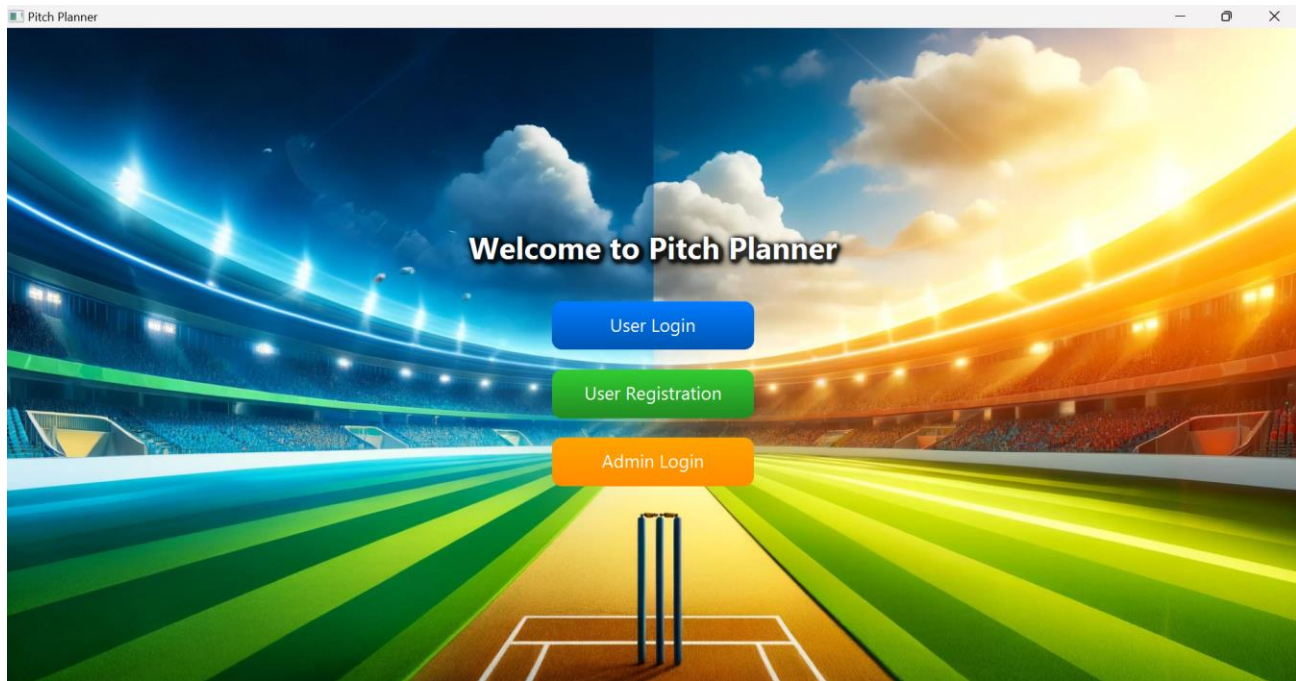
3.6 Operating Environment

The system will operate in a **web-based environment** and support:

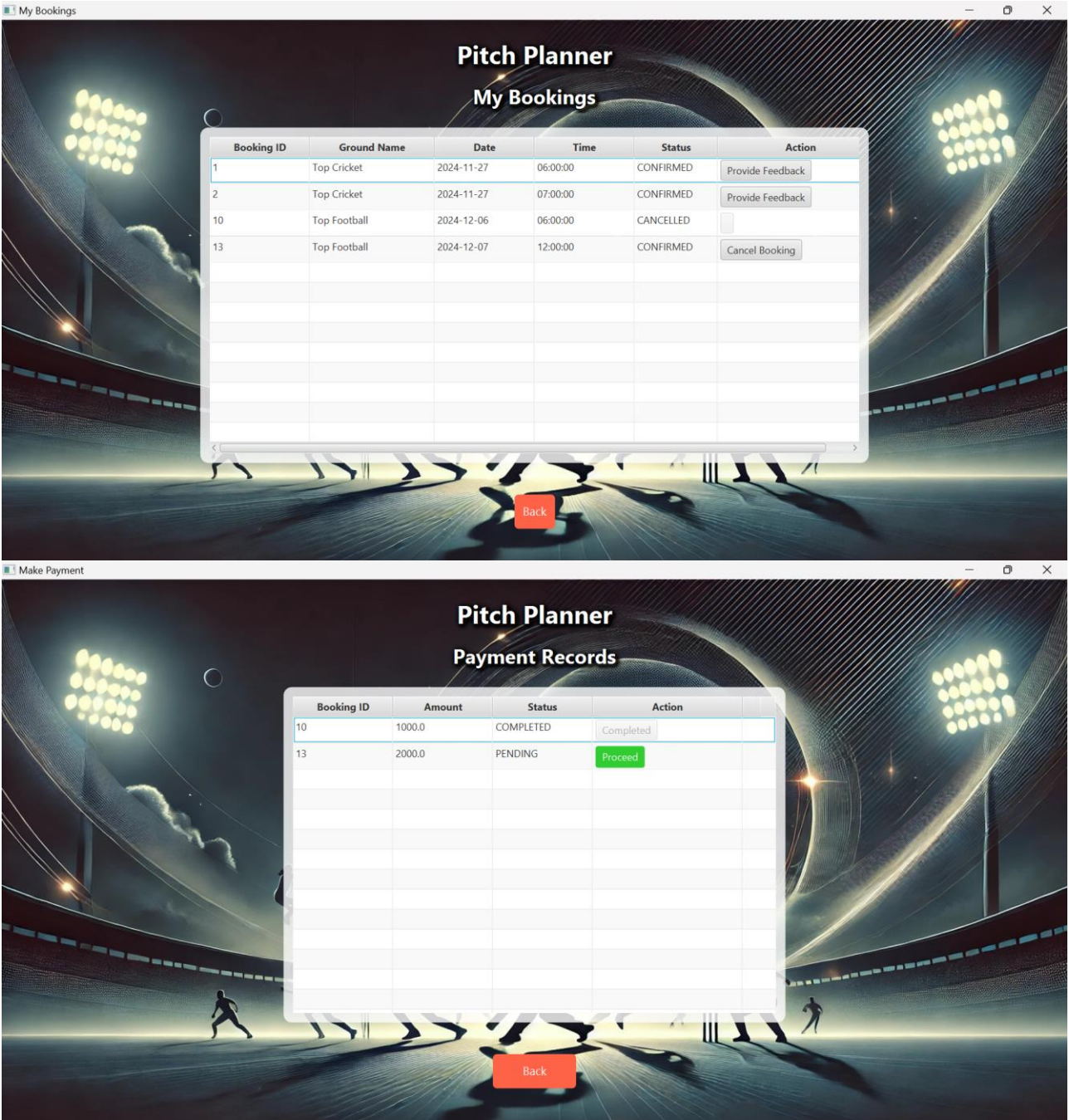
- **Hardware Platforms:** Desktop or laptop with a minimum of 4GB RAM and dual-core processor.
- **Operating Systems:** Windows 10+, macOS 10.15+, and Linux (Ubuntu 20.04 or above).
- **Backend Environment:** Java, JavaFx and MySQL
- **Payment Gateway:** Easypaisa API integration for secure transactions.

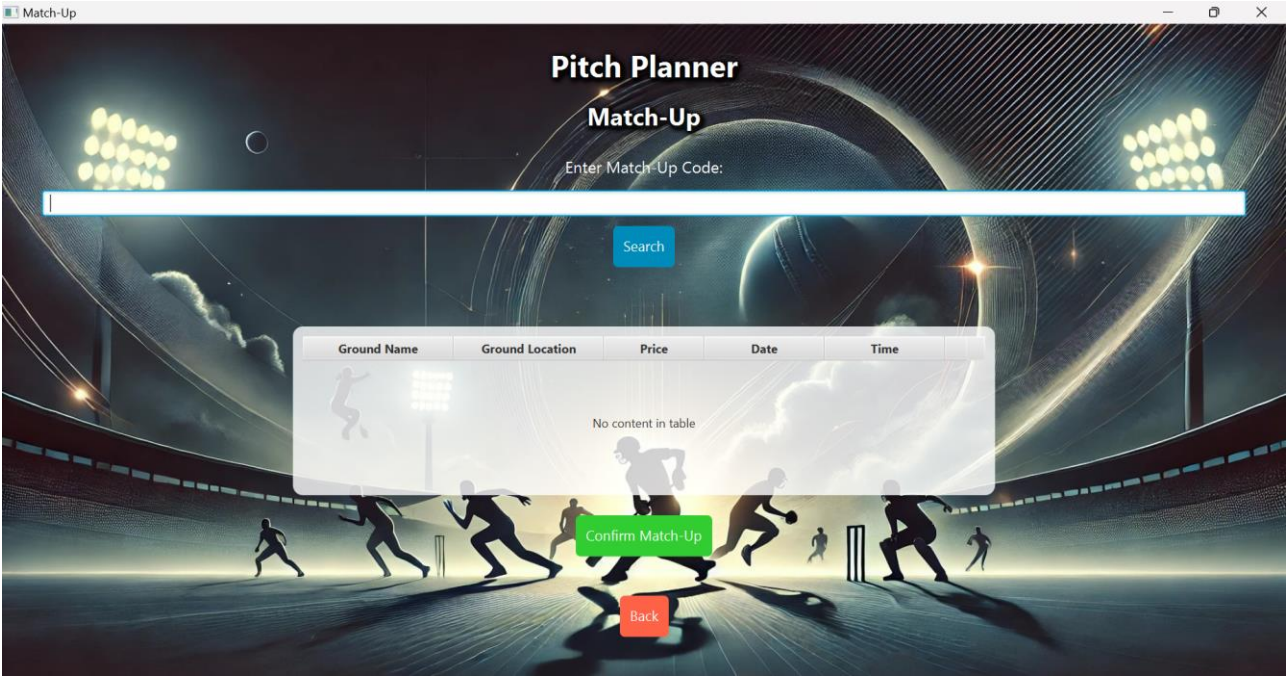
3.7 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>









The image displays two screenshots of a web application interface. The top screenshot shows the 'GROUND_OWNER Dashboard' with a welcome message 'Welcome, Ali!' and a central menu with five buttons: 'Add Ground' (green), 'Update Ground' (orange), 'Remove Ground' (red), 'Scheduled Bookings' (blue), and 'Logout' (purple). The background features a collage of sports images including basketball, cricket, and soccer. The bottom screenshot shows the 'Pitch Planner' 'Add Ground' form. It includes input fields for 'Ground Name:', 'Location:' (with placeholder text 'Enter ground location'), and 'Price (per hour):' (with placeholder text 'Enter price'). Below these is a 'Ground Type:' dropdown menu with a 'Select type' placeholder. At the bottom of the form are two buttons: 'Add Ground' (green) and 'Back' (red). The background of the form is a stylized image of a cricket pitch at night.

GROUND_OWNER Dashboard

Welcome, Ali!

Ground Owner Dashboard

Add Ground

Update Ground

Remove Ground

Scheduled Bookings

Logout

Add Ground

Pitch Planner

Add Ground

Ground Name:

Location:

Enter ground location

Price (per hour):

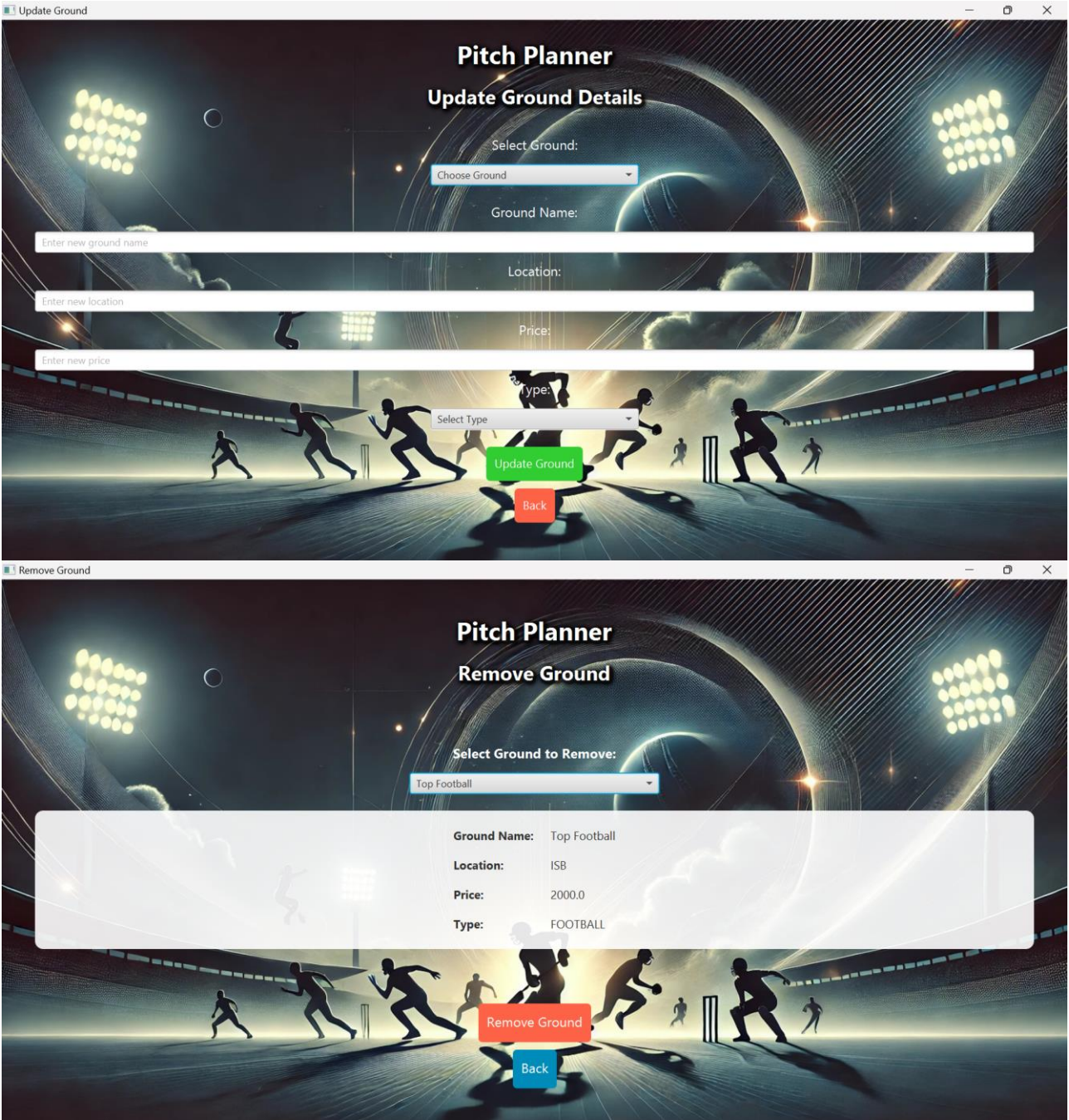
Enter price

Ground Type:

Select type

Add Ground

Back



Scheduled Bookings

Pitch Planner

Scheduled Bookings

Select Ground:

Top Cricket

Booking ID	Player Name	Date	Time	Price	Status	Action
17	Salu Pacer	2024-11-01	06:00:00	1500.0	CONFIRMED	Complain
11	Ibrahim	2024-11-25	14:00:00	1500.0	CONFIRMED	Complain
5	Omer	2024-11-26	06:00:00	1500.0	CONFIRMED	Complain
1	Mould Ali	2024-11-27	06:00:00	1500.0	CONFIRMED	Complain
2	Mould Ali	2024-11-27	07:00:00	1500.0	CONFIRMED	Complain
12	jawad	2024-11-27	10:00:00	1500.0	CONFIRMED	Complain
9	Ibrahim	2024-11-28	13:00:00	1500.0	CONFIRMED	Complain
15	Player4	2024-12-05	08:00:00	1500.0	PENDING	Complain

Back

User Registration

Register for Pitch Planner

Name:

Username:

Password:

Confirm Password:

Age:

City:

Phone:

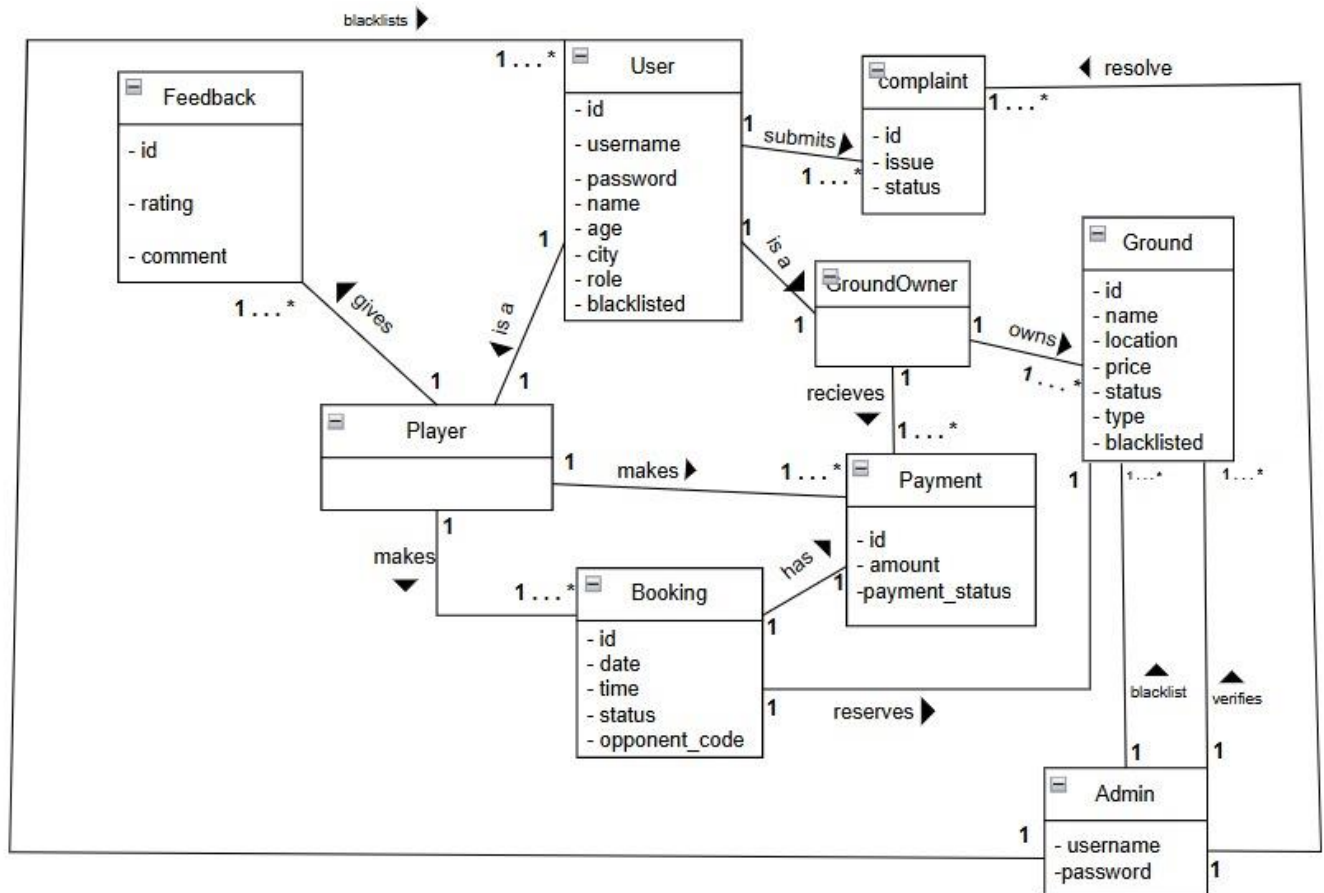
Role: ☐ Player ☐ Ground Owner

Register

Back

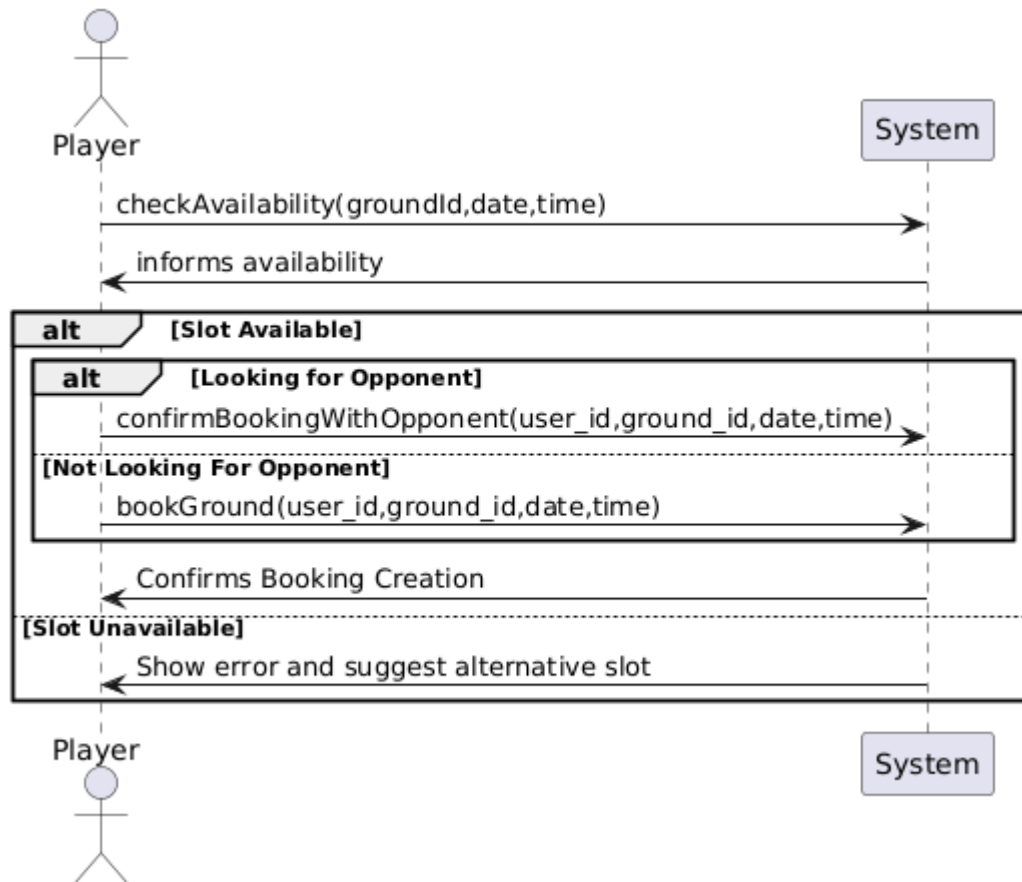


4. Domain Model

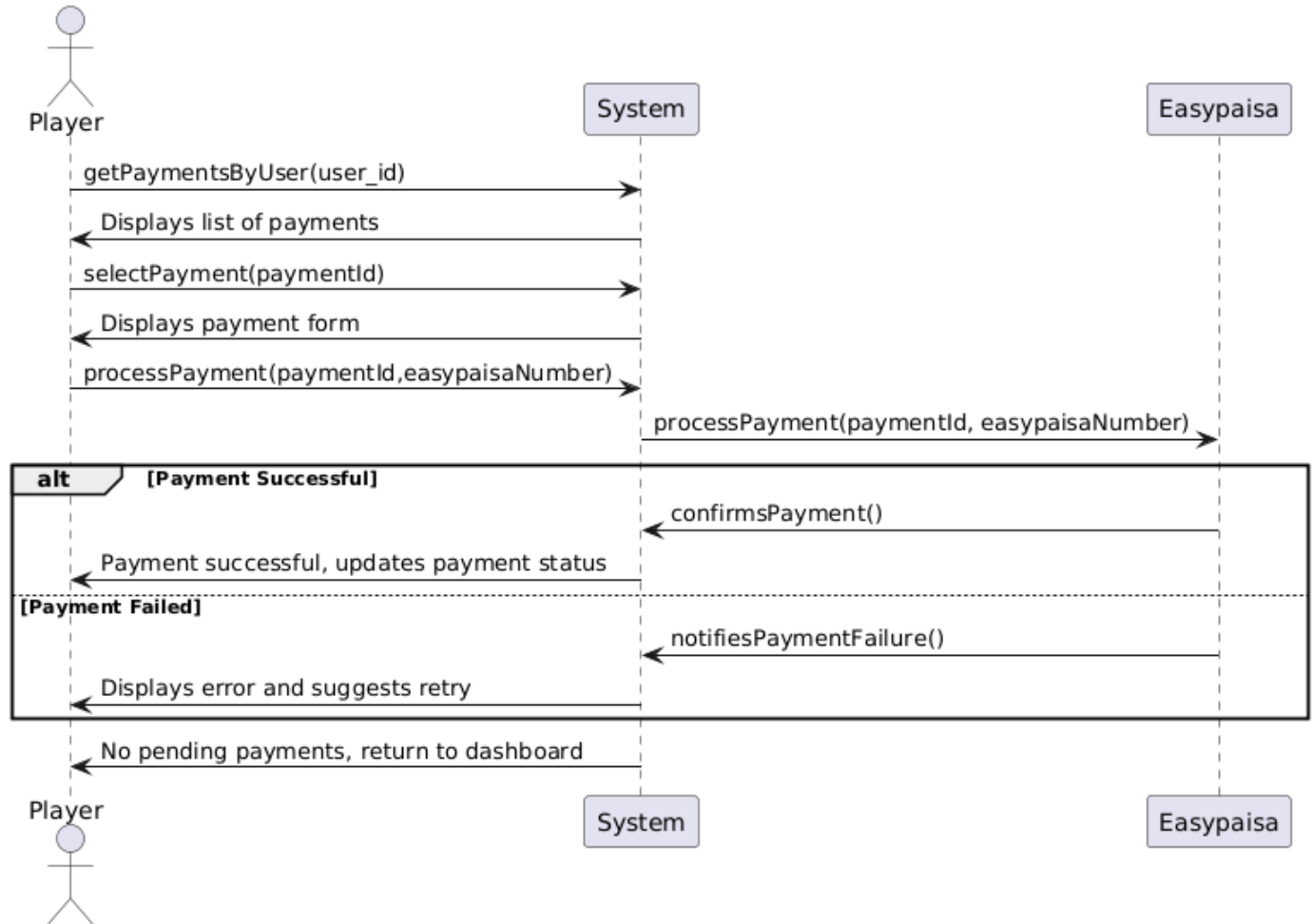


5. System Sequence Diagram

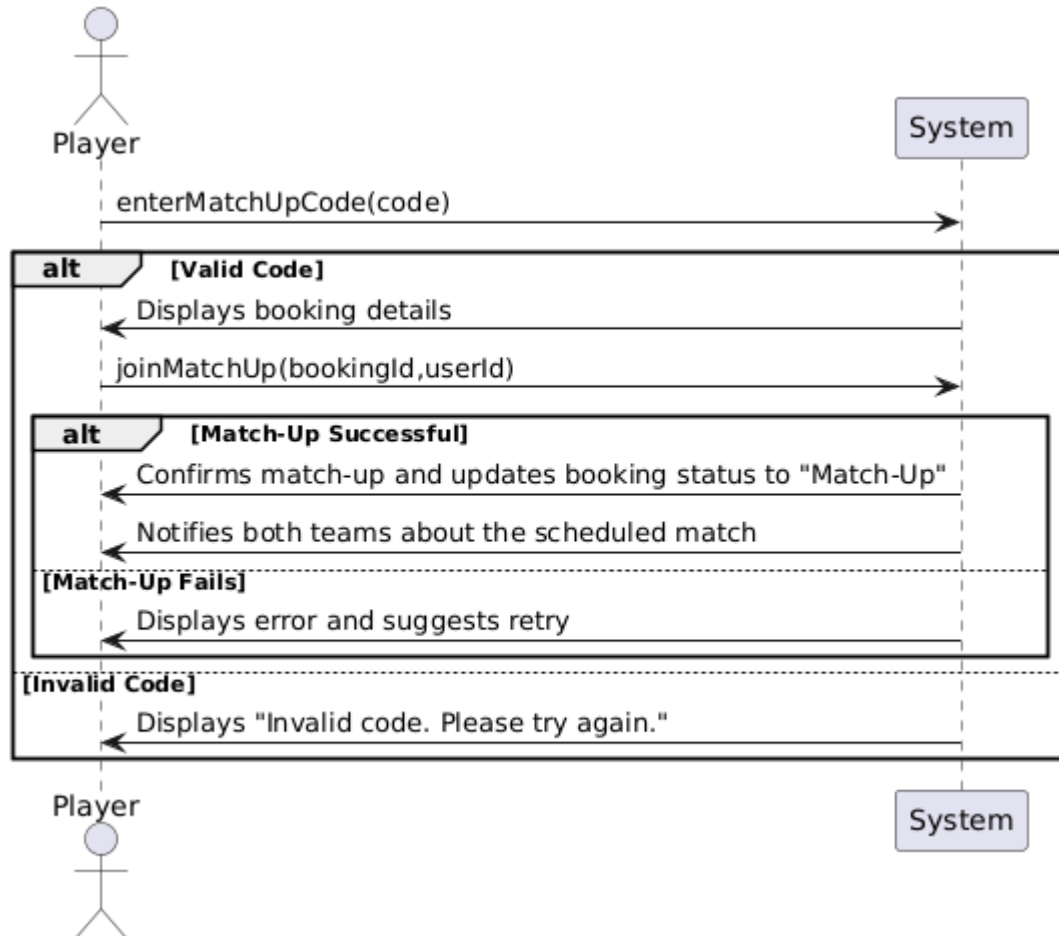
1) Book the Ground



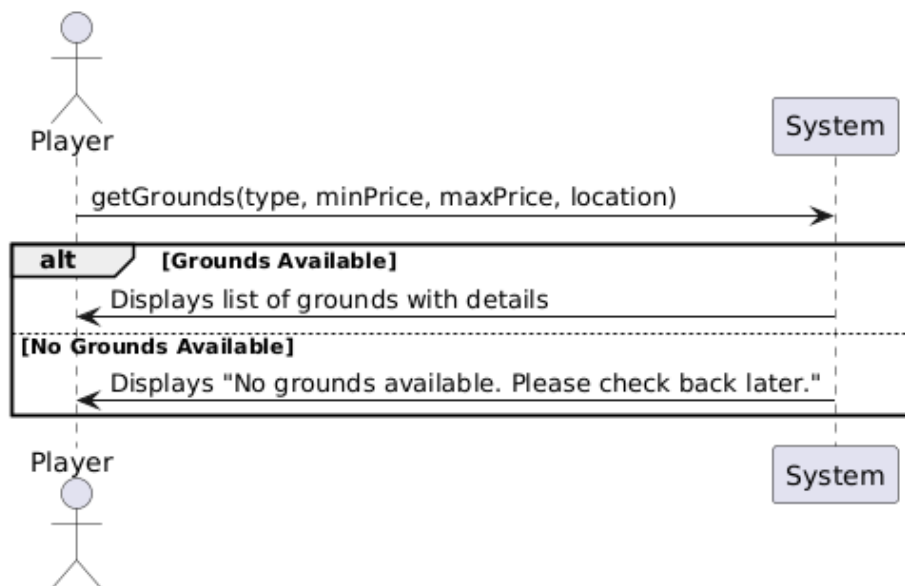
2) Make Payment



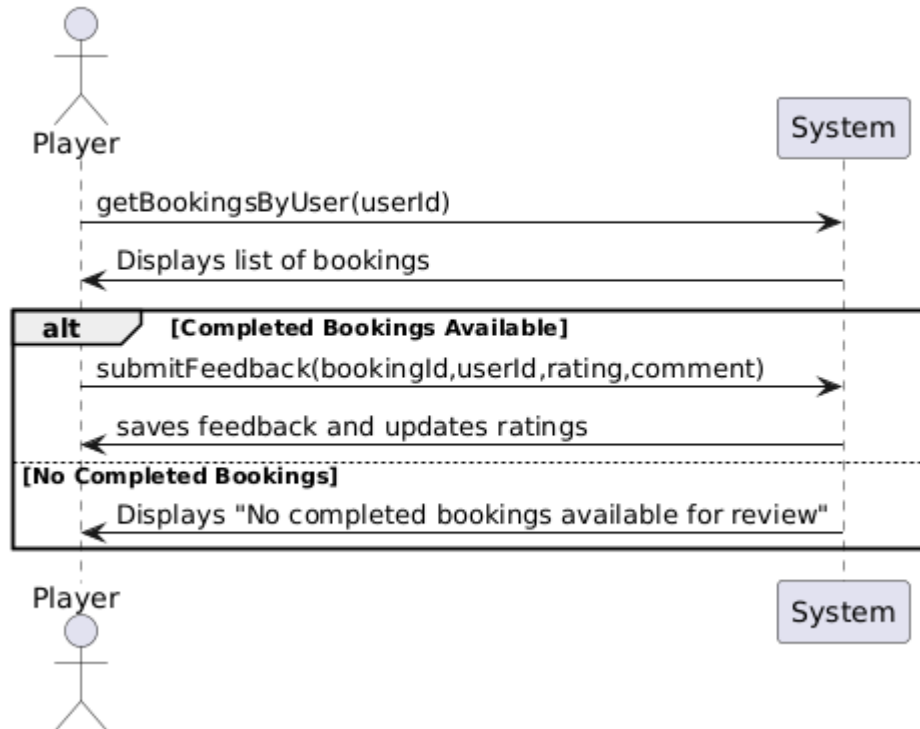
3) Match-Up



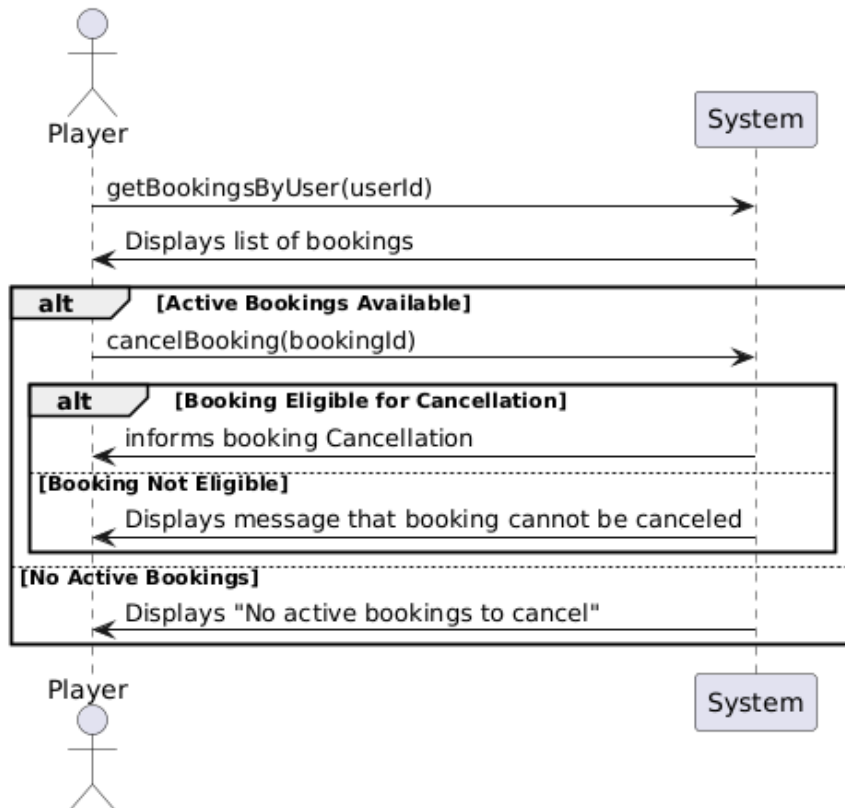
4) View Ground Details



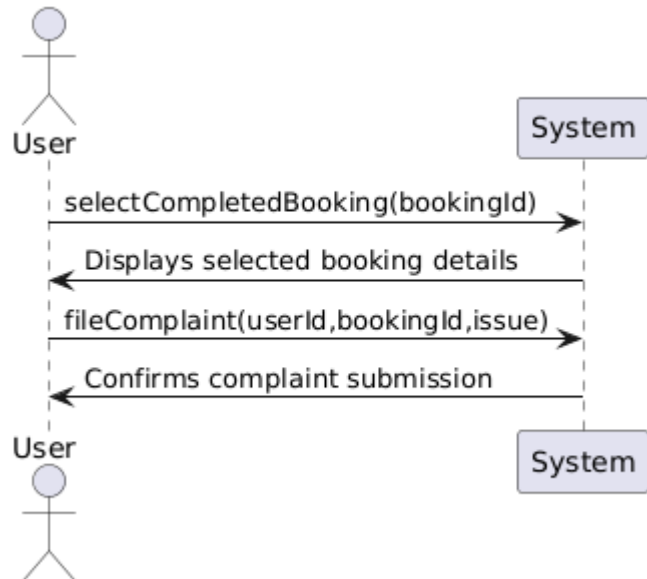
5) Give Feedback



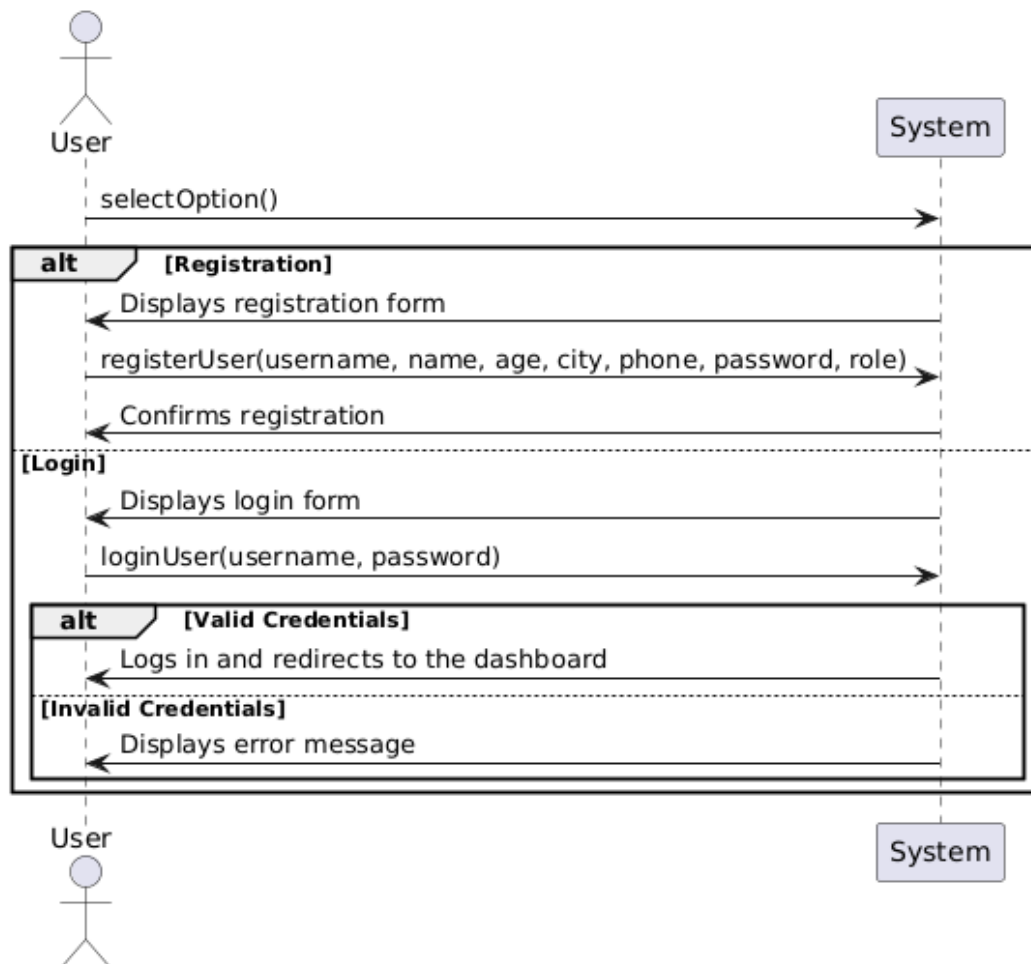
6) Cancel Booking



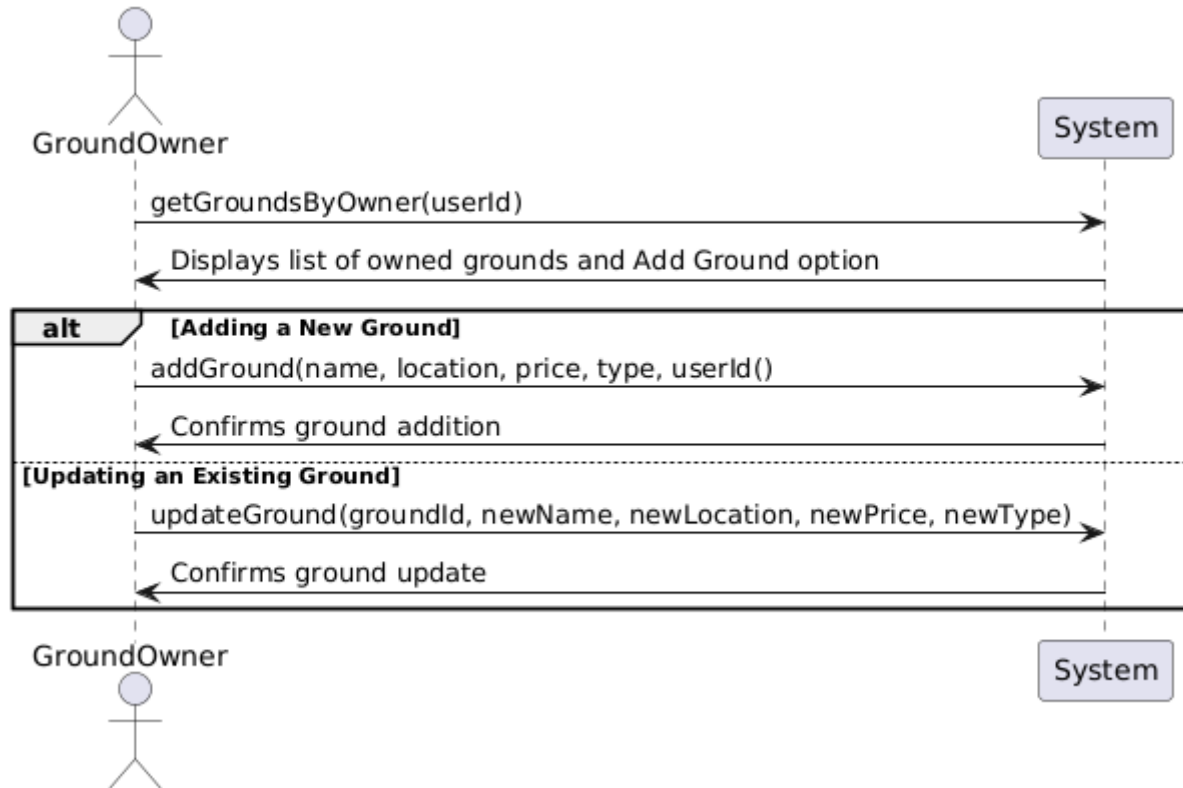
7) File Complaint



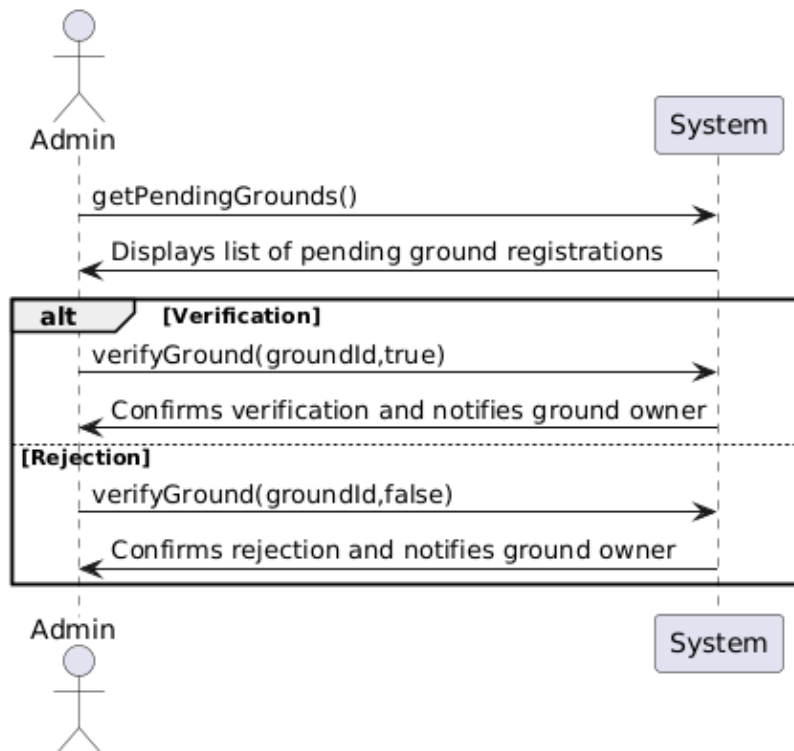
8) Register/Login

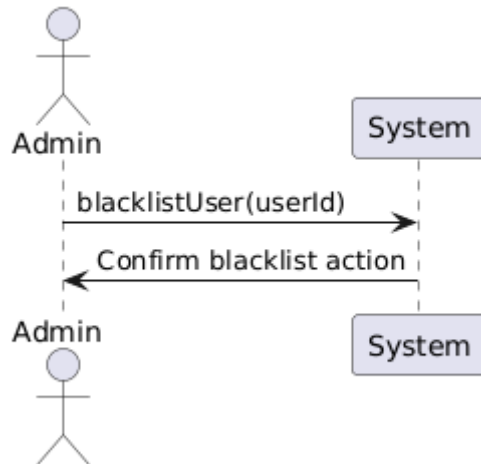
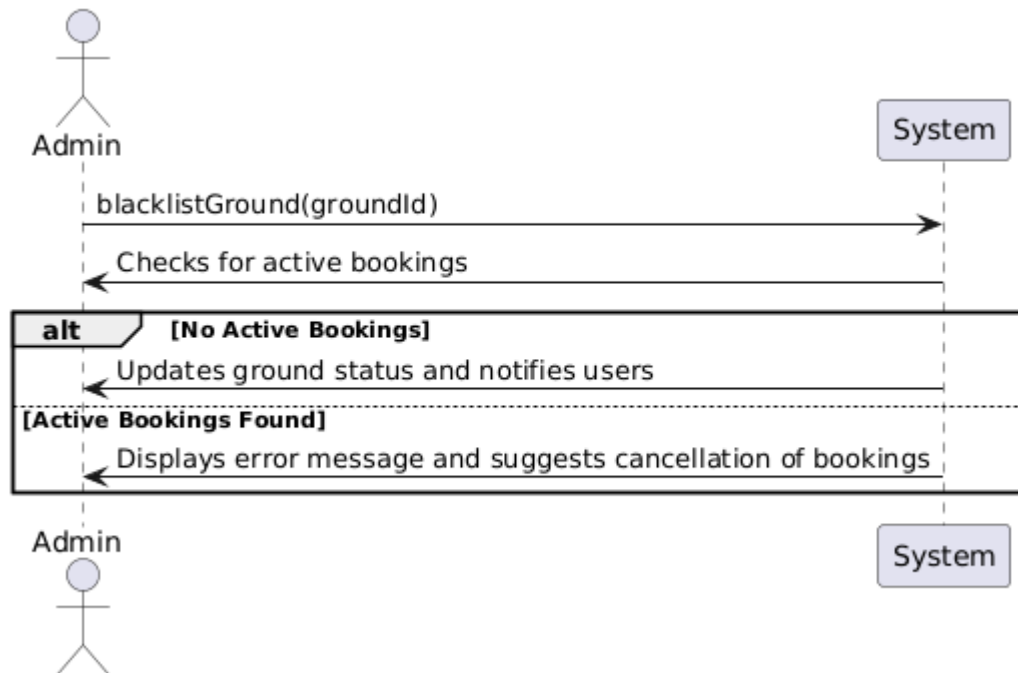


9) Manage Ground

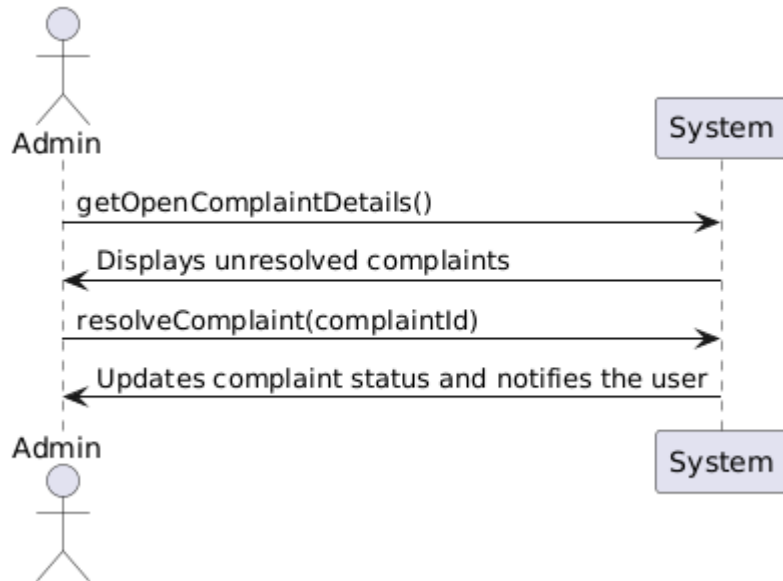


10) Verify Ground



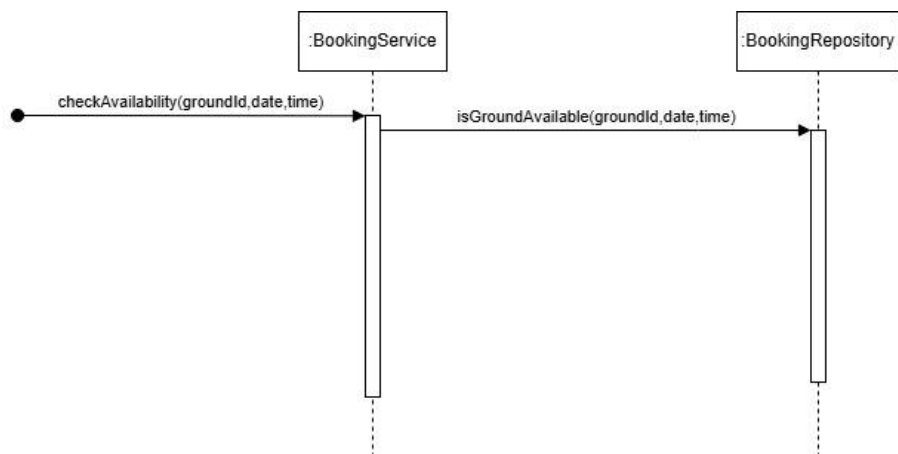
11) Blacklist User**12) Blacklist Ground**

13) Resolve Complaint

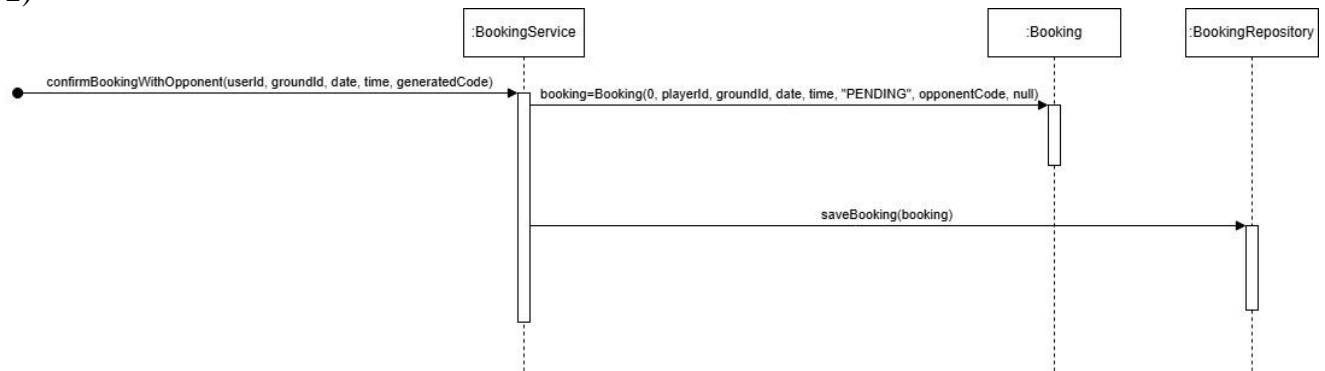


6. Sequence Diagram

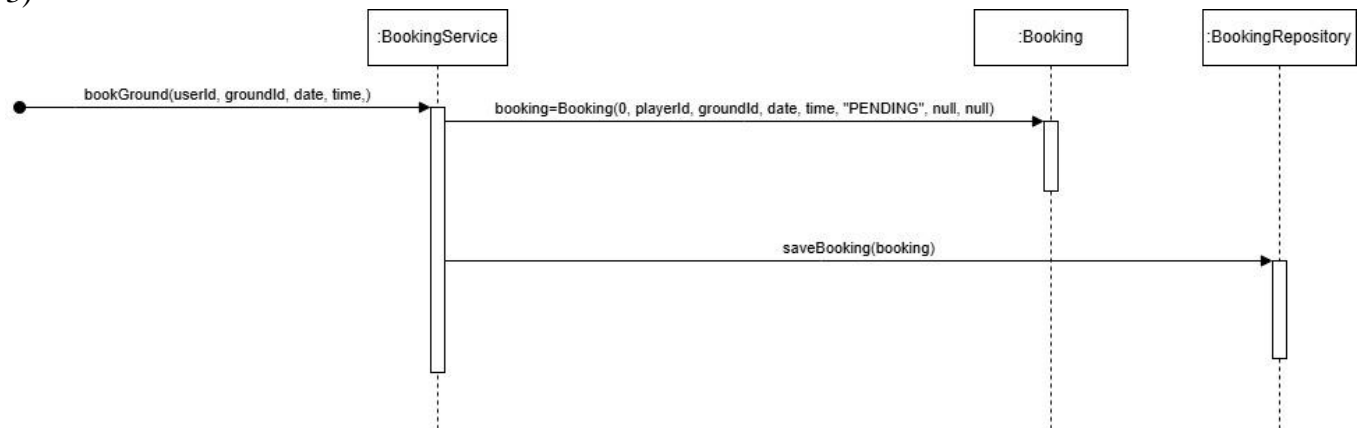
1)



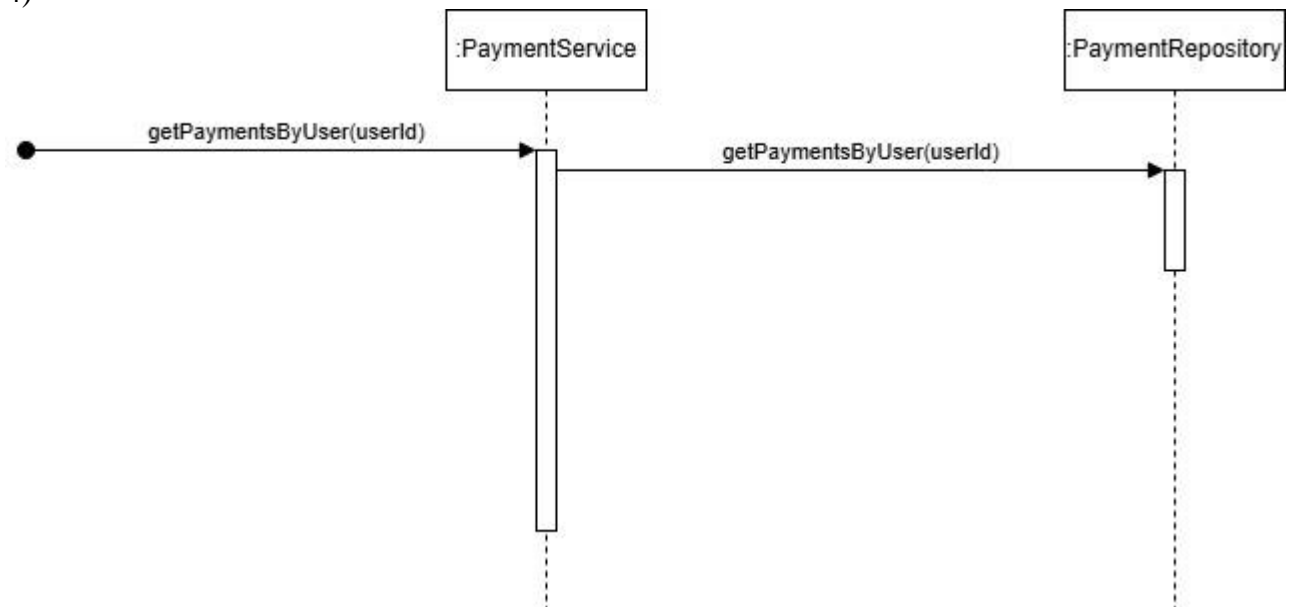
2)



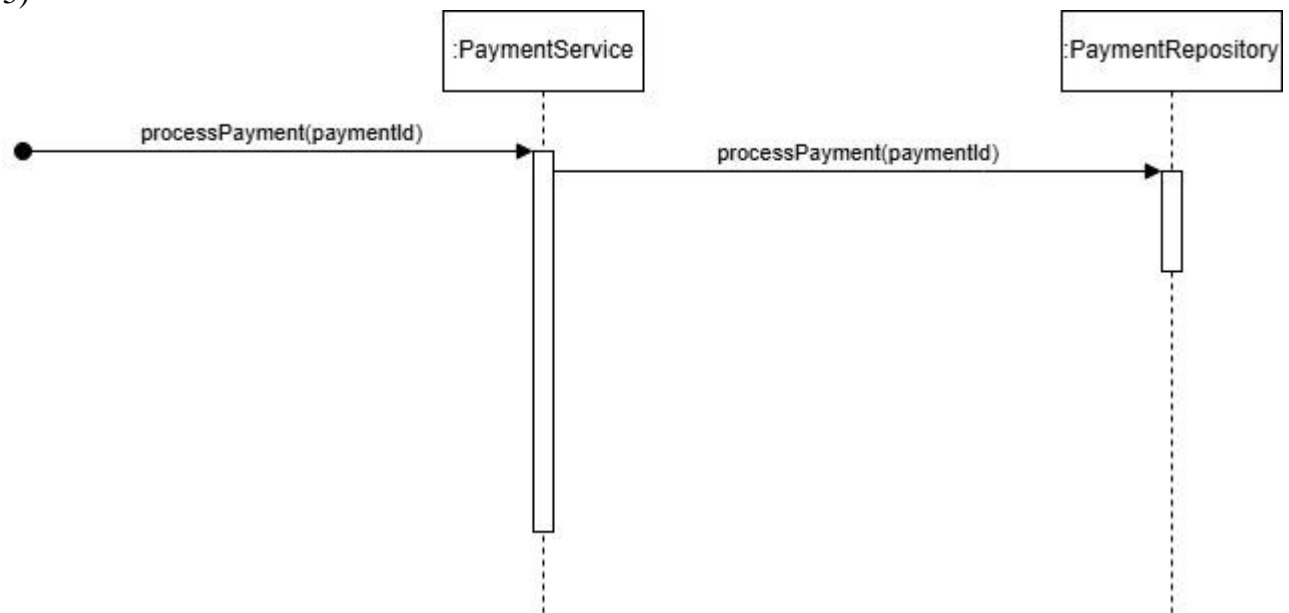
3)



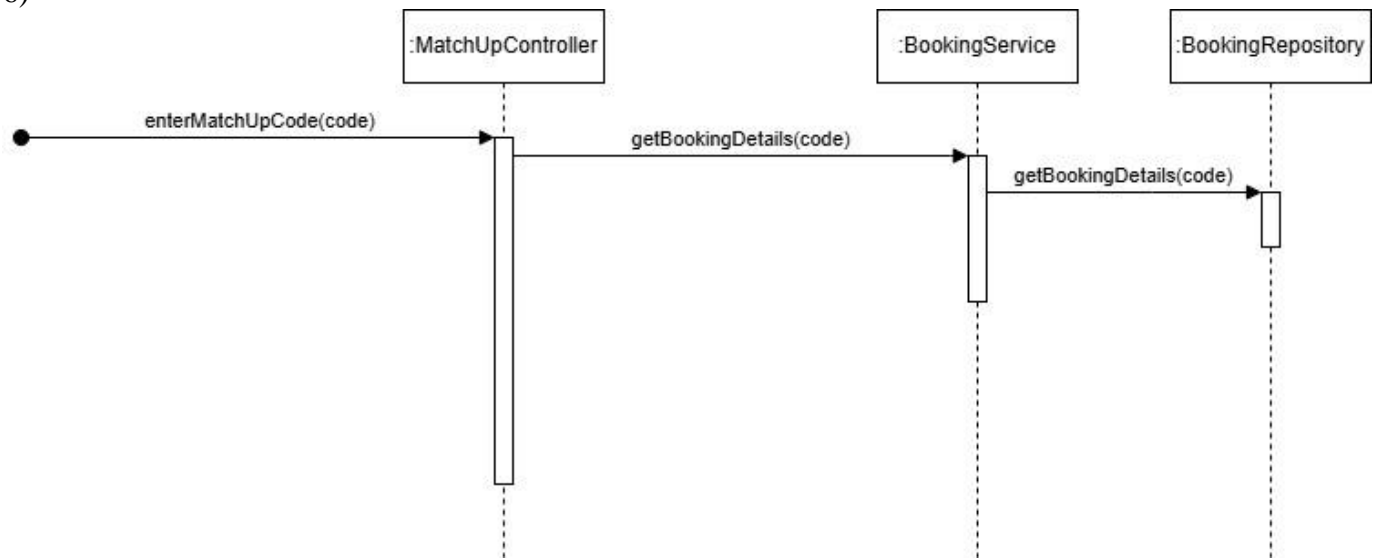
4)



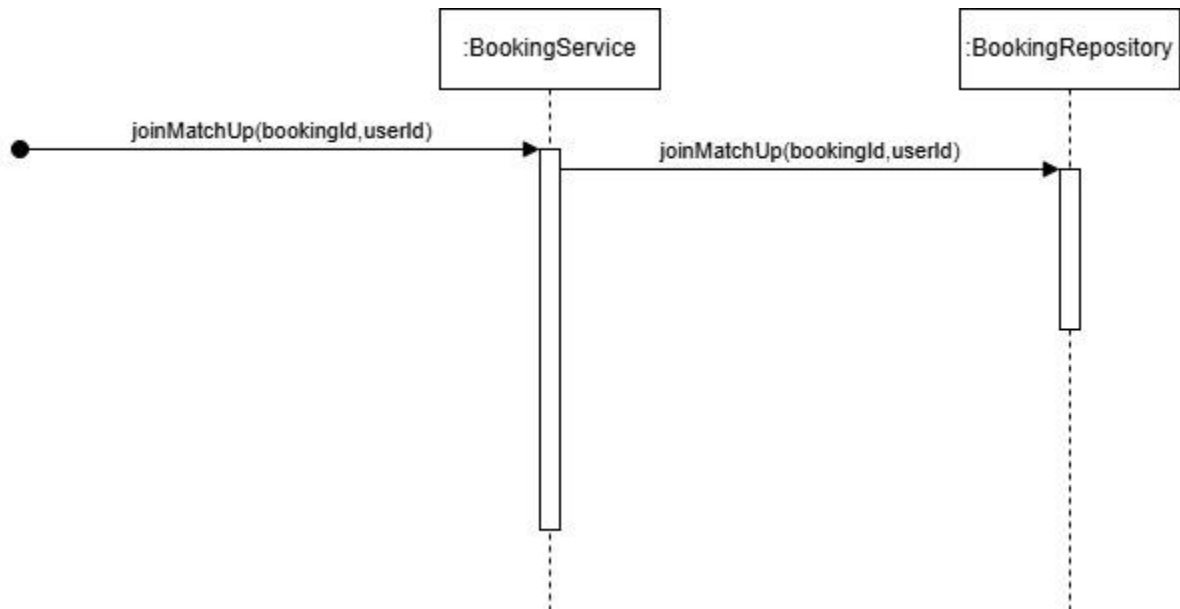
5)



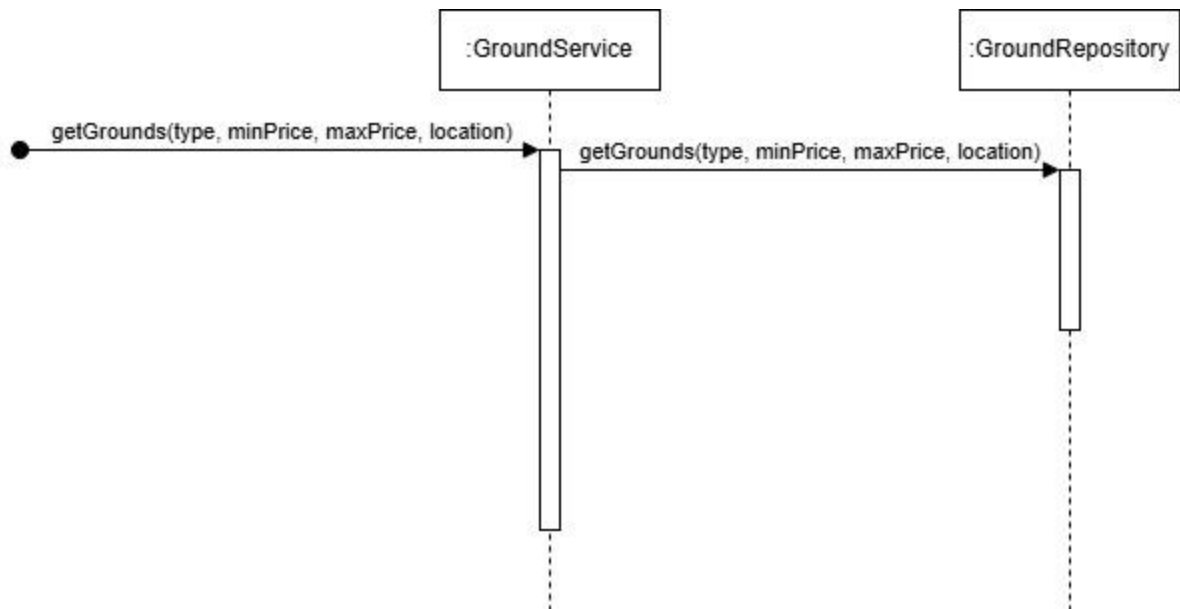
6)



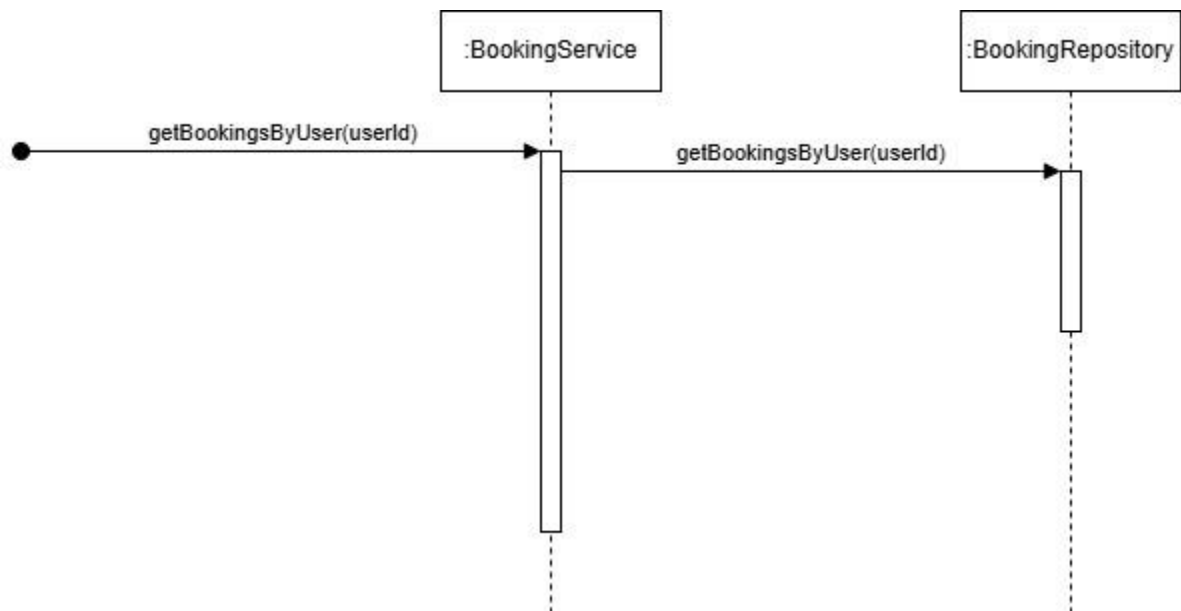
7)



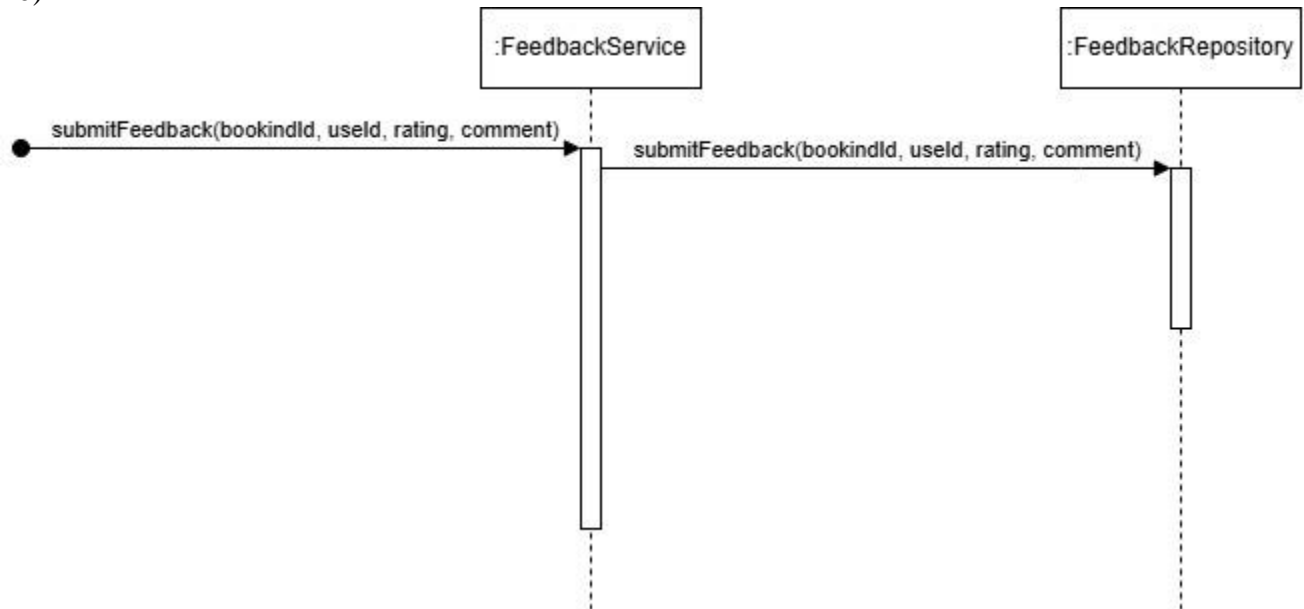
8)



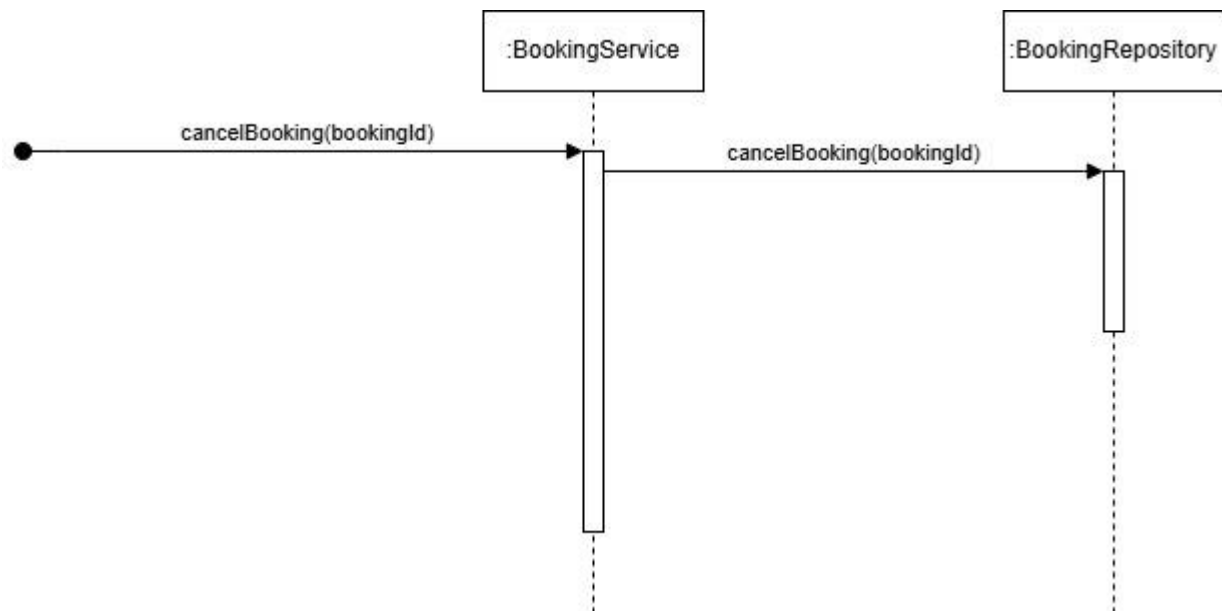
9)



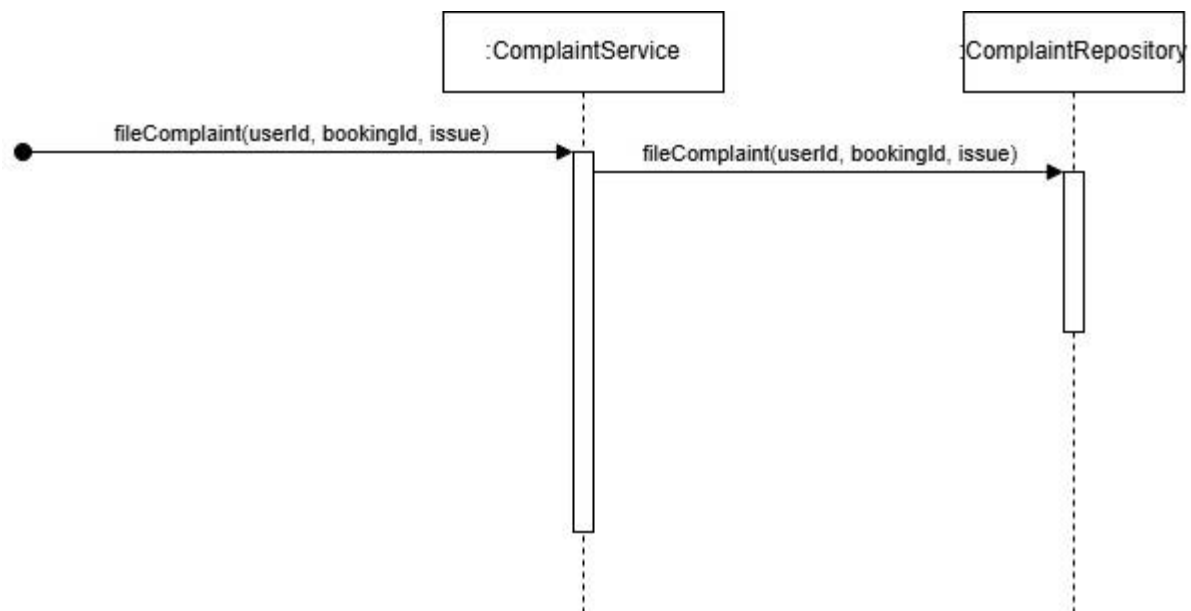
10)



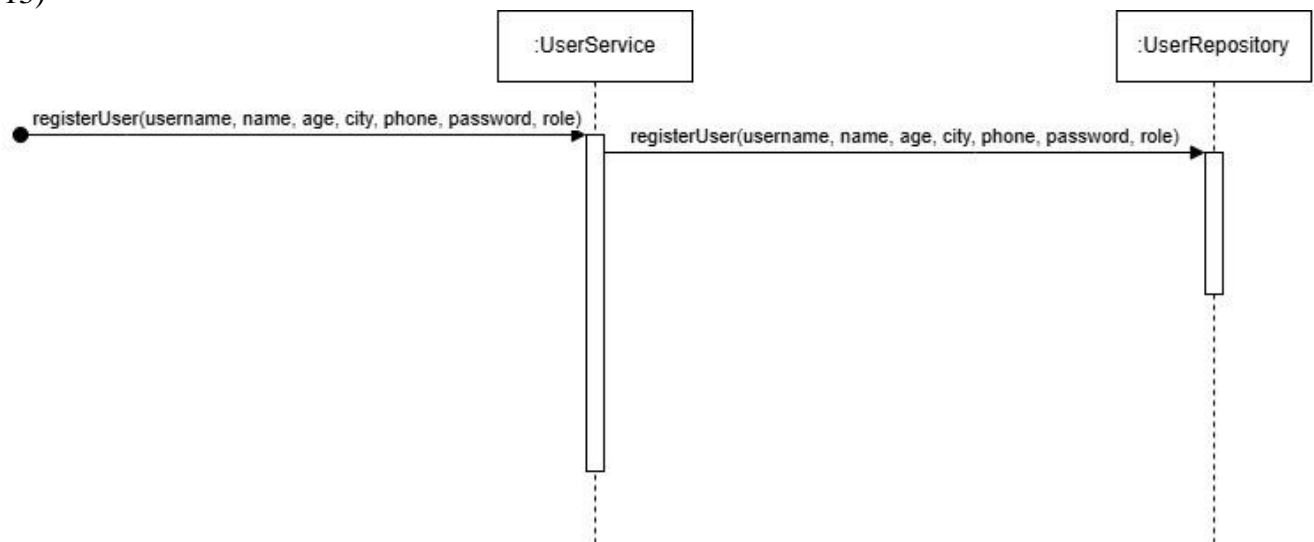
11)



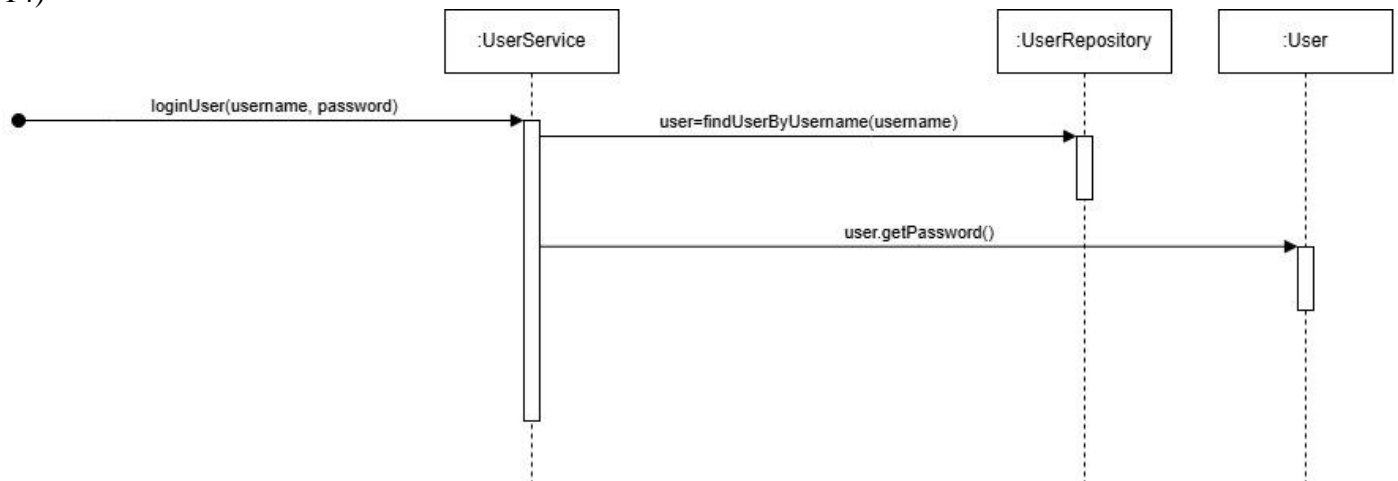
12)



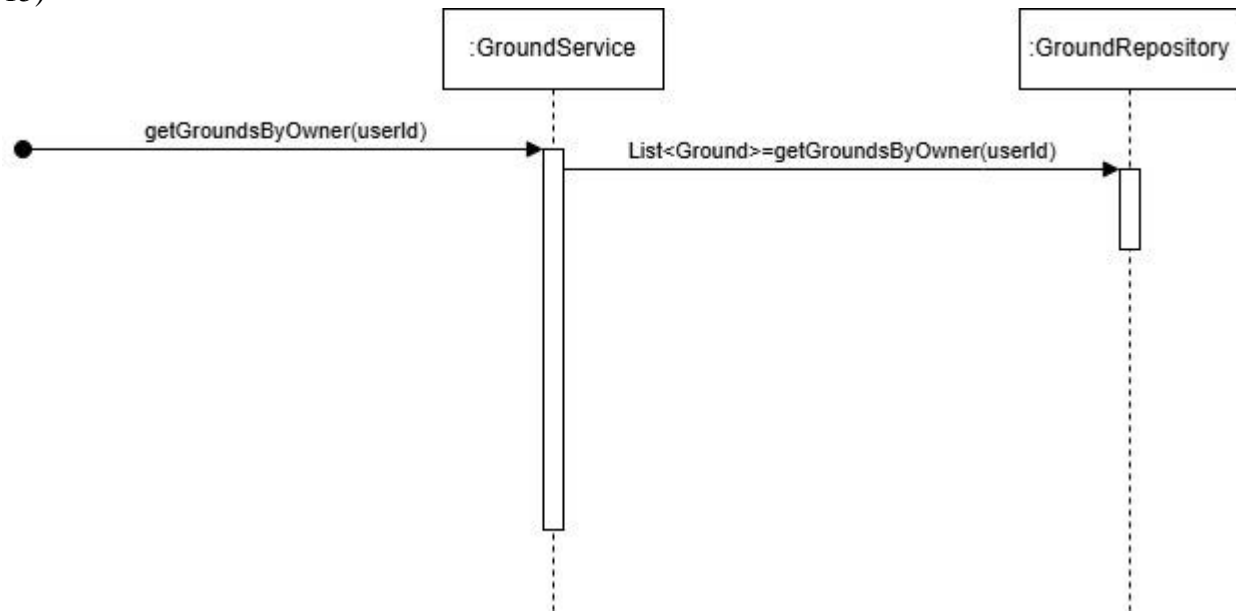
13)



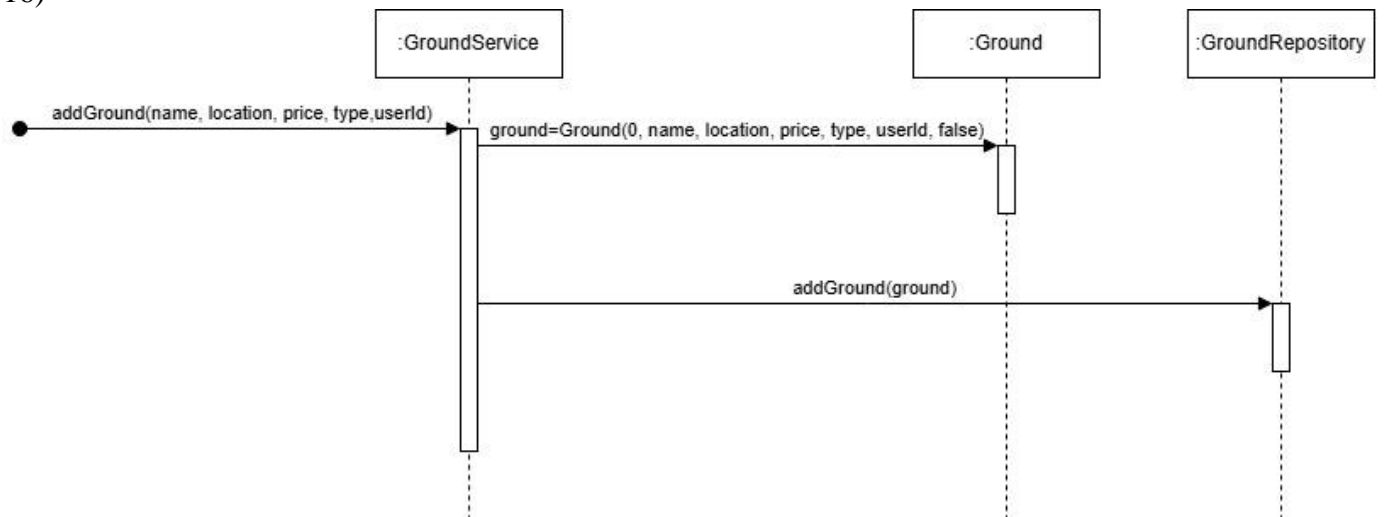
14)



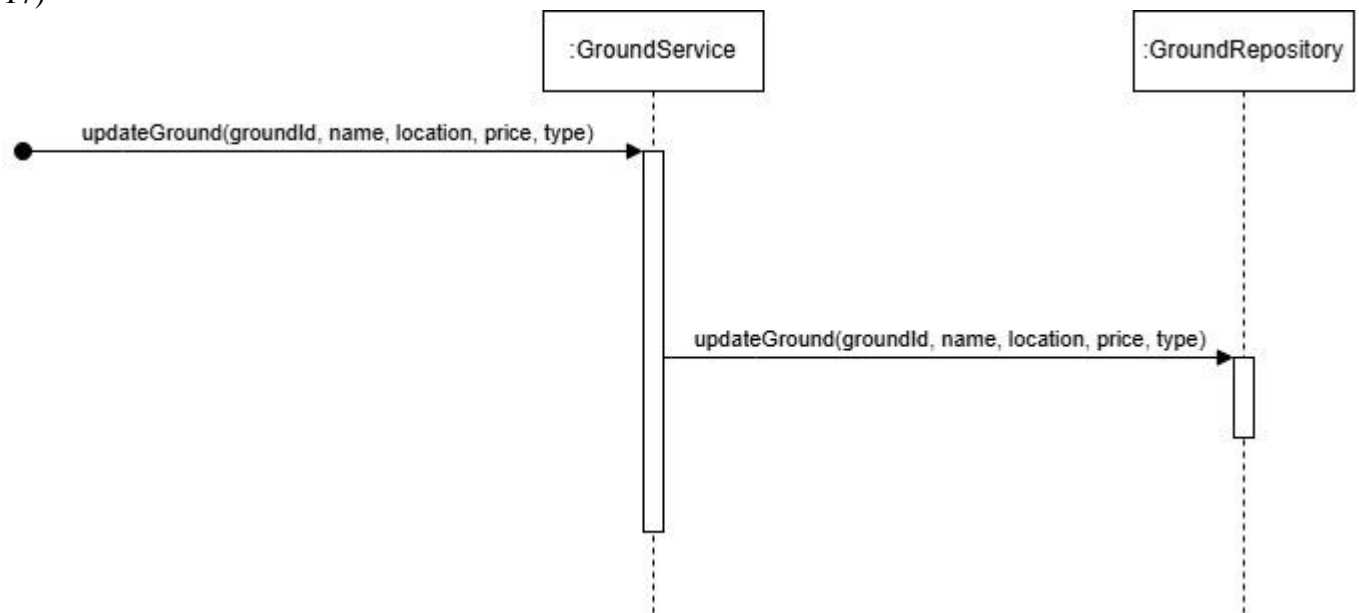
15)



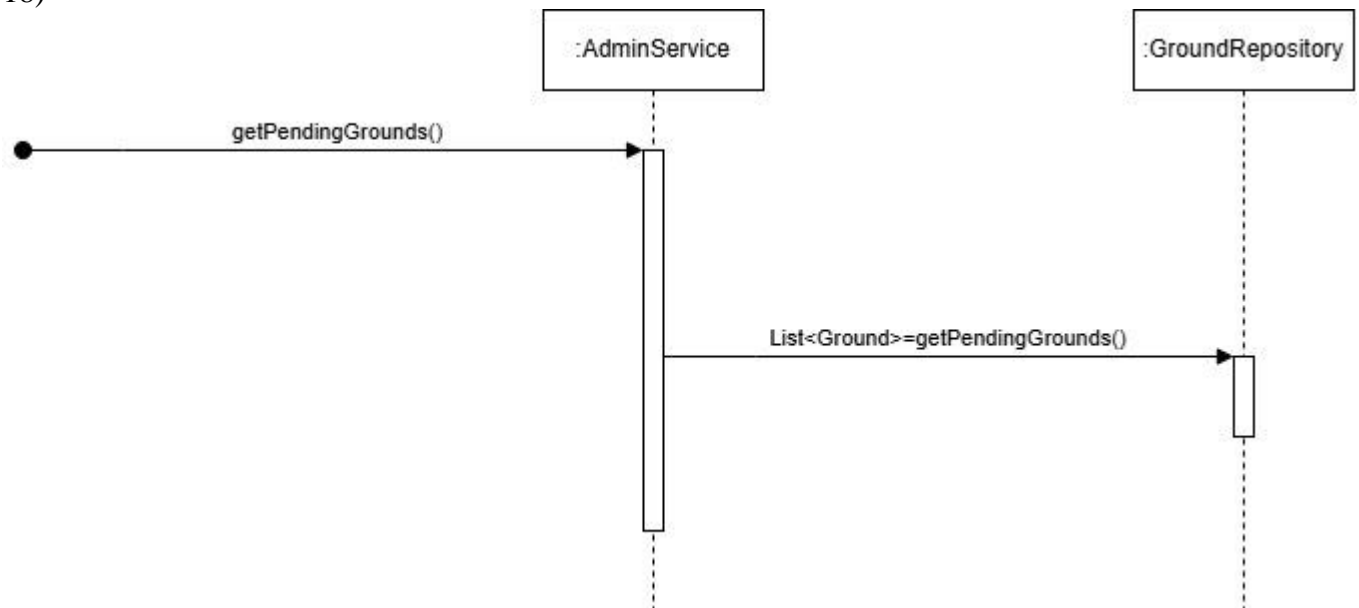
16)



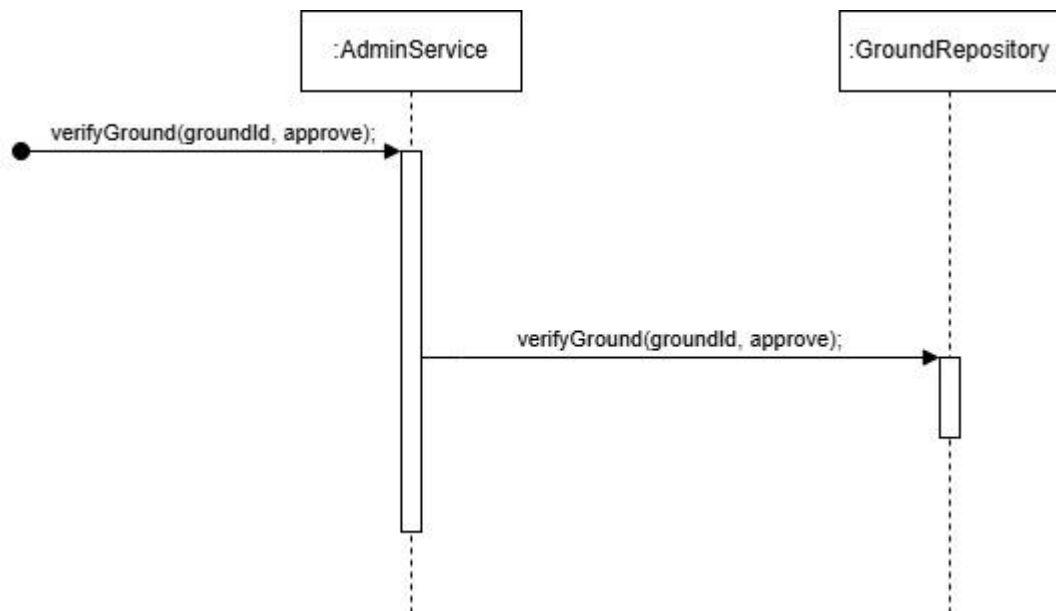
17)



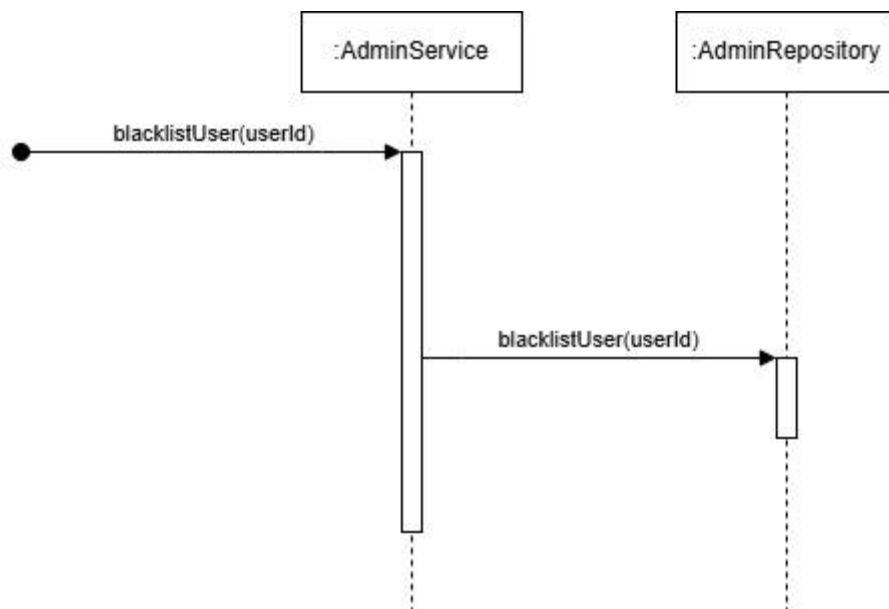
18)



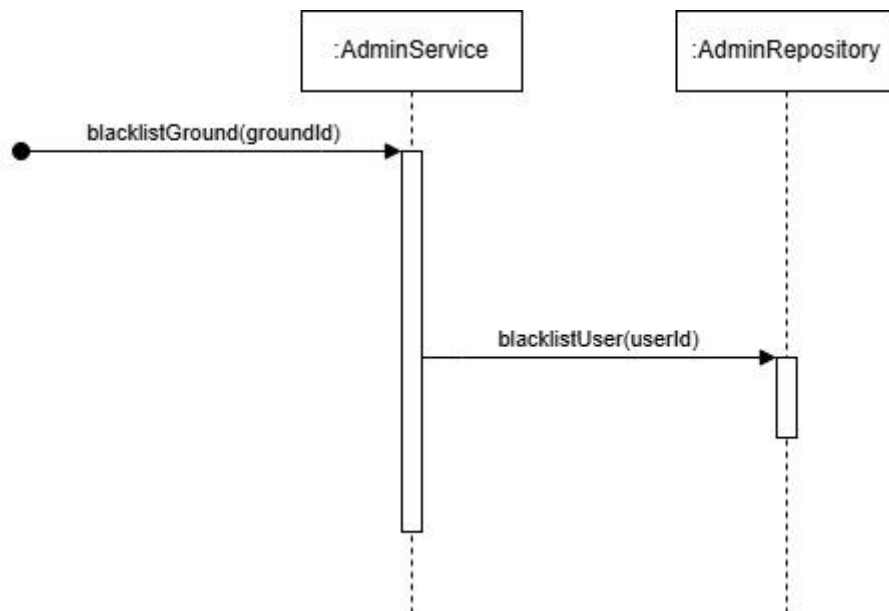
19)



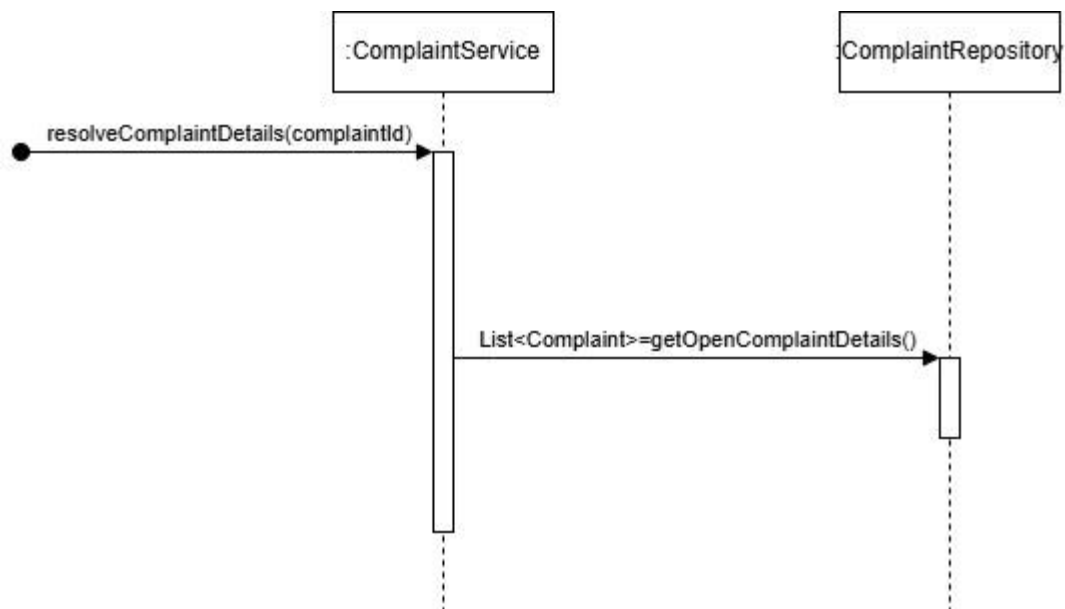
20)



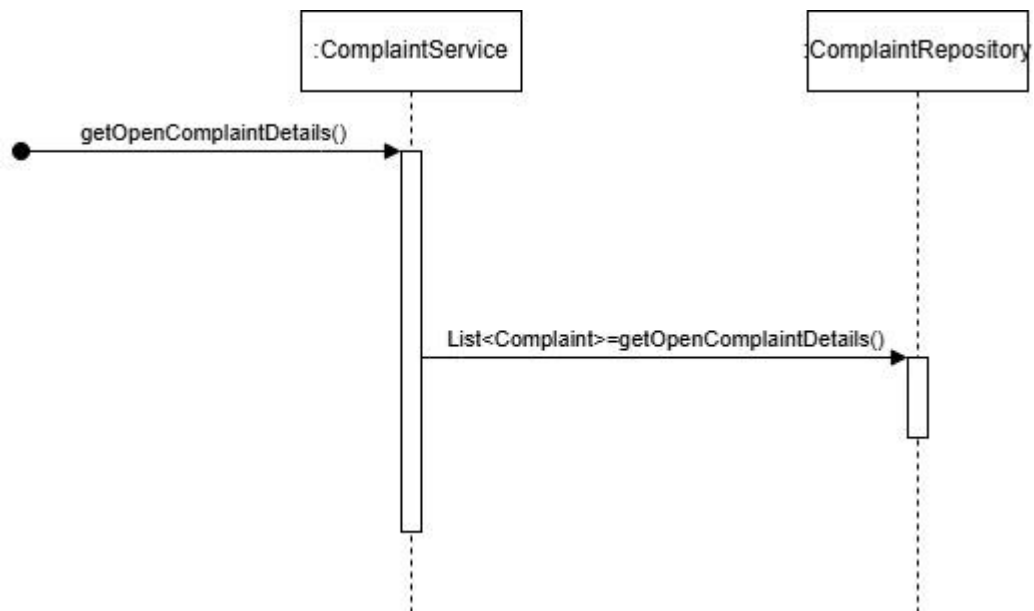
21)



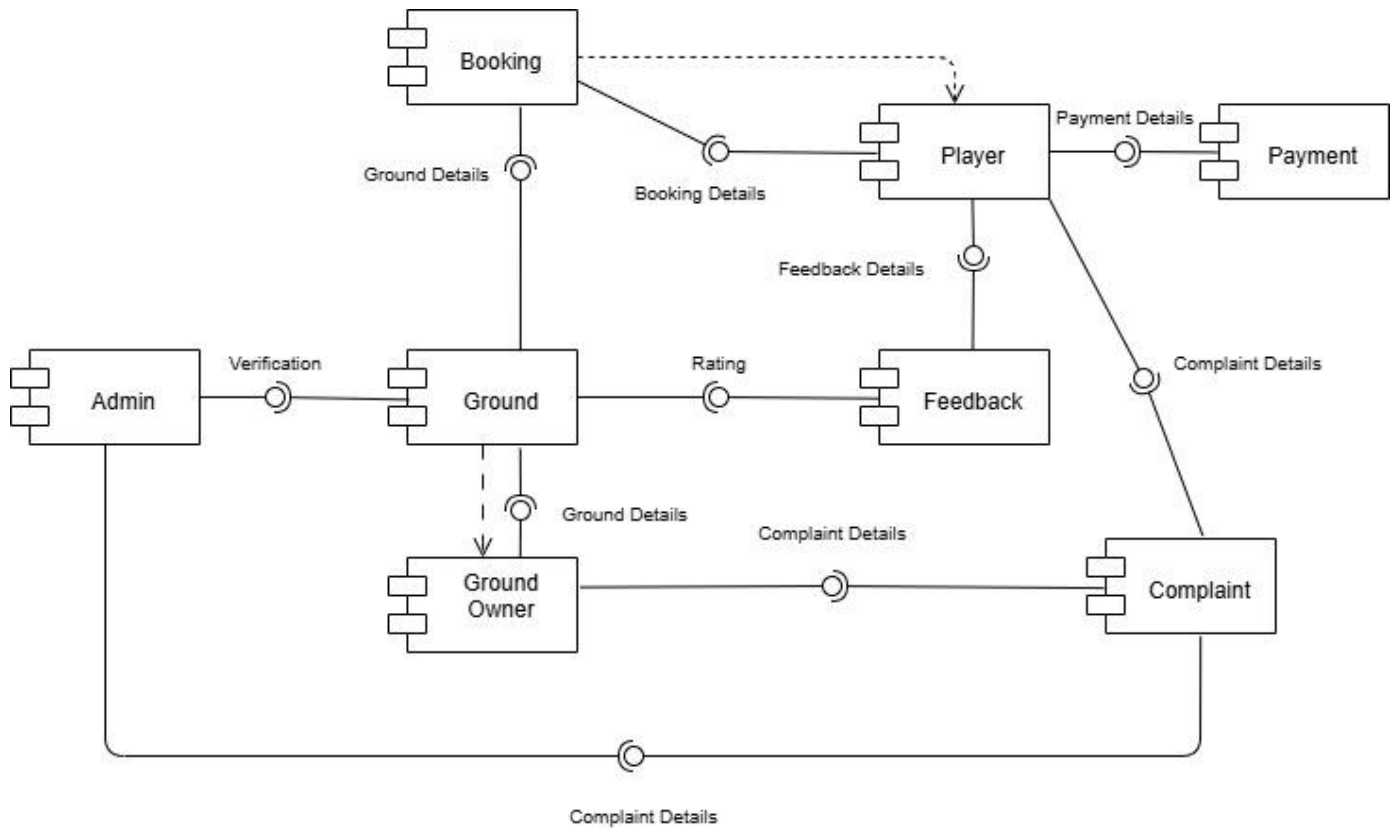
22)



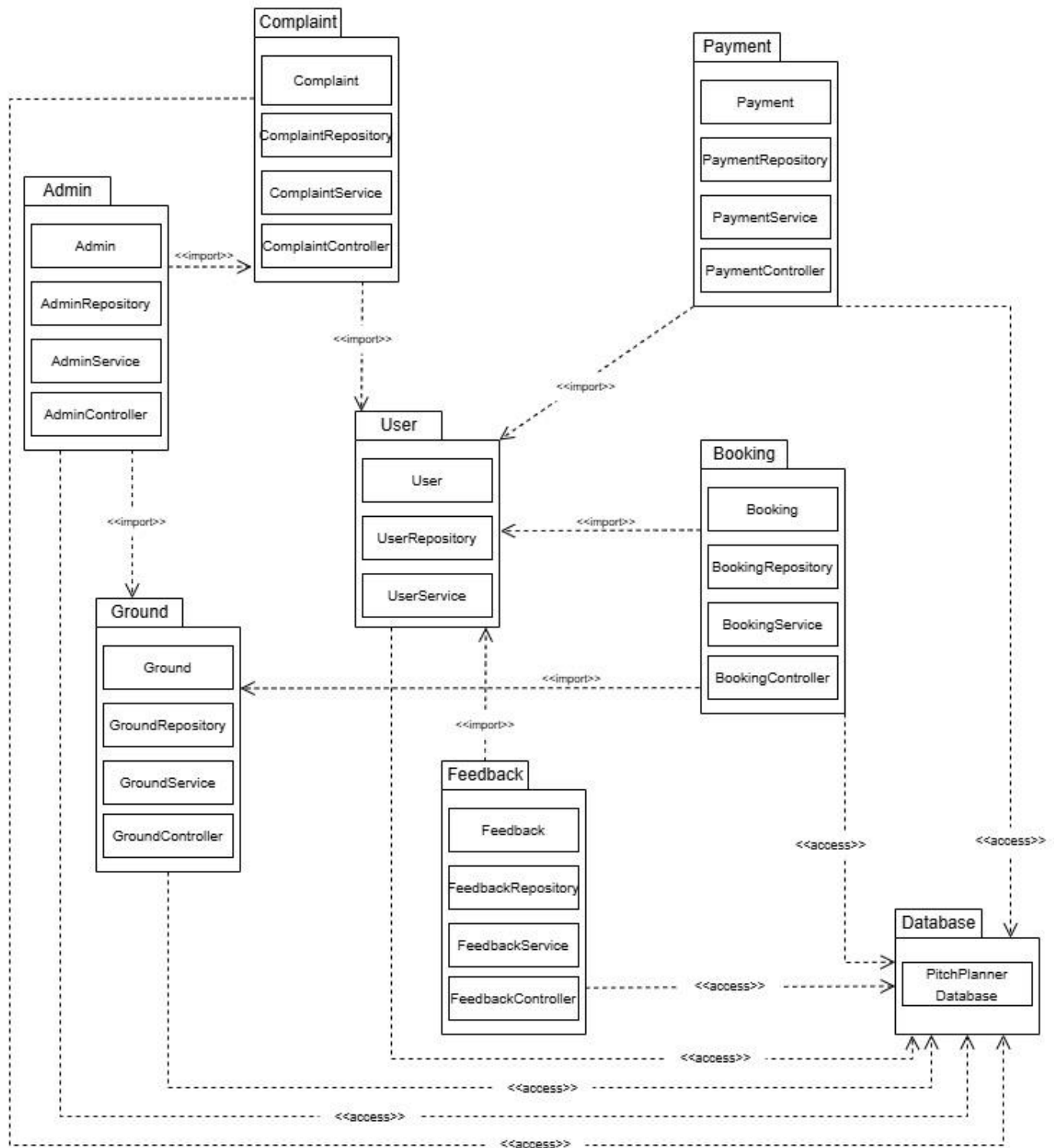
23)



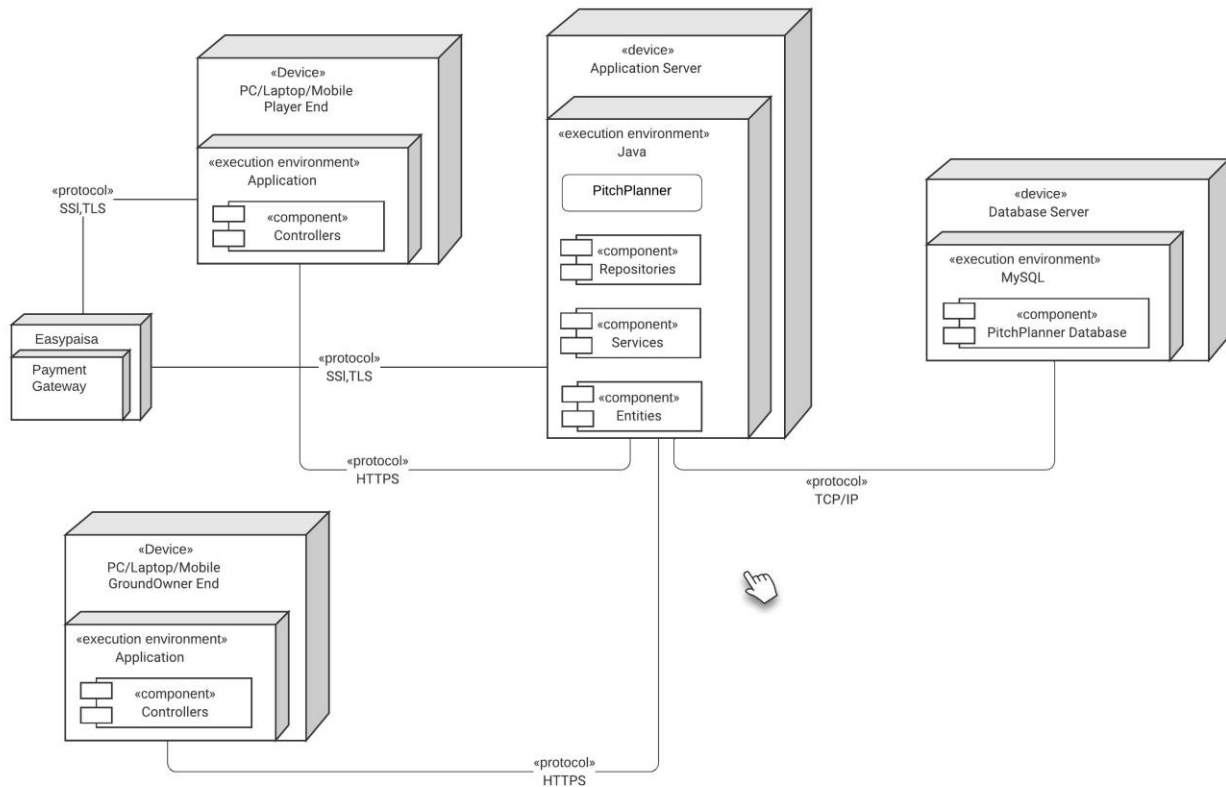
7. Component Diagram



8. Package Diagram



9. Deployment Diagram



10. Class Diagram

https://lucid.app/lucidchart/6a980ef9-4a52-45bd-b309-c0eb1c017048/edit?viewport_loc=-2604%2C-930%2C5527%2C2229%2C0_0&invitationId=inv_5d0ec214-5c34-4942-8d5b-fb936edff3fd

