

**CAPSTONE PROJECT REPORT**

**Report 3 – Software Requirement Specification**

– Hanoi, April 2025 –

**Table of Contents**

[I. Record of Changes 3](#_heading=h.z37isa77hq02)

[II. Software Requirement Specification 4](#_heading=h.pvdgw5k7xtep)

[1. Overall Requirements 4](#_heading=h.bcl76a4d5er8)

[1.1 Context Diagram 4](#_heading=h.2esdi22hs49j)

[1.2 Main Workflows 6](#_heading=h.7pmnns5ok8eo)

[1.3 User Requirements 6](#_heading=h.7ukgtvbo6itb)

[1.3.2 Use Case Diagrams 18](#_heading=h.xhiqcbce0iug)

[1.3.2.1 UseCase for Pickleball Court Management System 18](#_heading=h.bslvcqqmkifn)

[1.4 System Functionalities 22](#_heading=h.h6nga4phbvdz)

[2. Use Case Specifications 25](#_heading=h.sktorm80mi1o)

[2.1 Common Feature 25](#_heading=h.vaf5jzwq1i0n)

[2.2 User Management 29](#_heading=h.9f3knrc8y68o)

[2.3 Booking Management 39](#_heading=h.nhhys120mofi)

[2.4 Court Management 63](#_heading=h.phx5s5yn2shb)

[2.5 Service Management 86](#_heading=h.cygqiokm7mrc)

[2.6 Staff Management 93](#_heading=h.wfffny14n9nm)

[2.7 Payment System 97](#_heading=h.z8ykioadzhwc)

[2.8 Notification Management 100](#_heading=h.2i31xz5nv0he)

[2.9 Financial Management 101](#_heading=h.vhl1be7bsjqs)

[3. Functional Requirements 106](#_heading=h.hg9y0xchunul)

[3.3 System Owneristration 107](#_heading=h.oclar3jrrhok)

[4. Non-Functional Requirements 143](#_heading=h.8pms216zlptj)

[4.1 External Interfaces 143](#_heading=h.6gyqnyh3ivhn)

[4.2 Quality Attributes 144](#_heading=h.va1wy6ajqkz6)

[5. Requirement Appendix 146](#_heading=h.hb3280w06ujp)

[5.1 Common Requirements 146](#_heading=h.qxo52eqb0m7i)

[5.2 Business Rules 147](#_heading=h.qft4pqqmwthz)

[5.3 System Messages 147](#_heading=h.2guthmna3g3b)

[5.4 Other Requirements… 148](#_heading=h.dgizln8g0vlc)

# I. Record of Changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Software Requirement Specification

## 1. Overall Requirements

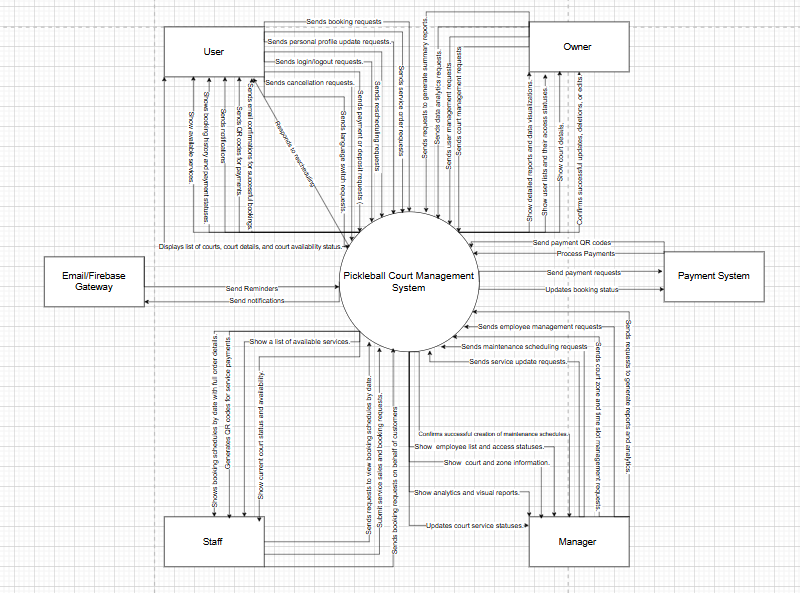
### 1.1 Context Diagram

The **Pickleball Court Management System** is a digital platform designed to facilitate court reservations and equipment rentals. It connects users, guests, owners, staff, and managers to ensure a seamless booking experience and efficient court operations.

The system's main features include real-time court booking, dynamic pricing management, intelligent scheduling suggestions, and automated notifications via the system and email. Users can book courts, rent equipment, and receive booking updates directly through the platform or email notifications.

Owners and staff manage court bookings, monitor user memberships, configure court availability, and handle maintenance schedules. The platform supports integrated payment processing, automatic deposit handling, and streamlined refund management to improve transaction transparency.

By leveraging automation and structured workflows, the system aims to optimize court utilization, enhance user convenience, and provide a complete pickleball playing and management experience without relying on SMS-based communication.

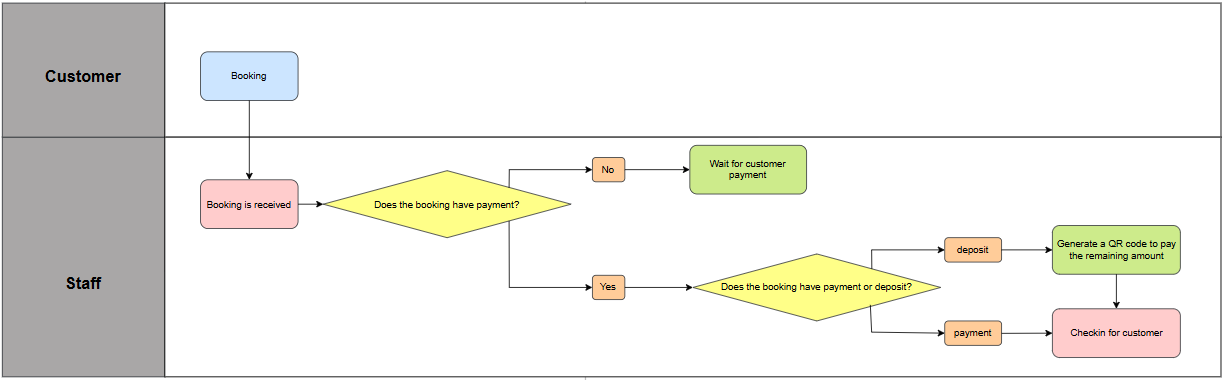
The context diagram below illustrates the system architecture for version 1.0.  


***Figure 4.1****: Context Diagram*

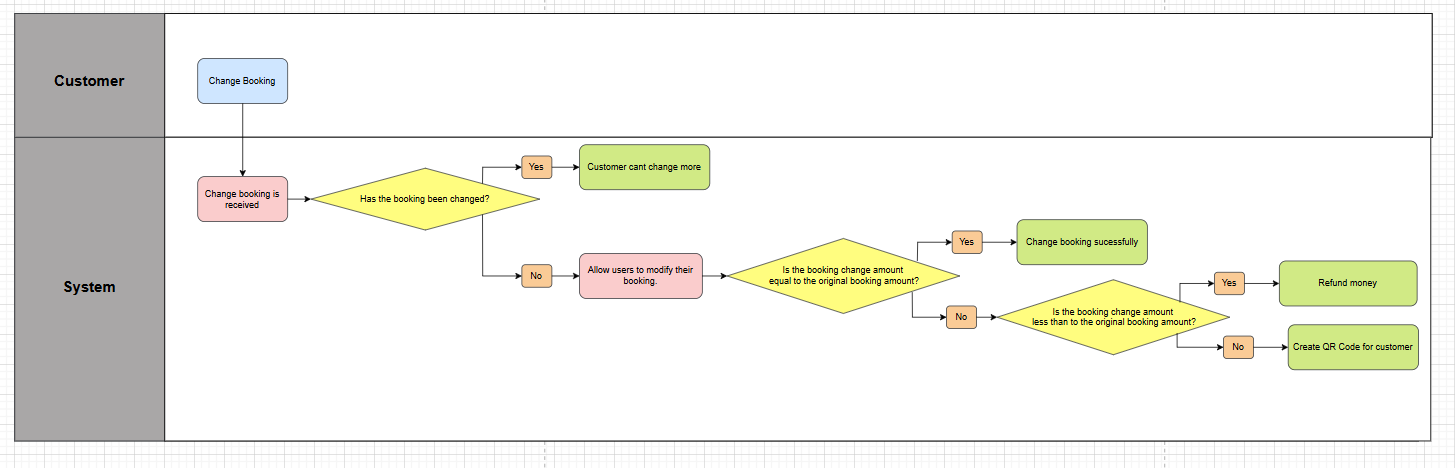
### 1.2 Main Workflows

#### 1.2.1. Booking Processing

#### 1.2.2. Received Booking Processing



#### 1.2.3. Change Booking Processing



#### 

### 1.3 User Requirements

#### 1.3.1 Actors

**Actors:** Guest, User, Owner, Manager, Notification System, Payment System, Staff

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | User | A registered and logged-in individual who can fully interact with the system. Users can book courts and services, manage personal profiles, track their order history, and make payments. |
| 2 | Guest | A visitor who has not logged in or registered. Guests can browse available courts and services, view details, and make bookings. However, some features are limited until they register or log in. |
| 3 | Owner | The system Owner with full access to all features. Owners manage the entire platform, user permissions, and ensure system integrity. |
| 4 | Staff | An internal team member responsible for assisting customer bookings, handling payments, and monitoring court availability. Staff also support service sales and perform real-time scheduling. |
| 5 | Manager | A management-level user who oversees staff, operations, and service performance. The manager handles facility configurations, reporting, and maintenance scheduling. |
| 6 | Payment System | The module responsible for generating QR codes linked to each booking or service order. It facilitates user payments by providing secure QR codes but does not directly process or hold funds. Once a user completes payment via external banking apps (e.g., MB Bank,TP Bank, Viettel Money, ...), the Payment System verifies the transaction status, updates the associated order in the system, and triggers confirmation notifications. |
| 7 | Notification System | The module responsible for sending real-time notifications to users, guests, staff, managers, and owners. Notifications are triggered by key events such as booking confirmations, payment completions, cancellations, maintenance updates, and promotional messages. Notifications are delivered via in-app alerts, emails, depending on the user’s configured preferences. |

***Table 4.1****: Actors*

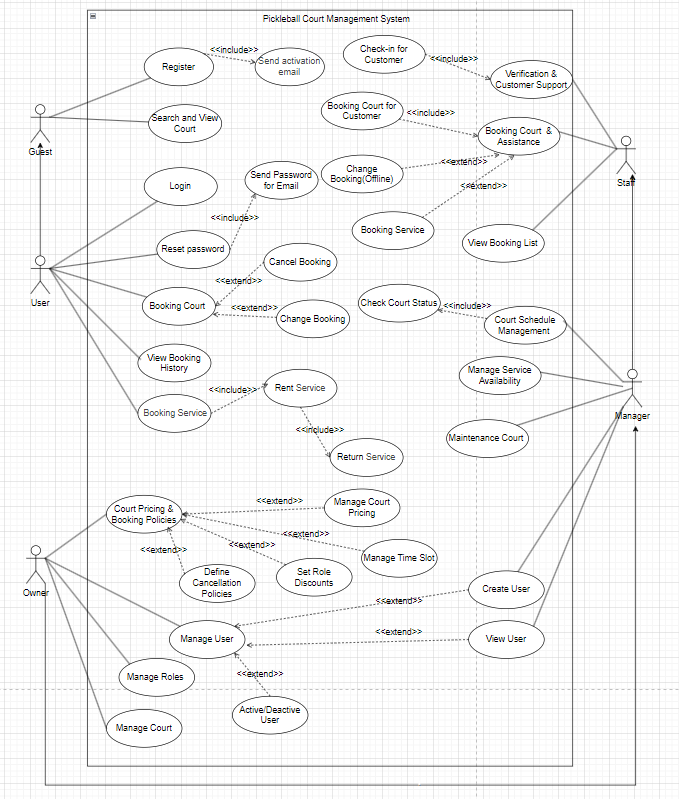
#### 1.3.2 Use Cases (UC)

| **ID** | **Use Case** | **Feature** | **Use Case Description** |
| --- | --- | --- | --- |
| 01 | Register New User | Common | Allows new users to create an account. |
| 02 | Register Student Account | Common | Students can register an account by providing required information (e.g, email(@edu.vn). The system verifies their student status before account creation is completed. |
| 03 | Login | Common | This use case allows users to log in and access the system. The system will authenticate and authorize the user. |
| 04 | Logout | Common | This use case allows users to securely terminate their current session and log out of the system. |
| 05 | Forgot Password | Common | This use case allows a user who has forgotten their password to reset their password. The system generates a new random string of 8 characters (e.g. uppercase, numeric). The new password is sent to the user's registered email address and the user must use the new password to log in. |
| 06 | Update User Profiles | User Management | This use case allows users to update personal information (e.g, image, birthday, first name, last name) |
| 07 | Update Role User by Owner | Role Management | Owner updates the user's role(Manager, Staff, User, Student ) |
| 08 | Update Role User by Manager | Role Management | Manager updates the user's role (Staff, User, Student ) |
| 09 | Create Role | Role Management | Owner creates the user's role. |
| 10 | Delete Role | Role Management | Owner deleted the user's role. |
| 11 | View Role | Role Management | Owner views the user's role. |
| 12 | View List of Users | User Account Management | Displays all registered users with filtering options. |
| 13 | Create New User by Owner | User Account Management | Allows the Owner to create a new user account, assign a role (Manager, Staff, User, Student), and set the account status (active/inactive). |
| 14 | Create New User by Manager | User Account Management | Allows the Manager to create a new user account, assign a role( Staff, User, Student), and set the account status (active/inactive). |
| 15 | Deactivate User | User Account Management | Allows Owner/Manager to deactivate accounts. |
| 16 | Activate User | User Account Management | Allows Owner/Manager to deactivate accounts. |
| 17 | Booking Service On Mobile | Booking Management | This use case allows User or Guest book service on mobile apps (e.g., drink, snack,...) from the list of available services. |
| 18 | Change Booking | Booking Management | Users can modify their booking date/time or court once (within allowed time frame). |
| 19 | Cancel Booking | Booking Management | Users can cancel bookings, which may result in deposit loss. |
| 20 | View Booking Personal | Booking Management | This use case allows a User to view their complete order history, including court bookings and service purchases. Orders are listed chronologically with key details like order type, amount, status, and timestamps. The user can filter by date, status, or type, and click each entry for detailed viewing. |
| 21 | View Booking Detail Personal | Booking Management | Shows detailed information about a specific order. |
| 22 | Confirm Booking | Booking Management | Allows confirmation of purchase orders before processing. |
| 23 | Deposit Booking | Booking Management | Users deposit to reserve a spot, the system generates a QR code for deposit. |
| 24 | Payment Booking | Booking Management | Users deposit to reserve a spot, the system generates a QR code for payment. |
| 25 | Refund on cancellation | Booking Management | Refund system (minus deposit) when canceling booking |
| 26 | Refund for rescheduling | Booking Management | Refund system for difference (deposit) when changing booking. |
| 27 | Refund without check in | Booking Management | Refund system (minus deposit) if the user does not check-in. |
| 28 | Book Court Daily | Booking Management | Users book courts daily, multiple time slots and days. |
| 29 | Book Court Regular | Booking Management | Users book courses on a recurring basis for multiple days. |
| 30 | Book Court Optimal | Booking Management | The system suggests alternative venues if schedules overlap. |
| 31 | Hold Court While Booking | Booking Management | The system temporarily remains in the same state while the user completes the payment. |
| 32 | View Booking History User by Owner | Booking Management | This use case allows the Owner to view the entire booking history of a user, for example, within a specific time period, by venue, by day, by month, etc. This information helps the owner track the customer's booking behavior and provide better services. |
| 33 | View Booking History User by Manager | Booking Management | This use case allows allowed to view the entire booking history of a user, for example, within a specific time period, by venue, by day, by month, etc. This information helps the owner track the customer's booking behavior and provide better services under the Owner's authority |
| 34 | View Reservation | Booking Management | Users can select time slots by selecting the date. |
| 35 | View Schedule occupancy analysis by Owner | Court Management | Owner views all court usage analysis over time. |
| 36 | View Schedule occupancy analysis by Manager | Court Management | Manager views all court usage analysis over time under the Owner's authority |
| 37 | View Peak hours analysis by Owner | Court Management | Owner views peak hour analysis of all courts. |
| 38 | View Peak hours analysis by Manager | Court Management | Owner views peak hour analysis all courts under the Owner's authority |
| 39 | Create Court | Court Management | Owner adds a new court with details like name, type, location. |
| 40 | Update Court | Court Management | Owner updates existing course information. |
| 41 | Active Court | Court Management | This use case allows Owners to activate a court that was previously set as inactive, making it visible and available for booking by users. This is commonly used after court maintenance or administrative review. |
| 42 | Deactivate Court | Court Management | This use case allows the Owner to deactivate a court, making it unavailable for user bookings. This action is typically taken when the court is under maintenance, unavailable due to events, or retired from regular use. Deactivated courts are not visible to users but remain in the system for editing or future reactivation. |
| 43 | Create Court Slot by Owner | Court Management | This use case allows the Owner to define available time slots for each court, including start/end times, duration, and availability rules (e.g., day of week, custom pricing). Once created, these time slots appear in the booking calendar for users. |
| 44 | Create Court Slot by Manager | Court Management | This use case allows the Manager to define available time slots for each court, including start/end times, duration, and availability rules (e.g., day of week, custom pricing). Once created, these time slots appear in the booking calendar for users under the Owner's authority |
| 45 | Update Court Slot by Owner | Court Management | This use case allows **Owners** to update information related to a specific court slot, including its start time, end time, price, availability status, or applicable days. Updates are reflected immediately and can affect upcoming booking availability. |
| 46 | Update Court Slot by Manager | Court Management | This use case allows **Managers** to update information related to a specific court slot, including its start time, end time, price, availability status, or applicable days. Updates are reflected immediately and can affect upcoming booking availability under the Owner's authority |
| 47 | Active Sub-Court by Owner | Court Management | This use case allows the **Owner** to activate an existing sub-court that was previously set as **inactive**. Once activated, the sub-court becomes available for booking and is visible to users in the booking interface. |
| 48 | Active Sub-Court by Manager | Court Management | This use case allows the **Manager** to activate an existing sub-court that was previously set as **inactive**. Once activated, the sub-court becomes available for booking and is visible to users in the booking interface under the Owner's authority |
| 49 | Deactivate Sub-Court by Owner | Court Management | This use case allows Owners to deactivate a specific sub-court, making it unavailable for booking. This can be done due to maintenance, low demand, or scheduling changes. |
| 50 | Deactivate Sub-Court by Manager | Court Management | This use case allows Managers to deactivate a specific sub-court, making it unavailable for booking. This can be done due to maintenance, low demand, or scheduling changes under the Owner's authority |
| 51 | Create Maintenance by Owner | Court Management | This use case allows **Owners** to schedule maintenance periods for specific courts or court areas. During the maintenance window, the court will be **unavailable for booking**. The maintenance record includes a description, duration, and affected court(s), and ensures proper scheduling and communication to users and staff. |
| 52 | Create Maintenance by Manager | Court Management | This use case allows **Managers** to schedule maintenance periods for specific courts or court areas. During the maintenance window, the court will be **unavailable for booking**. The maintenance record includes a description, duration, and affected court(s), and ensures proper scheduling and communication to users and staff under the Owner's authority |
| 53 | Update Maintenance by Owner | Court Management | This use case enables Owners to modify an existing maintenance record for a court, including changes to time, description, or affected court area. Updating maintenance ensures accuracy and allows for changes due to scheduling conflicts or work delays. |
| 54 | Update Maintenance by Manager | Court Management | This use case enables Managers to modify an existing maintenance record for a court, including changes to time, description, or affected court area. Updating maintenance ensures accuracy and allows for changes due to scheduling conflicts or work delays under the Owner's authority |
| 55 | View Maintenance by Owner | Court Management | This use case allows Owners to view a list of all scheduled and completed maintenance records for courts or court areas. Maintenance entries include time period, status, description, and affected courts. This supports operational visibility and planning. |
| 56 | View Maintenance by Manager | Court Management | This use case allows Managers to view a list of all scheduled and completed maintenance records for courts or court areas. Maintenance entries include time period, status, description, and affected courts. This supports operational visibility and planning under the Owner's authority |
| 57 | Create Court Image by Owner | Court Management | This use case allows Owners to upload and assign one or more images to a specific court. These images will be displayed on the booking interface, giving users a visual preview of the court (e.g., surface, lighting, surroundings). |
| 58 | Create Court Image by Manager | Court Management | This use case allows Managers to upload and assign one or more images to a specific court. These images will be displayed on the booking interface, giving users a visual preview of the court (e.g., surface, lighting, surroundings) under the Owner's authority |
| 59 | Delete Court Image by Owner | Court Management | This use case enables Owners to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts. |
| 60 | Delete Court Image by Manager | Court Management | This use case enables Managers to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts under the Owner's authority |
| 61 | View Court Image by Owner | Court Management | This use case enables Owners to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts. |
| 62 | View Court Image  by Manager | Court Management | This use case enables Managers to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts under the Owner's authority |
| 63 | View Court Details | Court Management | Users can see court details, service, position on map, images, and pricing by time slot |
| 64 | View Court | Court Management | This use case allows **Users** and **Guests** to view available courts for booking, along with relevant details such as court type, location, available time slots, price, and court images. **Owners** and **Managers** can view all courts with additional management details, including maintenance status and booking statistics. |
| 65 | View Court Slot | Court Management | This use case allows **Users/Guests** to view available **court time slots** for booking. Time slots show detailed scheduling, availability, and pricing information. |
| 66 | View Court Slot by Owner | Court Management | This use case allows **Owners** to view the full list of time slots associated with each court, including inactive, booked, or under maintenance slots. Time slots show detailed scheduling, availability, and pricing information. |
| 67 | View Court Slot by Manager | Court Management | This use case allows **Managers** to view the full list of time slots associated with each court, including inactive, booked, or under maintenance slots. Time slots show detailed scheduling, availability, and pricing information. |
| 68 | Create Service by Owner | Service Management | This use case enables **Owners** to create new services that will be available for users to purchase. Services can include food and drinks, pickleball racket rental, or coaching sessions. Each service includes a name, description, price, status, and optional images or categories. |
| 69 | Create Service by Manager | Service Management | This use case enables **Managers** to create new services that will be available for users to purchase. Services can include food and drinks, pickleball racket rental, or coaching sessions. Each service includes a name, description, price, status, and optional images or categories under the Owner's authority |
| 70 | Active Status Service by Owner | Service Management | This use case allows Owners to change the status of a service (e.g., water, rental, coaching) that is no longer needed or available. Changing the status will hide the service from both the user interface and the administrative dashboard. If the service is linked to past orders, the system may archive it or mark it as inactive instead of removing it permanently. |
| 71 | Active Status Service by Manager | Service Management | This use case allows Managers to change the status of a service (e.g., water, rental, coaching) that is no longer needed or available. Changing the status will hide the service from both the user interface and the administrative dashboard. If the service is linked to past orders, the system may archive it or mark it as inactive instead of removing it permanently under the Owner's authority |
| 72 | Deactive Status Service by Owner | Service Management | This use case allows Owners to deactivate a service (e.g., water, rental, coaching) that is no longer needed or available. Deactivating the service will hide it from both the user interface and the administrative dashboard. If the service is linked to past orders, the system may archive it or mark it as inactive instead of removing it permanently. |
| 73 | Deactive Status Service by Manager | Service Management | This use case allows Managers to deactivate a service (e.g., water, rental, coaching) that is no longer needed or available. Deactivating the service will hide it from both the user interface and the administrative dashboard. If the service is linked to past orders, the system may archive it or mark it as inactive instead of removing it permanently under the Owner's authority |
| 74 | Update Service by Owner | Service Management | This use case allows the **Owner** to edit details of an existing service (e.g., equipment rental, snacks, water). Updates can include changes to the name, price, description, availability status, category, or associated image. |
| 75 | Update Service by Manager | Service Management | This use case allows the **Manager** to edit details of an existing service (e.g., equipment rental, snacks, water). Updates can include changes to the name, price, description, availability status, category, or associated image under the Owner's authority |
| 76 | View Service | Service Management | This use case enables **Users** and **Guests** to view available services such as drinks, snacks, rentals, |
| 77 | View List of Service by Owner | Service Management | Displays a list of all available services |
| 78 | View List of Service by Manager | Service Management | Displays a list of all available services under the Owner's authority |
| 79 | View List Order for Staff | Staff Management | This use case allows Staff to view a complete list of court and service orders created by users, including booking details, order status, payment status, and time. Staff can use this list to manage on-site service preparation, verify arrivals, and prepare resources for each time slot. |
| 80 | Perform Check-In | Staff Management | This use case allows Staff to mark a booking or service order as "Checked-in" when the user arrives at the facility. This helps ensure proper scheduling, prepare the court, and confirm attendance for reporting and billing purposes. |
| 81 | Create Booking for Customer | Staff Management | This use case enables **Staff** to manually create a court booking on behalf of a **Customer**. This is useful when the customer books directly at the facility, via phone call, or doesn’t have access to the website or app. Staff can select courts, time slots, and input basic customer details. |
| 82 | View Staff Dashboard | Staff Management | This use case allows **Staff** to access the centralized dashboard after logging in. The dashboard provides a summary of daily bookings, service orders, check-in status, upcoming schedules, and quick actions such as creating bookings or processing payments. |
| 83 | Generate QR for Payment | Payment Management | This use case allows Users or Guests to generate a QR code linked to their booking or service order. The QR code contains transaction details and can be scanned using supported banking or e-wallet apps to complete payment or deposit. |
| 84 | Online Payment Processing | Payment Management | Allows users to complete transactions via MB Bank integration. |
| 85 | Receive Notifications | Notification Management | Users receive notifications via email (booking, password reset) and app/web (order status). |
| 86 | View All Transaction History by Owner | Financial Management | This use case allows Owners to access and review the full list of all payment transactions that occurred in the system, including those related to court bookings, service purchases, deposits, and refunds. Transactions include payment method, status, amount, and are filterable for financial reporting and auditing |
| 87 | View All Transaction History by Manager | Financial Management | This use case allows Managers to access and review the full list of all payment transactions that occurred in the system, including those related to court bookings, service purchases, deposits, and refunds. Transactions include payment method, status, amount, and are filterable for financial reporting and auditing under the Owner's authority |
| 88 | View Revenue Report by Owner | Financial Management | This use case enables Owners to access a visual and/or tabular summary of total revenue generated from court bookings, service purchases, and other transactions within a selected time frame. The revenue report helps monitor business performance, identify trends, and support financial decision-making. |
| 89 | View Revenue Report by Manager | Financial Management | This use case enables Managers to access a visual and/or tabular summary of total revenue generated from court bookings, service purchases, and other transactions within a selected time frame. The revenue report helps monitor business performance, identify trends, and support financial decision-making under the Owner's authority |
| 90 | Export Revenue Report by Owner | Financial Management | This use case allows Owners to export the full list of registered users into an Excel file (.xlsx). The export includes essential user data such as name, role, contact information, account status, and registration date. This feature is typically used for administrative review, backups, or reporting. |
| 91 | Export Revenue Report by Manager | Financial Management | This use case allows Managers to export the full list of registered users into an Excel file (.xlsx). The export includes essential user data such as name, role, contact information, account status, and registration date. This feature is typically used for administrative review, backups, or reporting under the Owner's authority |
| 92 | Export Booking Orders by Owner | Financial Management | Enables exporting booking order data to Excel. |
| 93 | Export Booking Orders by Manager | Financial Management | Enables exporting booking order data to Excel under the Owner's authority |
| 94 | Export Transaction by Owner | Financial Management | Export transaction data to Excel. |
| 95 | Export Transaction by Manager | Financial Management | Export transaction data to Excel under the Owner's authority |

***Table 4.1****: Use Cases*

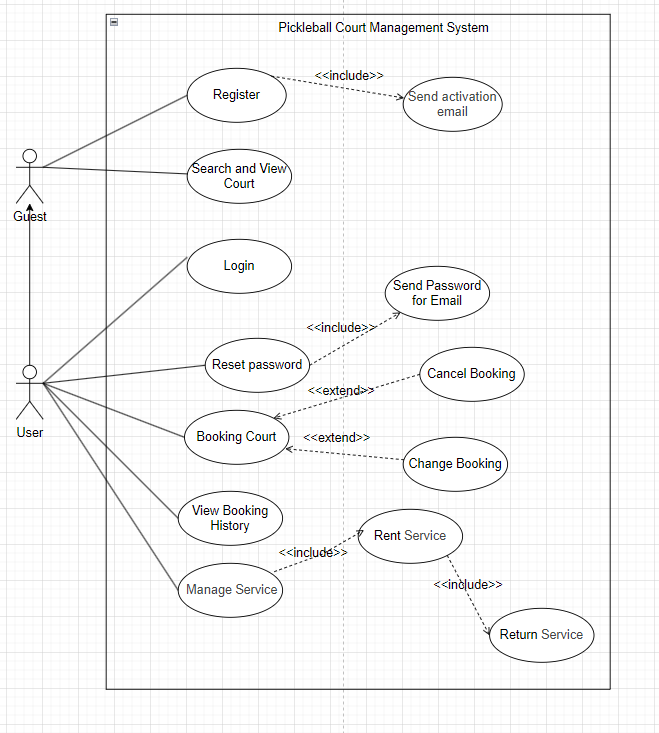
### 1.3.2 Use Case Diagrams

### 1.3.2.1 UseCase for Pickleball Court Management System

**

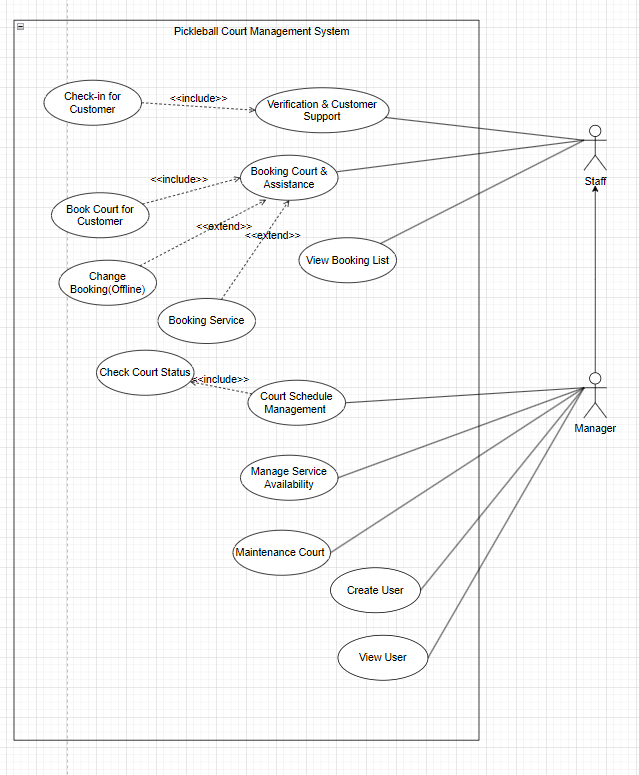
***Figure 4.2****: UseCase for Pickleball Court Management System*

##### 1.3.2.2 UseCase for Users

**

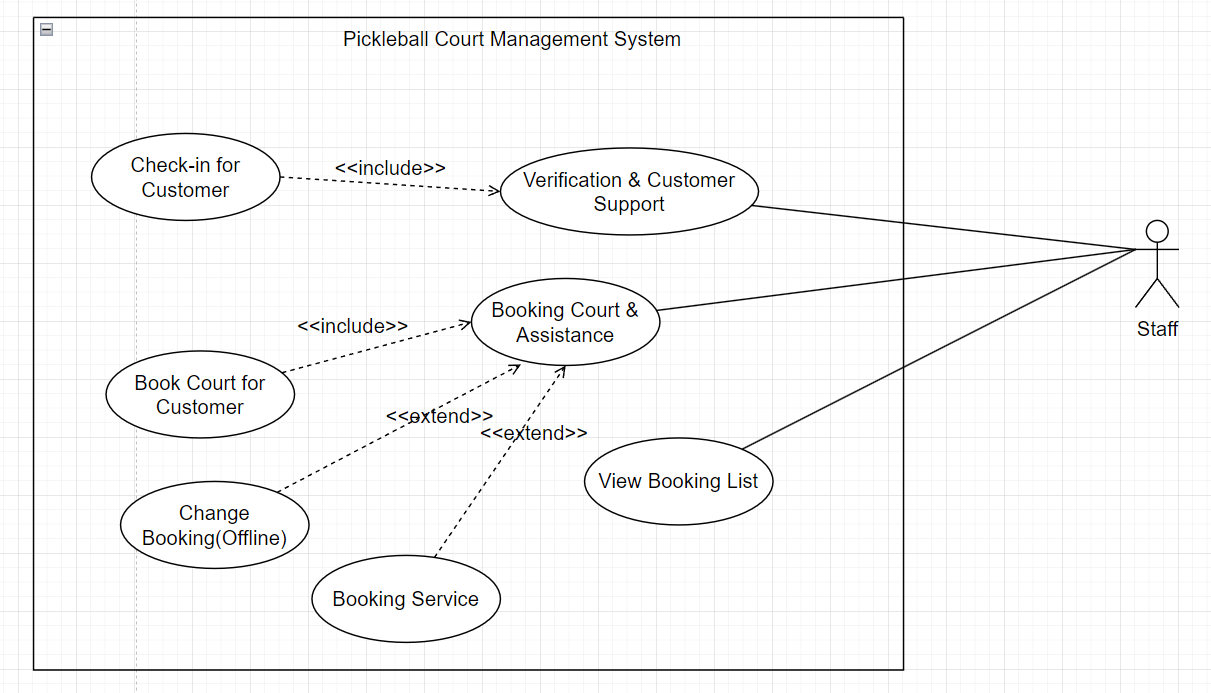
***Figure 4.3****: UseCase for Users*

##### 1.3.2.3 UseCase for Managers



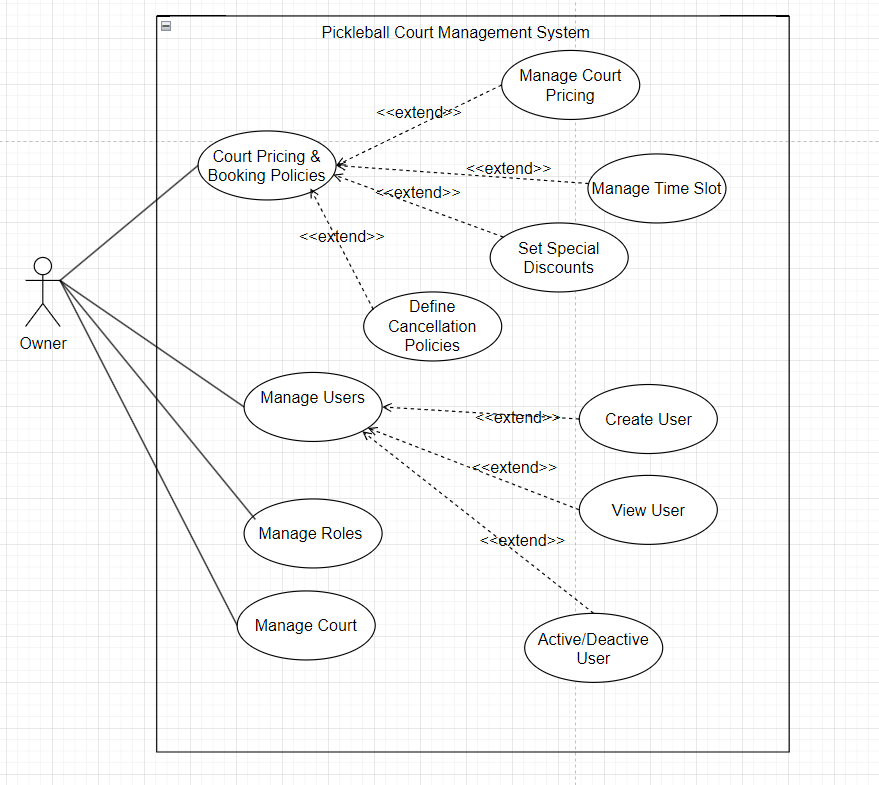
***Figure 4.4****: UseCase for Managers*

##### 1.3.2.4 UseCase for Staff



***Figure 4.5****: UseCase for Staff*

##### 1.3.2.5 UseCase for Owner

**2.1 Common Feature**

#### 2.1.1 Register Student Account

| Primary Actors | Guest | Secondary Actors | Staff |
| --- | --- | --- | --- |
| Description | This use case allows a student to create a user account by providing required information such as a valid university email (e.g., ending with **@edu.vn**) and phone number. The system verifies the information to confirm the student’s eligibility before allowing access to student-specific features (e.g., discounted pricing, student promotions). | | |
| Preconditions | * The student has not previously registered with the same email or phone number. * The system is connected to the student verification mechanism (e.g., email check, token validation). * The registration feature is enabled and accessible. | | |
| Postconditions | * A new student account is successfully created and marked with the **student role**. * The student receives confirmation via email or notification. * The student is granted access to benefits such as student pricing. | | |
| Normal Sequence/Flow | 1. Student navigates to the **Register** page. 2. Select “Register as Student”. 3. Inputs required information:  * First Name * Last Name * Student email (e.g., abc@university.edu) * Phone number * Password * Date of birth * Gender  1. System validates:  * Email domain (e.g., must end in edu.vn) * Unique email/phone  1. If valid, the system creates a **student user account** and assigns appropriate roles. 2. System sends confirmation messages via email. 3. Student is logged in and redirected to the home screen. | | |
| Alternative Sequences/Flows | * **Invalid Email Format:** * System shows an error: “Please use a valid university email address.” * **Duplicate Email or Phone:** * System shows: “This email/phone is already in use.” | | |

#### 2.1.2 Login

| Primary Actors | User/Guest, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | Allows users to log in and access the system. The system will authenticate and authorize the user. | | |
| Preconditions | User account has been created & authorized. | | |
| Postconditions | * User logs in the system successfully * The system tracked successful login into the Activity Log | | |
| Normal Sequence/Flow | 1. User clicks on the link to navigate to the login page. 2. User enter username/gmail/phone number and password and click on “Đăng nhập” (A-F\_1) 3. System redirect user to the home page | | |
| Alternative Sequences/Flows | **A-F\_1**. **System can’t authenticate the user**  Users can not login if:   1. User leaves the email/username/phone number or password field blank (MSG01) 2. User enters the invalid email/username/phone number (MSG02) 3. User input wrong password format (MSG03, MSG04, MSG05) 4. The account is not registered with the system or is blocked (MSG07) | | |

#### 2.1.3 Logout

| Primary Actors | User, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | Users can securely terminate their active sessions | | |
| Preconditions | User has account and logged in | | |
| Postconditions | User logout successfully | | |
| Normal Sequence/Flow | 1. User click the avatar and click button “Log out” 2. System logs out the user | | |
| Alternative Sequences/Flows | If the user's token times out, the user will be automatically logged out. | | |

#### 2.1.4 Change Password

| Primary Actors | User, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the user to reset the password. | | |
| Preconditions | * User has account and logged in * User is on profile page | | |
| Postconditions | User resets the password for the account. | | |
| Normal Sequence/Flow | 1. User get on to the profile page and click on “Đặt lại mật khẩu” button 2. System display Change Password form 3. User enter old password, new password and re-password and click on “Save” (A-F\_1) 4. System display success message (MSG07) | | |
| Alternative Sequences/Flows | **A-F\_1: System cannot change password**  User cannot change to new password if:   1. User input wrong password format (MSG03, MSG04, MSG05) 2. User input re-password not match the password (MSG06) | | |

#### 2.1.5 Register New User

#### 

| Primary Actors | Guest | Secondary Actors |  |
| --- | --- | --- | --- |
| Description | As a new user, I want to register an account so that I can book pickleball courts, and manage my profile within the system. | | |
| Preconditions | The user must have access to a valid email or phone number for verification. | | |
| Postconditions | 1. The new user account is successfully created. 2. The user can log in and access the system based on their assigned role. | | |
| Normal Sequence/Flow | 1. The system prompts the user to enter required information (e.g., name, email, phone, password). 2. The user is redirected to the login page with a success message. | | |
| Alternative Sequences/Flows | **AF1: Email already exists in the system**   * The system notifies the user that the email is already in use and prompts them to log in or recover their account.   **AF2: System or network failure during registration**   * The system displays an error message and prompts the user to try again later. | | |

### 2.2 User Management

#### 2.2.1 Update User Profiles

| Primary Actors | User | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | As a registered user, I want to update and manage my profile details so that I can ensure my information is accurate and up to date. | | |
| Preconditions | 1. The user must be logged into the system. 2. The system must have existing user profile data. | | |
| Postconditions | 1. The user’s profile information is successfully updated. 2. If information is changed, a confirmation notification is sent. | | |
| Normal Sequence/Flow | 1. The user navigates to the profile management page. 2. The system displays the current profile information. 3. The user updates one or more profile fields (e.g., name, contact info, password). 4. The user submits the changes. 5. The system validates the input data and updates the profile. 6. A success message is displayed, and a notification is sent if necessary. | | |
| Alternative Sequences/Flows | **AF1: User enters an invalid email or phone number**   * The system displays an error message and asks for a valid input.   **AF2: System failure during update**   * The system logs the error and prompts the user to try again later. | | |

#### 

#### 2.2.2. Update Role User by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Owner to update the role of an existing user (e.g., Staff, Student, Manager, User) changes affect what functionalities the user can access in the system. | | |
| Preconditions | * The actor is logged in as an Owner * The user must already exist in the system. * The new role must be valid and active in the system. | | |
| Postconditions | * The selected user has a new role assigned. * Their access rights are updated based on the new role. * The update is logged for auditing purposes (if applicable). | | |
| Normal Sequence/Flow | 1. Owner logs into the system. 2. Navigates to **User Management** section. 3. Searches for and selects the user whose role needs to be updated. 4. Clicks the **"Edit"** button next to the user’s profile. 5. Selects the new role from a dropdown list (e.g., User, Staff, Manager). 6. Clicks **"Save"** . 7. System validates the role change and updates the user’s profile. 8. A confirmation message is shown (e.g., "Success"). 9. The new role permissions take effect immediately on the next login or page refresh. | | |
| Alternative Sequences/Flows | 1. **Invalid Role Selected:**  * System displays an error if the role is inactive or unauthorized for assignment.  1. **Update Fails (e.g., network/server issue):**  * System displays an error message and logs the issue for retry. | | |

#### 2.2.3 Update Role User by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Manager to update the role of an existing user (e.g., Staff, Student, Manager, User) changes affect what functionalities the user can access in the system | | |
| Preconditions | * The actor is logged in as an Manager * The user must already exist in the system. * The new role must be valid and active in the system. | | |
| Postconditions | * The selected user has a new role assigned. * Their access rights are updated based on the new role. * The update is logged for auditing purposes (if applicable). | | |
| Normal Sequence/Flow | 1. Owner logs into the system. 2. Navigates to **User Management** section. 3. Searches for and selects the user whose role needs to be updated. 4. Clicks the **"Edit"** button next to the user’s profile. 5. Selects the new role from a dropdown list (e.g., User, Staff, Manager). 6. Clicks **"Save"** . 7. System validates the role change and updates the user’s profile. 8. A confirmation message is shown (e.g., "Success"). 9. The new role permissions take effect immediately on the next login or page refresh. | | |
| Alternative Sequences/Flows | * **Invalid Role Selected:** * System displays an error if the role is inactive or unauthorized for assignment. * **Update Fails (e.g., network/server issue):** * System displays an error message and logs the issue for retry. | | |

#### 

#### 2.2.4 Create Role

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to define and create a **new role** in the system, such as "Coach", "Customer Support", or any custom role. A role defines the level of access a user has to specific features and modules. After creation, the role can be assigned to users and have permissions configured accordingly. | | |
| Preconditions | 1. The actor is authenticated as **Owner**. 2. The system is in a state that allows configuration changes. 3. The role name to be created is unique. | | |
| Postconditions | 1. A new role is successfully created in the system. 2. The role is visible in the role management module. 3. Permissions can now be assigned to the new role. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Role Management**. 3. Clicks **“Create New Role”**. 4. Enters:  * Role name (e.g., "Staff Assistant") * Optional description  1. Clicks **“Add”**. 2. System checks for name uniqueness and validates input. 3. System creates the new role and confirms success. 4. Owner is redirected to assign permissions to the new role (optional next step). | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * System display: “Error” | | |

#### 2.2.5 Delete Role

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to delete a custom-defined role from the system. The system will ensure that the role is **not currently assigned** to any user before allowing deletion. This helps maintain role management cleanliness and prevent unused roles from cluttering the system. | | |
| Preconditions | * The actor is authenticated as an **Owner**. * The role to be deleted is **not a system default role** (e.g., User, Manager). * The role is **not assigned** to any active users. | | |
| Postconditions | * The selected user has a new role assigned. * Their access rights are updated based on the new role. * The update is logged for auditing purposes (if applicable). | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to  **Role Management**. 3. Sees a list of roles with action buttons. 4. Clicks **“Delete”** next to a custom role. 5. System checks if the role is:  * Custom-defined (not system default) * Not assigned to any users  1. System deletes the role and shows success message. | | |
| Alternative Sequences/Flows | 1. **Owner Cancels Confirmation:**  * No action is taken; role remains.  1. **System Error During Deletion:**  * System displays error , try again again. | | |

#### 2.2.6 View Role

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to view a list of all defined user roles in the system, including system default roles (e.g., User, Guest, Staff) and custom roles created by the Owner. The role view includes details such as the role name, description, assigned permissions, and the number of users currently using the role. | | |
| Preconditions | * The actor is authenticated as **Owner** * The role management module is enabled. * Roles have already been defined in the system. | | |
| Postconditions | * The system displays a list of all roles. * The actor may view role details, including associated permissions and user count. * (Optional) The actor can navigate to **Edit**, **Assign**, or **Delete** role actions. | | |
| Normal Sequence/Flow | 1. Actor logs into the admin dashboard. 2. Navigates to **Role Management → View Roles**. 3. System displays a table with the following info per role:  * Role name * Role description  1. Actor clicks on a specific role to view details. 2. System shows:  * Full description | | |
| Alternative Sequences/Flows | 1. **No Roles Exist:**  * System displays: “No data”  1. **Role Data Fails to Load:**  * System shows error:”Failed to load data” | | |

#### 

#### 2.2.7 View List of Users

| Primary Actors | Owner, Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | The "View User List" use case allows the Owner, Manager to view a list of all registered users in the Pickleball Court Management System. The list includes filtering options, allowing the Owner to refine the users displayed based on criteria such as login name, phone number, email, role. | | |
| Preconditions | * The Owner, Manager must be logged into the system using valid credentials. * The Owner, Manager must have appropriate permissions to access user information. | | |
| Postconditions | * The system displays a list of registered users, reflecting any filters applied by the Owner. | | |
| Normal Sequence/Flow | 1. **Access User Management:** The Owner navigates to the user management section of the system (e.g., via the menu or dashboard). 2. **Show All Users**: The system displays a list of all registered users, including details such as username, role, email, and phone number. 3. **Apply Filters**: The Owner selects filtering options (e.g., filter by role, phone number, or username). 4. **Update List**: The system refreshes the list to display only users that match the selected filters. 5. **View Details**: The Owner can select a user from the list to view additional details. | | |
| Alternative Sequences/Flows | 1. **No Matching Users**:  * If no users match the applied filters in step 4:   + - The system displays empty list.     - The Owner can modify or clear the filters to see the full list again.  1. **No Filters Used**:  * If the Owner skips step 3 and does not apply filters:   + - The system continues to show the complete list of all registered users.     - The Owner may proceed to step 5 to view individual user details.  1. **Data Retrieval Error**:  * If the system fails to load the user list in step 2 or step 4:   + - The system displays an error message (e.g., "Fail.").     - The Owner is prompted to retry.  1. **Exit Use Case**:  * At any point, if the Owner chooses to leave the user list view:   + - The system returns the Owner to the previous screen (e.g., dashboard).     - The use case ends. | | |

#### 2.2.8 Create New User by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables Owners to create new user accounts in the Pickleball Court Management System. Owners can input essential user details such as name, email, phone number, and assign an appropriate role (e.g., user, staff, manager) to the new user. | | |
| Preconditions | * The Owners must be logged into the system with valid credentials and possess the necessary permissions to create user accounts. * The email address provided for the new user must not already exist in the system. * The phone number provided for the new user must not already exist in the system. * The username provided for the new user must not already exist in the system. | | |
| Postconditions | * A new user account is successfully created with the provided details and assigned role. | | |
| Normal Sequence/Flow | 1. **Owner initiates user creation**: The Owner navigates to the user management section of the system and selects the option to create a new user. 2. **System provides a form**: A user creation form appears, containing fields for details such as name, email, phone number, and role,... 3. **Owner inputs user information**: The Owner enters the required details for the new user and selects a role from a predefined list (e.g., user, staff, owner). 4. **Owner submits the form**: After reviewing the entered data, The Owner submits the form. 5. **System validates the input**: The system checks the data for completeness and accuracy (e.g., ensuring the email format is valid and the email is unique). 6. **System creates the account**: Upon successful validation, the system generates a new user account with the provided details and assigned role. 7. **System confirms success**: The Owner receives a confirmation message indicating that the new user account has been successfully created. | | |
| Alternative Sequences/Flows | 1. Invalid Data Submission 2. **Scenario**: At step 5, if the system identifies invalid data (e.g., missing fields or incorrect email format):    * The system displays an error message detailing the issues.    * The Owner corrects the errors and resubmits the form.    * The process resumes at step 4. 3. Email, Phone number, Username Already in Use 4. **Scenario**: At step 5, if the email address, phone number, username is already registered:    * The system shows an error message stating the email, phone number, username is in use and prompts for a different email.    * The Owner provides a new email, phone number, username and resubmits the form.    * The process resumes at step 4. 5. Role Assignment Error 6. **Scenario**: At step 3, if The Owner and Manager selects an unavailable or restricted role:    * The system displays an error message indicating the role cannot be assigned.    * The Owner chooses a valid role and proceeds.    * The process resumes at step 4. 7. Notification Failure 8. **Scenario**: At step 7, if the notification fails to send (e.g., due to an invalid email or technical issue):    * The system logs the error and warns The Owner that the account was created but the notification failed.    * The Owner may manually notify the user or retry sending the notification.    * The process continues to step 8. 9. The Owner Cancels the Process 10. **Scenario**: After step 3, if The Owner decides not to proceed:     * The system discards the entered data and returns The Owner to the user management screen.     * The use case ends without creating an account. | | |

#### 2.2.9 Create New User by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to create a new user account in the Pickleball Court Management System and assign one of the roles: Staff, User, or Student | | |
| Preconditions | * The Manager must be logged into the system with valid credentials and possess the necessary permissions to create user accounts. * The email address provided for the new user must not already exist in the system. * The phone number provided for the new user must not already exist in the system. * The username provided for the new user must not already exist in the system. | | |
| Postconditions | * A new user account is successfully created with the provided details and assigned role. | | |
| Normal Sequence/Flow | 1. **Manager initiates user creation**: The Manager navigates to the user management section of the system and selects the option to create a new user. 2. **System provides a form**: A user creation form appears, containing fields for details such as name, email, phone number, and role,... 3. **Manage inputs user information**: The Manager enters the required details for the new user and selects a role (dropdown: Staff, User, Student) 4. **Manage submits the form**: After reviewing the entered data, The Owner and Manager submits the form. 5. **System validates the input**: The system checks the data for completeness and accuracy (e.g., ensuring the email format is valid and the email is unique). 6. **System creates the account**: Upon successful validation, the system generates a new user account with the provided details and assigned role. 7. **System confirms success**: The Manager receives a confirmation message indicating that the new user account has been successfully created. | | |
| Alternative Sequences/Flows | 1. Invalid Data Submission 2. **Scenario**: At step 5, if the system identifies invalid data (e.g., missing fields or incorrect email format):    * The system displays an error message detailing the issues.    * The Manager corrects the errors and resubmits the form.    * The process resumes at step 4. 3. Email, Phone number, Username Already in Use 4. **Scenario**: At step 5, if the email address, phone number, username is already registered:    * The system shows an error message stating the email, phone number, username is in use and prompts for a different email.    * The Manager provides a new email, phone number, username and resubmits the form.    * The process resumes at step 4. 5. Role Assignment Error 6. **Scenario**: At step 3, if The Manager selects an unavailable or restricted role:    * The system displays an error message indicating the role cannot be assigned.    * The Manager chooses a valid role and proceeds.    * The process resumes at step 4. 7. Notification Failure 8. **Scenario**: At step 7, if the notification fails to send (e.g., due to an invalid email or technical issue):    * The system logs the error and warns The Manager that the account was created but the notification failed.    * The Manager may manually notify the user or retry sending the notification.    * The process continues to step 8. 9. The Manager Cancels the Process 10. **Scenario**: After step 3, if The Manager decides not to proceed:     * The system discards the entered data and returns The Manager to the user management screen.     * The use case ends without creating an account. | | |

#### 2.2.10 Deactivate User

| Primary Actors | Owner | Secondary Actors | User |
| --- | --- | --- | --- |
| Description | This use case enables an Owner to deactivate a user’s account in the Pickleball Court Management System. Once deactivated, the user can no longer log in or access any system features, but their data remains stored for potential reactivation later. | | |
| Preconditions | * The Owner must be logged into the system with valid credentials. * The Owner must have sufficient permissions to manage user accounts. * The user account targeted for deactivation must exist in the system and be currently active. | | |
| Postconditions | * The user’s account is marked as deactivated, blocking their ability to log in or use the system. * The user’s data is retained in the system for possible future reactivation. | | |
| Normal Sequence/Flow | 1. **Access User Management:** The Owner navigates to the user management section of the system web interface. 2. **View list users:** The system displays a list of user with status. 3. **Select a user:** The Owner selects the activated user account from the list. 4. **Start Activation:** The Owner selects the activation button for the selected user 5. **Update Account Status:** The system changes the user's status from "Activated" to "Deactive". 6. **Show notification:** The system displays “ Cập nhật trạng thái thành công” . | | |
| Alternative Sequences/Flows | 1. **System Error During Activation**:  * In step 5, if the system encounters an error while updating the account status:   The system displays an error message ("Cập nhật trạng thái thất bại").  If the retry is successful, the use case continues from step 5; otherwise, it ends. | | |

#### 2.2.11 Activate User

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to activate a previously deactivated user account within the Pickleball Court Management System. Once activated, the user regains access to their account and associated functionalities (e.g., booking courts, viewing orders). | | |
| Preconditions | * The Owner must be logged into the system with valid credentials and have permissions to manage user accounts. * The user account to be activated must exist in the system and be in a deactivated state. | | |
| Postconditions | The selected user account is activated, and the user can log in and use the system as per their assigned role and permissions. | | |
| Normal Sequence/Flow | 1. **Access User Management:** The Owner navigates to the user management section of the system web interface. 2. **View list users:** The system displays a list of user with status. 3. **Select a user:** The Owner selects the deactivated user account from the list. 4. **Start Activation:** The Owner selects the activation button for the selected user 5. **Update Account Status:** The system changes the user's status from "Deactivated" to "Active". 6. **Show notification:** The system displays “ Cập nhật trạng thái thành công” . | | |
| Alternative Sequences/Flows | 1. **System Error During Activation**:  * In step 5, if the system encounters an error while updating the account status:   The system displays an error message ("Cập nhật trạng thái thất bại").  If the retry is successful, the use case continues from step 5; otherwise, it ends. | | |

#### 

#### 

### *2.3 Booking Management*

#### 2.3.1 Booking Service On Mobile

| Primary Actors | User, Guest | Secondary Actors | System Notification |
| --- | --- | --- | --- |
| Description | This use case allows users to purchase additional services via the mobile app (e.g. drinks, snacks,... ) instead of having to buy them directly at the counter. Users can select services, make payment through a generated QR code, and once the payment is successful, the order is sent to the staff for processing and fulfillment. | | |
| Preconditions | * The user is logged in (or is a guest). * There are available services listed in the system. * The system is connected to a valid payment processor for QR generation. | | |
| Postconditions | * The number of services is reduced as the user completes payment * A QR code is generated and scanned to complete payment. * The order status is updated to "Đã thanh toán" or "Đặt dịch vụ tại sân". * The service order appears in the user’s order history. | | |
| Normal Sequence/Flow | 1. User/Guest navigates to the Services section. 2. System displays available services (e.g., drinks, snacks). 3. User/Guest selects one or more services để tính tiền 4. User/Guest xem lại chi tiết dịch vụ và tổng tiền ứng với dịch vụ đã chọn. 5. System generates a QR code showing the total amount. 6. User/Guest scans the QR code with their banking/payment app. 7. System confirms payment. 8. The order is marked as paid and sent to staff (if applicable). 9. User/Guest receives a notification "Đặt dịch vụ tại sân thành công". | | |
| Alternative Sequences/Flows | 1. **No services available:**  The system displays “No services available at this time” 2. **User fails to scan QR code in time:**  QR code expires, and the user can no longer make the payment 3. **Order processing fails:**  The system will refund the money to the bank account used for the payment 4. **User places an order but hasn’t paid:**  The user can cancel the order or proceed with the payment 5. **Service sold out:**  The service will display a quantity of 0 | | |

#### 2.3.2 Change Booking

| Primary Actors | User | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a user to change an existing court reservation. The system supports one-time rescheduling, allowing users to change their court, date, or time within the allowed policy. The system handles payment differences automatically (e.g., generate new QR if extra cost, refund if lower). | | |
| Preconditions | * The user is authenticated (or a guest with an active booking session). * The user has at least one active booking that is eligible for modification. * The modification is made within the system’s allowed timeframe. * Only one update per booking is allowed. | | |
| Postconditions | * The booking is updated with new details (court/time/date). * If the new court costs more, a new QR code is generated for the additional amount. * If the new court costs less, the system initiates an automatic refund. | | |
| Normal Sequence/Flow | 1. User logs in and navigates to **Booking History**. 2. Selects a booking and clicks **“Change Booking”**. 3. System checks if the booking is eligible for modification (not expired, not modified before). 4. User selects new court, date, or time. 5. System compares the new booking price to the original:  * If equal: update is applied immediately. * If higher: system generates new QR for the additional amount. * If lower: system processes an automatic refund.  1. User confirms the changes. 2. System updates the booking and sends a confirmation message. | | |
| Alternative Sequences/Flows | 1. **Booking Already Modified Once:**  * System shows a message: “This booking has already been modified and cannot be updated again.”  1. **New Slot Unavailable:**  * System prevents selection and prompts the user to choose another time.  1. **Additional Payment Required:**  * QR code is generated. System waits for successful payment before confirming update.  1. **Price Reduction:**  * System notifies user that the difference will be refunded automatically.  1. **User Cancels Before Confirming:**  * No changes are saved; system returns to booking details.  1. **7A. Update Fails (Network/Server Issue):**  * Displays error message and allows retry. | | |

#### 2.3.3 View Booking History User by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to view the entire booking history of a user, for example, within a specific time period, by venue, by day, by month, etc. This information helps the owner track the customer's booking behavior and provide better services | | |
| Preconditions | * The Owner must be logged into the system using valid credentials. * The Owner must have appropriate permissions to access user information. | | |
| Postconditions | The system displays all booking history of the User, including details of previous and upcoming bookings as well as their payment information. | | |
| Normal Sequence/Flow | 1. Owner logs into the system. 2. Navigates to Order section. 3. Displays all booking history of the User | | |
| Alternative Sequences/Flows | 1. **No Booking History**:  * The system no displays data   2. **User Does Not Exist:**   * The system checks the database but does not find the corresponding User.   + The system displays an error message: “No user found. Please check” | | |

#### 

#### 2.3.4 View Booking History User by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to view the entire booking history of a user, for example, within a specific time period, by venue, by day, by month, etc. This information helps the owner track the customer's booking behavior and provide better services under the Owner's authority | | |
| Preconditions | * The Manager must be logged into the system using valid credentials. * The Manager must have appropriate permissions to access user information. | | |
| Postconditions | The system displays all booking history of the User, including details of previous and upcoming bookings as well as their payment information. | | |
| Normal Sequence/Flow | 1. Manager logs into the system. 2. Navigates to Order section. 3. Displays all booking history of the User under the Owner's authority | | |
| Alternative Sequences/Flows | 1. **No Booking History**:  * The system no displays data   2. **User Does Not Exist:**   * The system checks the database but does not find the corresponding User.   + The system displays an error message: “No user found. Please check” | | |

#### 

#### 2.3.5 Cancel Booking

| Primary Actors | User | Secondary Actors | System Notification Service |
| --- | --- | --- | --- |
| Description | This use case allows users to cancel their existing court bookings in the Pickleball Court Management System. Depending on the system's cancellation policy, canceling a booking may result in the loss of the deposit paid by the user at the time of booking. | | |
| Preconditions | * The user must have an active booking in the system. * The user must be logged in (for registered users) or provide booking details (for guests) to access the booking. | | |
| Postconditions | * The booking is canceled in the system, and the court becomes available for others to book. * If applicable, the user’s deposit is forfeited as per the cancellation policy. * The user receives a confirmation of the cancellation. | | |
| Normal Sequence/Flow | 1. **Access Booking Details**: The user navigates to the booking management section (e.g., “My Bookings” for registered users or a booking lookup for guests). 2. **Select Booking**: The system displays the user’s active bookings, and the user selects the specific booking to cancel. 3. **Initiate Cancellation**: The user chooses the option to cancel the selected booking. 4. **Display Cancellation Policy**: The system informs the user of the cancellation policy, including potential deposit loss (e.g., “Canceling this booking will result in the loss of your deposit.”). 5. **Confirm Cancellation**: The user confirms their intent to cancel the booking. 6. **Process Cancellation**: The system cancels the booking, updates the court availability, and marks the deposit as forfeited (if applicable). 7. **Notify User**: The system sends a confirmation to the user via email (e.g., “Your booking has been canceled. Your deposit has been forfeited.”). | | |
| Alternative Sequences/Flows | 1. **Cancellation Within Grace Period**:  * If the cancellation occurs within a predefined grace period (e.g., 24 hours before the booking time) where no deposit loss applies:   + - At step 4, the system informs the user that no deposit will be lost.     - At step 6, the deposit is refunded or not deducted, and the booking is canceled.     - The use case continues to step 7 with a notification (e.g., “Your booking has been canceled, and your deposit is safe.”).  1. **User Cancels Cancellation**:  * After step 4, if the user decides not to proceed with the cancellation:   + - The user selects an option to abort the cancellation (e.g., “Keep Booking”).     - The system returns the user to the booking management screen, and the booking remains active.     - The use case ends.  1. **Booking Not Found**:  * At step 2, if the selected booking cannot be located (e.g., already canceled or expired):   + - The system displays an error message (e.g., “Booking not found or already canceled.”).     - The user is prompted to select a different booking or refresh the list.     - The use case returns to step 1.  1. **Notification Failure**:  * At step 7, if the system fails to send the cancellation confirmation (e.g., due to an email service error):   + - The cancellation is still processed, and the system logs the failure.     - The user can view the cancellation status in the app, but no external notification is sent. | | |

#### 2.3.6 View Booking Personal

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to view a list of their own court bookings, including information such as booking date, time slot, court name, booking status, and payment status. Users can access and review both upcoming and past bookings in their personal dashboard. | | |
| Preconditions | * User or Guest is authenticated (or recognized by session token for guest booking). * There are existing bookings linked to the user’s account or session. | | |
| Postconditions | * The user successfully views a list of their bookings. * The system displays updated statuses (e.g., confirmed, canceled, completed). | | |
| Normal Sequence/Flow | 1. User/Guest logs into the system (or is already authenticated). 2. User/Guest navigates to **Personal Dashboard → My Bookings**. 3. System retrieves all bookings linked to the user/guest account. 4. System displays booking information including:  * Booking ID / reference number * Court name and location * Date and time slot * Booking status (Pending / Confirmed / Canceled / Completed) * Payment status (Paid / Deposit / Pending)  1. User can filter bookings by:  * Status * Date range  1. User clicks on a booking to view detailed information (optional). | | |
| Alternative Sequences/Flows | 1. **No Bookings Found:**  * System shows message: “No bookings found. Start booking your first court!”  1. **Booking Data Retrieval Error:**  * System shows error message: “Unable to load bookings. Please try again later.” | | |

#### 2.3.7 View Booking Detail Personal

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to view the detailed information of a specific court booking that they have made. The detail view includes comprehensive information about the court, booking time, services added (if any), payment status, and other relevant booking actions (e.g., eligible for cancellation or rescheduling). | | |
| Preconditions | * User or Guest is authenticated (or session is valid). * The booking record exists and is linked to the user or session. | | |
| Postconditions | * The system displays full details of the selected booking. * User can take available actions based on booking status (view only, cancel, reschedule if permitted). | | |
| Normal Sequence/Flow | 1. User/Guest navigates to **My Bookings**. 2. User/Guest selects a booking from the list. 3. System retrieves detailed information for the selected booking. 4. System displays booking details including:  * Booking ID / Reference Code * Court Name and Location * Booking Date and Time Slot * Booking Type (Daily, Regular, Optimized) * Booking Status (Pending, Confirmed, Completed, Canceled) * Payment Details (Amount paid, Deposit, Payment method, Payment status) * Services or Equipment Ordered (if any) * Cancellation/Rescheduling Options (if applicable)  1. User/Guest reviews the information. | | |
| Alternative Sequences/Flows | 1. **Booking Not Found:**  * System displays: “Booking not found or already removed.”  1. **Unauthorized Access:**  * System prevents access if the user tries to view a booking that does not belong to them. | | |

#### 2.3.8 Confirm Booking

| Primary Actors | User, Guest, Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest (or Staff on behalf of a customer) to confirm a court booking after initial creation. Confirmation can happen automatically after successful payment or manually if staff intervention is required. The confirmation secures the booking, reserves the court, and updates the booking status from "Pending" to "Confirmed." | | |
| Preconditions | * A booking has been created and is in a **Pending** status. * Required payment has been made (deposit or full payment as required by booking rules). * The court and time slot are still available. | | |
| Postconditions | * The booking status is updated to **Confirmed**. * The court time slot is locked for that user (unavailable to others). * A booking confirmation notification (email/SMS/in-app) is sent to the user. | | |
| Normal Sequence/Flow | 1. User completes the booking process and initiates payment (if not already done). 2. System verifies:  * Payment is successful (or deposit received). * Court/time slot availability remains unchanged.  1. System updates booking status to **Confirmed**. 2. System generates a booking confirmation message with booking details. 3. System sends a notification to the user (email/SMS/app). 4. User views confirmation in the **Booking Personal** or **Booking Detail** screens. | | |
| Alternative Sequences/Flows | 1. **Payment Fails:**  * System keeps the booking in **Pending** status and prompts user to retry payment.  1. **Court/Slot Becomes Unavailable Before Confirmation:**  * System informs user: “Selected court and time slot are no longer available.”  1. **Staff-initiated Confirmation:**  * Staff manually confirms a booking (e.g., walk-in customer) without requiring online payment (if applicable). | | |

#### 

#### 2.3.9 Deposit Booking

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a **User** or **Guest** to place a **deposit** to temporarily secure a court booking. By paying a predefined deposit amount instead of the full price, the system locks the booking until the remaining balance is paid (either before check-in or within a specified deadline). | | |
| Preconditions | * A booking has been created and is in **Pending** status. * Court and time slot are still available. * Deposit policy and amount are defined in the system. | | |
| Postconditions | * A deposit payment is successfully made. * The booking status is updated to **Deposit Paid** or **Partial Payment**. * The system records the deposit transaction and updates booking history. | | |
| Normal Sequence/Flow | 1. User/Guest creates a booking and selects **Deposit Payment** option. 2. System calculates and displays the required deposit amount (e.g., 30% of total booking value). 3. User/Guest confirms and proceeds to the payment gateway (QR code/payment link). 4. User completes the deposit payment. 5. System verifies payment success with the payment gateway. 6. System updates booking status to **Deposit Paid**. 7. System sends a confirmation notification to the user. 8. System schedules a reminder for full payment deadline (if applicable). | | |
| Alternative Sequences/Flows | 1. **Payment Failure:**  * System displays error: “Deposit payment failed. Please try again.”  1. **Payment Timeout:**  * If payment is not completed within X minutes, system cancels the booking.  1. **Deposit Option Disabled:**  * If deposit payment is not allowed for a particular booking (e.g., close to play time), system only allows full payment. | | |

#### 

#### 2.3.10 Payment Booking

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to pay for a court booking, either by completing the full amount at once or completing the remaining balance after a deposit. Successful payment finalizes the booking process, secures the court, and transitions the booking status to Confirmed. | | |
| Preconditions | * A booking exists with status **Pending** or **Deposit Paid**. * Court and time slot are still available. * Payment gateway is integrated and operational. | | |
| Postconditions | * Payment is successfully processed. * Booking status is updated to **Confirmed**. * System generates an invoice and transaction record. * Confirmation notification is sent to the user. | | |
| Normal Sequence/Flow | 1. User/Guest accesses My Bookings or completes the initial booking creation. 2. System prompts the user to Pay Now (full amount or remaining balance if deposit already paid). 3. User reviews payment details (amount, court name, booking date, time slot). 4. User selects payment method (e.g., QR Code, Bank Transfer, E-Wallet). 5. System redirects to the payment gateway. 6. User completes the payment. 7. System verifies payment success. 8. System updates booking status to Confirmed. 9. System generates and saves the payment invoice. 10. System sends confirmation notification (email/SMS/app) to user. | | |
| Alternative Sequences/Flows | 1. **Payment Failure:**  * System displays error: “Payment failed. Please try again.” * Booking remains in **Pending** or **Deposit Paid** status.  1. **Payment Timeout:**  * If no payment after X minutes, booking may expire or user is reminded to retry.  1. **Already Fully Paid:**  * If booking is already paid in full, system prevents duplicate payments. | | |

#### 2.3.11 Refund on cancellation

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to receive a refund after canceling a booking, based on the system's cancellation and refund policy. The refund amount may vary depending on whether the user paid in full or only made a deposit, and how close to the booking time the cancellation occurs. | | |
| Preconditions | * The booking was already **paid** (full payment or deposit). * The user initiates cancellation **within the allowable cancellation window** (if any). * System has the payment and banking information available to process the refund. | | |
| Postconditions | * The refund (full, partial, or none) is calculated and processed. * System updates the booking status to **Canceled**. * Refund transaction is recorded and a notification is sent to the user. | | |
| Normal Sequence/Flow | 1. User/Guest accesses **My Bookings** and selects a booking to cancel. 2. User clicks **Cancel Booking**. 3. System checks:  * Payment status * Cancellation policy (full refund, partial refund, deposit forfeited)  1. System calculates the eligible refund amount:  * **If full payment:** Refund = Total payment – Deposit (if deposit is non-refundable) * **If deposit only:** No refund or partial refund (depending on timing)  1. System sends refund request to the Bank API/Payment Gateway. 2. Bank processes the refund and returns confirmation. 3. System updates booking status to **Canceled**. 4. System logs the refund transaction. 5. User receives notification about refund processing status. | | |
| Alternative Sequences/Flows | 1. **Bank API Error/Failure:**   System retries or flags refund for manual processing.   1. **Cancellation Not Allowed (e.g., too close to booking time):**  * System informs the user: “Cancellation not eligible for refund at this time.”  1. **No Refund Eligible:**  * System shows message: “This booking is not eligible for a refund based on the policy.” | | |

#### 2.3.12 Refund for rescheduling

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to receive a partial refund when they reschedule a booking to a cheaper court or time slot. The system automatically calculates the price difference and initiates the refund process while retaining the original deposit amount (if applicable). | | |
| Preconditions | * The booking has been paid in full or deposit paid. * User successfully reschedules the booking to a cheaper court or cheaper time slot. * Payment and refund integration is active (Bank API or Payment Gateway). | | |
| Postconditions | * Refund for the price difference (excluding deposit) is processed automatically. * Booking is updated with new schedule details. * Refund transaction is logged and user is notified. | | |
| Normal Sequence/Flow | 1. User/Guest selects a booking and chooses the **Reschedule** option. 2. User selects a new court or time slot with **lower pricing**. 3. System calculates:  * Original Total Price * New Total Price * Refund Amount = (Original Total – New Total) – (Deposit, if deposit is non-refundable)  1. System initiates refund through the Bank API/Payment Gateway. 2. Refund is processed and confirmed. 3. System updates the booking with new details (court, time). 4. System logs the refund transaction. 5. User receives confirmation notification about the new booking and refund. | | |
| Alternative Sequences/Flows | 1. **New Court/Slot Has Higher or Equal Price:**  * No refund is processed. * If new price > old price, system requests additional payment.  1. **Refund Failure at Bank API:**  * System retries or flags for manual refund processing.  1. **New Slot No Longer Available:**  * System prompts the user to select another time or cancel reschedule request. | | |

#### 2.3.13 Refund without check in

#### 

| Primary Actors | User (indirectly), Admin (monitor only) | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables the system to **automatically detect** fully paid bookings where the user **did not check in** and the play time has passed. In such cases, the system **retains the deposit** and **refunds the remaining amount** via **MB Bank API**. This ensures fair handling of no-shows while maintaining deposit policy. | | |
| Preconditions | * Booking status = **Paid** * Check-in status = **Not Checked-in** * Current time > booking end time * MB Bank integration is configured * Booking is within eligible refund window for no-show policy | | |
| Postconditions | * Refund of (total payment - deposit) is processed via MB Bank * Booking is marked as **No-show / Refunded** * Deposit is retained as revenue * Admin and user are notified (optional) | | |
| Normal Sequence/Flow | 1. System runs **scheduled job** (e.g., every hour). 2. It filters for bookings where:  * status = paid * check-in status = not checked-in * current time > booking end time * refund\_status = pending  1. System calculates:  refund\_amount = total\_paid - deposit\_amount 2. If refund > 0, system sends refund request to **MB Bank API** with:  * User account * Amount * Booking reference  1. MB Bank returns status:  * Success → system sets refund\_status = completed * Fail → system sets refund\_status = failed and logs error  1. Booking is updated to **No-show / Refunded**, and notifications are sent. | | |
| Alternative Sequences/Flows | 1. **Refund Amount ≤ 0:**  * System logs: “No refund due – deposit retained.”  1. **MB Bank API Timeout/Error:**  * System retries; if still fails, marks status as **Refund Failed** and flags for manual review.  1. **Check-in Logged Late (After Slot):**  * System skips refund if check-in was eventually marked. | | |

#### 

#### 2.3.14 Book Court Daily

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to book a court for a specific date and time (single session booking). This is the most common type of booking for casual players who do not need a recurring reservation. The system displays available courts, times, and corresponding pricing, and allows the user to complete the reservation by paying a deposit or full amount. | | |
| Preconditions | * The user or guest is authenticated (or identified via session for Guest). * Courts are available for booking. * Booking settings for daily reservations are active. | | |
| Postconditions | * Court is reserved for the selected date and time. * Booking status is created as **Pending** until payment is completed. * Upon successful payment, booking status changes to **Confirmed**. | | |
| Normal Sequence/Flow | 1. User/Guest navigates to **Book a Court** screen. 2. User selects:  * Date * Time Slot * Court (optional filters: indoor/outdoor, location, price)  1. System shows available courts based on selection. 2. User selects a court and reviews booking details (court, date, time, price). 3. User chooses payment method (Deposit or Full Payment). 4. System generates a booking record with **Pending** status. 5. User proceeds with payment. 6. System verifies payment:  * If successful → Booking status changes to **Confirmed**. * If failed → Booking remains Pending or is canceled after timeout.  1. System sends booking confirmation notification. | | |
| Alternative Sequences/Flows | 1. **No Courts Available:**  * System displays: “No courts available for the selected date and time.”  1. **Payment Failure:**  * System prompts user to retry payment or cancel booking.  1. **Session Timeout Before Payment:**  * System automatically cancels the pending booking if payment is not completed within a configured time window (e.g., 15 minutes). | | |

#### 2.3.15 Book Court Regular

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User, Guest to book a court for multiple fixed dates and times in a recurring manner (e.g., every Monday and Wednesday at 7:00 PM for 4 weeks). It is designed for players or groups who want a consistent weekly schedule without manually booking each session. | | |
| Preconditions | * User is authenticated (Guest cannot book regular recurring slots). * Court has availability for all selected recurrence dates. * Recurring booking feature is enabled in the system settings. | | |
| Postconditions | * A series of court bookings are created based on the user's selected schedule. * Each individual booking is recorded with its own ID and linked to a main recurring group ID. * System reserves the court for all requested dates and times. | | |
| Normal Sequence/Flow | 1. User navigates to **Book a Court** and selects **Regular Booking** option. 2. User specifies:  * Start date * Days of the week (e.g., Monday, Wednesday, Friday) * Time slot (e.g., 19:00 – 20:30) * Duration (number of weeks)  1. System checks court availability for all requested dates and time slots. 2. System displays:  * Summary of selected schedule * Total price (if paying full) or deposit amount  1. User confirms booking and proceeds to payment (Deposit or Full Payment). 2. System creates multiple linked booking entries:  * Each booking assigned a unique ID. * All bookings linked under a regular booking group.  1. Payment is processed and verified. 2. System sends booking confirmation for all reserved sessions. 3. Booking status for each session is updated accordingly. | | |
| Alternative Sequences/Flows | 1. **Some Slots Unavailable:**  * System notifies the user which dates are unavailable and offers options:    + Skip unavailable dates.   + Choose alternative slots.   + Cancel the recurring booking process.  1. **Payment Failure:**  * System keeps all bookings in Pending state until payment is completed.  1. **User Cancels During Setup:**  * Booking setup is canceled without creating any entries. | | |

#### 2.3.16 Book Court Optimal

| Primary Actors | User/Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User/Guest to request a multi-day or recurring booking where the system will automatically find and assign the most optimal available courts if the User/Guest preferred court is not fully available across all dates. The system ensures that even if a specific court is fully booked on some dates, it substitutes with a similar available court so that the User/Guest can maintain a consistent playing schedule without manual rebooking. | | |
| Preconditions | * User selects the desired schedule across multiple dates. * Optimal booking feature is enabled in the system. | | |
| Postconditions | * A combination of bookings is created across different courts (if necessary). * Courts are reserved for all requested dates without manual intervention. * User/Guest is notified of any changes from their preferred court selections. | | |
| Normal Sequence/Flow | 1. User/Guest navigates to **Book a Court** and selects **Optimal Booking** option. 2. User/Guest specifies:  * Preferred court (optional) * Date range or recurring schedule (e.g., every Monday and Thursday for 6 weeks) * Preferred time slot  1. System attempts to reserve the preferred court for all selected dates. 2. If any conflicts occur:  * System automatically finds the most similar available courts based on criteria like location, indoor/outdoor, price range.  1. System generates a booking summary showing:  * Dates and courts assigned. * Indication of which dates had substitutions.  1. User/Guest reviews and confirms the booking schedule. 2. User/Guest proceeds to payment (deposit or full payment). 3. System verifies payment. 4. System finalizes and confirms all booking entries. 5. Notification is sent detailing full booking results. | | |
| Alternative Sequences/Flows | 1. **No Suitable Alternative Courts Found:**  * System alerts User/Guest and allows them to:    + Adjust schedule.   + Try a different time slot.   + Cancel the booking process.  1. **Payment Failure:**  * Bookings are held temporarily but not finalized until payment is completed.  1. **User Rejects Alternative Proposal:**  * User/Guest can cancel the entire optimal booking attempt. | | |

#### 2.3.17 Hold Court While Booking

| Primary Actors | User/Guest | Secondary Actors | Payment Gateway |
| --- | --- | --- | --- |
| Description | This use case allows the system to temporarily hold a pickleball court for a User/Guest during the booking process, ensuring the court remains reserved while the user completes payment. The hold prevents other users from booking the same court until the payment is confirmed or the hold expires. | | |
| Preconditions | * The User/Guest must have selected a specific court, date, and time slot for booking. * The selected court and time slot must be available at the time of initiating the booking. * The system must support a predefined hold duration (e.g., 10 minutes). | | |
| Postconditions | * If payment is completed successfully within the hold duration, the court is officially booked for the User/Guest . * If payment is not completed within the hold duration, the court hold is released, and the court becomes available again for other User/Guest . | | |
| Normal Sequence/Flow | 1. **Initiate Booking**: The User/Guestselects a court, date, and time slot from the available options and chooses to proceed with booking. 2. **System Holds Court**: The system temporarily reserves the selected court and time slot, marking it as "on hold" and unavailable to other User/Guest. 3. **Display Payment Prompt**: The system redirects the User/Guest to the payment interface and informs them of the hold duration (e.g., "Court held for 10 minutes"). 4. **User Completes Payment**: The user enters payment details (e.g., via an integrated payment method like MB Bank) and submits the payment. 5. **Payment Gateway Confirms**: The payment gateway processes the transaction and sends a confirmation to the system. 6. **Finalize Booking**: Upon successful payment confirmation, the system converts the temporary hold into a confirmed booking and notifies the user (e.g., via email ). 7. **Release Hold (if applicable)**: If the payment is completed within the hold duration, the process ends with the court booked. | | |
| Alternative Sequences/Flows | 1. **Payment Timeout**:  * If the user does not complete payment within the hold duration (e.g., 10 minutes) at step 4:   + - The system releases the hold on the court, making it available again for other User/Guest .     - The User/Guest receives a notification (e.g., "Your court hold has expired. Please try booking again.").     - The use case ends without a confirmed booking.  1. **Payment Failure**:  * At step 5, if the payment gateway returns a failure (e.g., insufficient funds or invalid card):   + - The system informs the User/Guest of the payment failure and provides an option to retry with a different payment method.     - If the hold duration has not expired, the User/Guest can attempt payment again, returning to step 4.     - If the hold duration expires during retries, the system releases the court, and the user is notified to start over.  1. **User Cancels Booking**:  * At any point before payment confirmation (e.g., during step 4), if the user chooses to cancel the booking:   + - The system immediately releases the court hold.     - The user is returned to the court selection screen, and the use case ends.  1. **System Error During Hold**:  * If the system fails to hold the court in step 2 (e.g., due to a technical glitch):   + - The system displays an error message (e.g., "Unable to hold the court. Please try again.").     - The user is prompted to retry the booking process from step 1. | | |

#### 

#### 

#### 2.3.19 View Reservation

| Primary Actors | User, Guest, Owner, Staff, Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case describes the process in which a user selects a specific pickleball court to view more information or proceed to booking. This is the first step in the booking process, where the user chooses a court based on location, availability, or preference. | | |
| Preconditions | * The user is logged into the system (or is a guest with limited access). * The system has active courts listed and available to view. * The user has access to the Home or Court List screen. | | |
| Postconditions | * The user is navigated to the Court Detail screen. * The selected court information is loaded and displayed. | | |
| Normal Sequence/Flow | 1. User opens the **Home Screen** or **Court Listing Page**. 2. System displays a list of available courts with filters (by name, location, availability, favorite). 3. User browses through the list and selects a specific court. 4. System navigates the user to the **Court Detail Page**. 5. System displays:  * Court name, address, phone number * Images and description * Services offered * Map link * Available booking slots | | |
| Alternative Sequences/Flows | 1. **No Courts Available:**  * System displays a message: “No courts found. Try adjusting your filters.”  1. **Court Unavailable (disabled or under maintenance):**  * System shows status (e.g., “Under Maintenance”) and prevents booking but still allows viewing.  1. **User Not Logged In (Guest):**  * System allows viewing court details but prompts login when proceeding to book.  1. **System Error / Timeout:**  * Displays an error message and offers a retry or reload option. | | |

#### 

#### 

#### 

#### 

#### 

#### 

### *2.4 Court Management*

#### 

#### 2.4.1 View Schedule occupancy analysis by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Owner to view analysis and reports on court occupancy rates based on schedules. It provides insights into how efficiently the courts are being utilized over time (by day, by time slot, or by specific courts) to help optimize operations and maximize revenue. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * Historical booking data exists in the system. * Analytics module is active and updated. | | |
| Postconditions | * Owner/Manager successfully views occupancy data visualizations. * System displays occupancy trends by court, by date, and by time slot. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Analytics → Occupancy Analysis** section. 2. User selects:  * Date range (e.g., last 7 days, this month) * Specific court(s) or all courts * Time frames (optional filter: morning, afternoon, evening)  1. System retrieves and processes booking data. 2. System displays:  * Occupancy percentage per court and per time slot. * Trends over days/weeks. * Visual charts (bar graphs, heat maps).  1. Owner reviews the report and optionally exports it (Excel, PDF). | | |
| Alternative Sequences/Flows | 1. **No Data Available:**  * System displays: “No occupancy data found for the selected period.”  1. **Export Failure:**  * System shows error: “Unable to export report. Please try again later.” | | |

#### 2.4.2 View Schedule occupancy analysis by Manager

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Manager to view analysis and reports on court occupancy rates based on schedules. It provides insights into how efficiently the courts are being utilized over time (by day, by time slot, or by specific courts) to help optimize operations and maximize revenue under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * Historical booking data exists in the system. * Analytics module is active and updated. | | |
| Postconditions | * Manager successfully views occupancy data visualizations. * System displays occupancy trends by court, by date, and by time slot. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Analytics → Occupancy Analysis** section. 2. User selects:  * Date range (e.g., last 7 days, this month) * Specific court(s) or all courts * Time frames (optional filter: morning, afternoon, evening)  1. System retrieves and processes booking data. 2. System displays:  * Occupancy percentage per court and per time slot. * Trends over days/weeks. * Visual charts (bar graphs, heat maps).  1. Manager reviews the report and optionally exports it (Excel, PDF). | | |
| Alternative Sequences/Flows | 1. **No Data Available:**  * System displays: “No occupancy data found for the selected period.”  1. **Export Failure:**  * System shows error: “Unable to export report. Please try again later.” | | |

#### 

#### 2.4.3 View Peak hours analysis by Owner

| Primary Actors | Owner | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to analyze **peak booking hours** across all courts or specific locations. The system provides visual insights such as bar charts to show which hours of the day (and which days of the week) have the highest demand, helping facilities optimize pricing, staff allocation, and slot availability. | | |
| Preconditions | * The actor is authenticated as **Owner** * Booking data is available for the selected timeframe. * The analytics module is functioning properly. | | |
| Postconditions | * A visual analysis of peak hours is shown. * Actors can identify busiest time slots by day/hour/court. * The data can be exported for further planning. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Analytics**. 3. Selects filters:  * Time range * Court Name  1. System processes booking data. 2. Displays analytics including:  * Hourly booking frequency * Most and least booked time slots * Bar Chart by day/hour * Comparison with previous periods (optional)  1. Actor uses this data to:  * Optimize pricing (e.g., dynamic pricing for peak hours) * Reallocate staff during busy times * Offer promotions for off-peak hours  1. Optionally clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Booking Data:**  * System displays: “No data ”  1. **Chart Generation Error:**  * System falls back to table view and logs the error.  1. **No Distinct Peak Patterns:**  * System highlights low variation and suggests broader time range. | | |

#### 2.4.4 View Peak hours analysis by Manager

| Primary Actors | Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to analyze **peak booking hours** across all courts or specific locations. The system provides visual insights such as bar charts to show which hours of the day (and which days of the week) have the highest demand, helping facilities optimize pricing, staff allocation, and slot availability under the Owner's authority | | |
| Preconditions | * The actor is authenticated as **Manager**. * Booking data is available for the selected timeframe. * The analytics module is functioning properly. | | |
| Postconditions | * A visual analysis of peak hours is shown. * Actor can identify busiest time slots by day/hour/court. * The data can be exported for further planning. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Analytics**. 3. Selects filters:  * Time range * Court Name  1. System processes booking data. 2. Displays analytics including:  * Hourly booking frequency * Most and least booked time slots * Bar Chart by day/hour * Comparison with previous periods (optional)  1. Actor uses this data to:  * Optimize pricing (e.g., dynamic pricing for peak hours) * Reallocate staff during busy times * Offer promotions for off-peak hours  1. Optionally clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Booking Data:**  * System displays: “No data ”  1. **Chart Generation Error:**  * System falls back to table view and logs the error.  1. **No Distinct Peak Patterns:**  * System highlights low variation and suggests broader time range. | | |

#### 2.4.5 Create Court

| Primary Actors | Owner | Secondary Actors | System |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to add a new court to the system, including details such as court name location and email, location, website, open time. The court will then be available for booking by users. | | |
| Preconditions | * The actor is logged in with the **Owner** role. * The court management module is active and accessible. * Required fields (e.g., court name, type) are filled in. | | |
| Postconditions | * A new court is added to the database. * The court is visible in the **booking interface** for users. * The court appears in the **court management dashboard** for future edits or updates. | | |
| Normal Sequence/Flow | 1. Owner/Manager logs in to the admin dashboard. 2. Navigates to **Court Management → Add New Court**. 3. Fills in required details:  * Court Name * Address * Phone Number * Open Time * Email  1. Defines:  * Status (Active/Inactive)  1. Clicks **“Save”**. 2. System validates input and creates court record. 3. Confirmation message is shown: "Success." | | |
| Alternative Sequences/Flows | 1. **Required Field Missing:**  * System prompts: "Please complete all required fields."  1. **System Error on Save:**  * Error message displayed; user may retry. | | |

#### 2.4.6 Update Court

| Primary Actors | Owner | Secondary Actors | System |
| --- | --- | --- | --- |
| Description | This use case allows an Owner to update the information of an existing court, including details like court name, type (indoor/outdoor), surface type, pricing, operational status, and maintenance notes. Keeping court information up to date ensures accurate booking management and player communication. | | |
| Preconditions | * The user is authenticated and has the role of **Owner** * The court to be updated already exists in the system. * Court update feature is available and the court is not locked by active critical transactions (e.g., booking modification in process). | | |
| Postconditions | * The selected court’s information is updated in the system. * Changes are reflected immediately in court listings and booking screens. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Management → Court List**. 2. User selects a specific court to update. 3. System displays current court details (name, type, description, pricing, availability, etc.). 4. Owner edits the desired fields (e.g., updates court type to indoor, changes pricing). 5. User saves the changes. 6. System validates the updated information. 7. System updates court record in the database. 8. System confirms the update with a success message. | | |
| Alternative Sequences/Flows | 1. **Validation Error (Missing Fields/Invalid Data):**  * System highlights errors and requests correction (e.g., invalid price format, missing court name).  1. **Update Failure (Database Error):**  * System displays error message: “Unable to save changes. Please try again later.”  1. **Court Not Found or Locked:**  * System notifies: “This court cannot be updated at the moment.” | | |

#### 2.4.7 Active Court

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to activate a court that was previously set as **inactive**, making it visible and available for booking by users. This is commonly used after court maintenance or administrative review. | | |
| Preconditions | * The actor is authenticated as **Owner**. * The court exists in the system and is currently set to **Inactive**. * The court has required configurations (e.g., time slots, pricing). | | |
| Postconditions | * The court status is changed to **Active**. * The court becomes visible in the user-facing interface. * The court is now bookable. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Court Management** 3. Clicks on the **"Edit"** button. 4. Clicks on the **"Activate"** button for a specific court. 5. Clicks **“Save”**. 6. System validates changes and updates the database. 7. Success message is shown: "Success." | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Confirmation:**  * No changes are made.  1. **System Error on Status Update:**  * Error message is shown: “Fail.” | | |

#### 

#### 

#### 

#### 2.4.8 Deactivate Court

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to deactivate a court, making it unavailable for user bookings. This action is typically taken when the court is under maintenance, unavailable due to events, or retired from regular use. Deactivated courts are not visible to users but remain in the system for editing or future reactivation. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The court exists in the system and is currently marked as **Active**. * The court is not tied to any pending or ongoing bookings. | | |
| Postconditions | * The court status is set to **Inactive**. * The court is hidden from the user booking interface. * The court remains accessible in admin view for editing or reactivation. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Court Management** 3. Clicks on the **"Edit"** button. 4. Clicks on the **"Deactivate"** button for a specific court. 5. Clicks **“Save”**. 6. System validates changes and updates the database. 7. Success message is shown: "Success." | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Action:**  * No changes are made.  1. **System Error During Update:**  * Error message is shown: “Fail.” | | |

#### 

#### 2.4.9 Create Court Slot by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to define available **time slots** for each court, including start/end times, duration, and availability rules (e.g., day of week, custom pricing). Once created, these time slots appear in the booking calendar for users. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The time slot module is active and accessible. | | |
| Postconditions | * New time slots are associated with a court and stored in the database. * These time slots appear on the **booking interface** for users. * Slot availability is ready for pricing and status updates. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Sub**-**Court Management** Clicks **“Add Time Slot”**. 3. Fills in:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Add”**. 2. System validates time range and overlap. 3. Slot is saved and visible in the time slot list for that court. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.10 Create Court Slot by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Manager** to define available **time slots** for each court, including start/end times, duration, and availability rules (e.g., day of week, custom pricing). Once created, these time slots appear in the booking calendar for users under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * The time slot module is active and accessible. | | |
| Postconditions | * New time slots are associated with a court and stored in the database. * These time slots appear on the **booking interface** for users. * Slot availability is ready for pricing and status updates. | | |
| Normal Sequence/Flow | 1. Manager logs into the admin dashboard. 2. Navigates to **Sub**-**Court Management** Clicks **“Add Time Slot”**. 3. Fills in:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Add”**. 2. System validates time range and overlap. 3. Slot is saved and visible in the time slot list for that court. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.11 Update Court Slot by Owner

| Primary Actors | Owner, | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to update information related to a specific court slot, including its start time, end time, price, availability status, or applicable days. Updates are reflected immediately and can affect upcoming booking availability. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * At least one court and one time slot already exist in the system. * The slot to be updated is not locked by an active or confirmed booking (depending on system policy). | | |
| Postconditions | * The selected time slot is updated with new configurations. * The updated slot is reflected in the booking interface (if still active). * Booking conflicts (if any) are handled according to system rules. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. Selects an existing slot and clicks **“Edit”**. 4. Updates one or more fields:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Update”**. 2. System validates updated time range and potential overlaps. 3. Changes are saved and confirmation is displayed. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.12 Update Court Slot by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to update information related to a specific court slot, including its start time, end time, price, availability status, or applicable days. Updates are reflected immediately and can affect upcoming booking availability under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * At least one court and one time slot already exist in the system. * The slot to be updated is not locked by an active or confirmed booking (depending on system policy). | | |
| Postconditions | * The selected time slot is updated with new configurations. * The updated slot is reflected in the booking interface (if still active). * Booking conflicts (if any) are handled according to system rules. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. Selects an existing slot and clicks **“Edit”**. 4. Updates one or more fields:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Update”**. 2. System validates updated time range and potential overlaps. 3. Changes are saved and confirmation is displayed. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.13 Active Sub-Court by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **activate a sub-court** (a smaller unit or division of a main court) to make it available for booking. A sub-court may be deactivated temporarily for maintenance or scheduling purposes, and activating it again restores its visibility and bookability for users. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The sub-court exists in the system and is currently in an **Inactive** state. * No conflicts exist (e.g., sub-court not under ongoing maintenance or reserved for exclusive events). | | |
| Postconditions | * The sub-court is now marked as **Active**. * The sub-court becomes available for booking immediately or based on an effective date. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Inactive Sub-Courts**. 3. User selects a sub-court to activate. 4. System displays the current status and related information. 5. User clicks **Activate** button. 6. System updates the sub-court status to **Active**. 7. Sub-court becomes available on the booking platform. 8. System logs the activation action. | | |
| Alternative Sequences/Flows | 1. **Validation Error:**  * System prevents activation if maintenance is ongoing or scheduled. * System shows warning: “Sub-court cannot be activated due to ongoing maintenance.”  1. **Update Failure (Database Issue):**  * System displays: “Activation failed. Please try again later.” | | |

#### 2.4.14 Active Sub-Court by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Manager** to **activate a sub-court** (a smaller unit or division of a main court) to make it available for booking. A sub-court may be deactivated temporarily for maintenance or scheduling purposes, and activating it again restores its visibility and bookability for users under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The sub-court exists in the system and is currently in an **Inactive** state. * No conflicts exist (e.g., sub-court not under ongoing maintenance or reserved for exclusive events). | | |
| Postconditions | * The sub-court is now marked as **Active**. * The sub-court becomes available for booking immediately or based on an effective date. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Inactive Sub-Courts**. 3. User selects a sub-court to activate. 4. System displays the current status and related information. 5. User clicks **Activate** button. 6. System updates the sub-court status to **Active**. 7. Sub-court becomes available on the booking platform. 8. System logs the activation action. | | |
| Alternative Sequences/Flows | 1. **Validation Error:**  * System prevents activation if maintenance is ongoing or scheduled. * System shows warning: “Sub-court cannot be activated due to ongoing maintenance.”  1. **Update Failure (Database Issue):**  * System displays: “Activation failed. Please try again later.” | | |

#### 2.4.15 Deactivate Sub-Court by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **deactivate a sub-court**, temporarily removing it from the booking system. This can be necessary for reasons such as scheduled maintenance, operational adjustments, or special events requiring full-court usage. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The sub-court exists in the system and is currently in an **Active** state. * No active bookings conflict with the deactivation timeframe (or a rescheduling/cancellation policy is applied). | | |
| Postconditions | * The selected sub-court is updated to **Inactive** status. * The sub-court is no longer available for users to book. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Active Sub-Courts**. 3. User selects the sub-court to deactivate. 4. System displays the current status and sub-court details. 5. User clicks the **Deactivate** button and confirms the action. 6. System checks for upcoming bookings:  * If none → Proceed. * If existing bookings → Prompt user to handle them (cancel or reschedule).  1. System updates sub-court status to **Inactive**. 2. System logs the deactivation action. 3. System sends notification if needed (to affected users, if applicable). | | |
| Alternative Sequences/Flows | 1. **Existing Bookings Conflict:**  * System blocks deactivation and shows warning:  “Sub-court has active bookings. Please resolve bookings before deactivation.”  1. **Deactivation Aborted by User:**  * If user cancels at confirmation step, no changes are made.  1. **Database Error:**  * System displays: “Unable to deactivate sub-court. Please try again later.” | | |

#### 2.4.16 Deactivate Sub-Court by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Manager** to **deactivate a sub-court**, temporarily removing it from the booking system. This can be necessary for reasons such as scheduled maintenance, operational adjustments, or special events requiring full-court usage under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The sub-court exists in the system and is currently in an **Active** state. * No active bookings conflict with the deactivation timeframe (or a rescheduling/cancellation policy is applied). | | |
| Postconditions | * The selected sub-court is updated to **Inactive** status. * The sub-court is no longer available for users to book. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Active Sub-Courts**. 3. User selects the sub-court to deactivate. 4. System displays the current status and sub-court details. 5. User clicks the **Deactivate** button and confirms the action. 6. System checks for upcoming bookings:  * If none → Proceed. * If existing bookings → Prompt user to handle them (cancel or reschedule).  1. System updates sub-court status to **Inactive**. 2. System logs the deactivation action. 3. System sends notification if needed (to affected users, if applicable). | | |
| Alternative Sequences/Flows | 1. **Existing Bookings Conflict:**  * System blocks deactivation and shows warning:  “Sub-court has active bookings. Please resolve bookings before deactivation.”  1. **Deactivation Aborted by User:**  * If user cancels at confirmation step, no changes are made.  1. **Database Error:**  * System displays: “Unable to deactivate sub-court. Please try again later.” | | |

#### 2.4.17 Create Maintenance by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to schedule maintenance periods for specific courts or court areas. During the maintenance window, the court will be **unavailable for booking**. The maintenance record includes a description, duration, and affected court(s), and ensures proper scheduling and communication to users and staff. | | |
| Preconditions | * The actor is authenticated as **Owner**. * At least one court exists in the system. * The maintenance scheduling module is available. | | |
| Postconditions | * A maintenance entry is created and associated with one or more courts. * The court becomes unavailable for booking during the maintenance window. * Users are prevented from booking and (optionally) receive notification of closure. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Maintenance**. 3. Select  * Select court * Select sub-court  1. Fills in the maintenance form:  * Start date & time * End date & time * Description  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Confirmation message is shown: “Tạo lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **System Error on Save:**  * System shows: “Tạo lịch bảo trì thất bại.” | | |

#### 2.4.18 Create Maintenance by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to schedule maintenance periods for specific courts or court areas. During the maintenance window, the court will be **unavailable for booking**. The maintenance record includes a description, duration, and affected court(s), and ensures proper scheduling and communication to users and staff under the Owner's authority | | |
| Preconditions | * The actor is authenticated as **Manager**. * At least one court exists in the system. * The maintenance scheduling module is available. | | |
| Postconditions | * A maintenance entry is created and associated with one or more courts. * The court becomes unavailable for booking during the maintenance window. * Users are prevented from booking and (optionally) receive notification of closure. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Maintenance**. 3. Select  * Select court * Select sub-court  1. Fills in the maintenance form:  * Start date & time * End date & time * Description  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Confirmation message is shown: “Tạo lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **System Error on Save:**  * System shows: “Tạo lịch bảo trì thất bại.” | | |

#### 2.4.19 Update Maintenance by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to modify an existing maintenance record for a court, including changes to time, description, or affected court area. Updating maintenance ensures accuracy and allows for changes due to scheduling conflicts or work delays. | | |
| Preconditions | * The actor is authenticated as **Owner** * At least one maintenance entry exists in the system. * The maintenance record has not yet expired (optional policy). | | |
| Postconditions | * The maintenance entry is updated with new details. * The booking system reflects the new maintenance window. * Users are blocked from booking during the updated period. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → Maintenance List**. 3. Select  * Select court * Select sub-court  1. Click”Edit” 2. Fills in the maintenance form:  * Start date & time * End date & time * Description * Status  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Success message is shown: “Cập nhật lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **Save Error:**  * System displays error message and allows retry. | | |

#### 2.4.20 Update Maintenance by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to modify an existing maintenance record for a court, including changes to time, description, or affected court area. Updating maintenance ensures accuracy and allows for changes due to scheduling conflicts or work delays under the Owner's authority | | |
| Preconditions | * The actor is authenticated as **Manager**. * At least one maintenance entry exists in the system. * The maintenance record has not yet expired (optional policy). | | |
| Postconditions | * The maintenance entry is updated with new details. * The booking system reflects the new maintenance window. * Users are blocked from booking during the updated period. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → Maintenance List**. 3. Select  * Select court * Select sub-court  1. Click”Edit” 2. Fills in the maintenance form:  * Start date & time * End date & time * Description * Status  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Success message is shown: “Cập nhật lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **Save Error:**  * System displays error message and allows retry. | | |

#### 2.4.21 View Maintenance by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to view a list of all scheduled and completed maintenance records for courts or court areas. Maintenance entries include time period, status, description, and affected courts. This supports operational visibility and planning. | | |
| Preconditions | * The actor is logged in with **Owner** role. * At least one court maintenance record exists in the system. | | |
| Postconditions | * A list of maintenance records is displayed. * Each record can be filtered, searched, and optionally expanded for detail. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → View Maintenance**. 3. System displays a list or table of maintenance entries:  * Select court * Select sub-court | | |
| Alternative Sequences/Flows | 1. **No Maintenance Found:**  * System shows: “Trống.”  1. **Entry Load Failure:**  * Error message shown with retry option. | | |

#### 2.4.22 View Maintenance by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to view a list of all scheduled and completed maintenance records for courts or court areas. Maintenance entries include time period, status, description, and affected courts. This supports operational visibility and planning under the Owner's authority | | |
| Preconditions | * The actor is logged in with **Manager** role. * At least one court maintenance record exists in the system. | | |
| Postconditions | * A list of maintenance records is displayed. * Each record can be filtered, searched, and optionally expanded for detail. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → View Maintenance**. 3. System displays a list or table of maintenance entries:  * Select court * Select sub-court | | |
| Alternative Sequences/Flows | 1. **No Maintenance Found:**  * System shows: “Trống.”  1. **Entry Load Failure:**  * Error message shown with retry option. | | |

#### 2.4.23 Create Court Image by Owner

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to upload and assign one or more images to a specific court. These images will be displayed on the booking interface, giving users a visual preview of the court (e.g., surface, lighting, surroundings). | | |
| Preconditions | * The actor is authenticated with the **Owner** role. * The court already exists in the system. * The image file meets system format and size requirements. | | |
| Postconditions | * One or more images are associated with a court. * Images are visible to users when viewing or booking the court. * Images can be managed (edited or deleted) later. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court → Manage Images**. 3. Clicks **“Select File”**. 4. Selects image file(s) from local device. 5. Clicks **“Upload”**. 6. System validates the file (format, size) and associates it with the selected court. 7. Confirmation message is displayed:  “Create Successfully” | | |
| Alternative Sequences/Flows | 1. **Upload Failure:**  * System logs error and shows: “Fail.” | | |

#### 

#### 2.4.24 Create Court Image by Manager

#### 

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to upload and assign one or more images to a specific court. These images will be displayed on the booking interface, giving users a visual preview of the court (e.g., surface, lighting, surroundings) under the Owner's authority | | |
| Preconditions | * The actor is authenticated with the **Manager** role. * The court already exists in the system. * The image file meets system format and size requirements. | | |
| Postconditions | * One or more images are associated with a court. * Images are visible to users when viewing or booking the court. * Images can be managed (edited or deleted) later. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court → Manage Images**. 3. Clicks **“Select File”**. 4. Selects image file(s) from local device. 5. Clicks **“Upload”**. 6. System validates the file (format, size) and associates it with the selected court. 7. Confirmation message is displayed:  “Create Successfully” | | |
| Alternative Sequences/Flows | 1. **Upload Failure:**  * System logs error and shows: “Fail.” | | |

#### 2.4.25 Delete Court Image by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts. | | |
| Preconditions | * The actor is authenticated as **Owner** * The image exists and is currently associated with a valid court. * The image is **not the only image** remaining if court requires at least one display image. | | |
| Postconditions | * The selected image is deleted from the system. * The court no longer displays the deleted image in the booking interface. * If the image was marked as default, the system may assign another default automatically. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court** 3. Views image gallery and clicks **“Delete”** on the chosen image. 4. System prompts:  “Are you sure you want to delete this item? ” 5. Actor confirms deletion. 6. System removes the image from storage and updates the database. 7. Success message is displayed:  “ Deleted successfully.” | | |
| Alternative Sequences/Flows | 1. **Deletion Error:**  * System displays: “Failed to delete image. Please try again.” | | |

#### 2.4.26 Delete Court Image by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts. | | |
| Preconditions | * The actor is authenticated as **Manager**. * The image exists and is currently associated with a valid court. * The image is **not the only image** remaining if court requires at least one display image. | | |
| Postconditions | * The selected image is deleted from the system. * The court no longer displays the deleted image in the booking interface. * If the image was marked as default, the system may assign another default automatically. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court** 3. Views image gallery and clicks **“Delete”** on the chosen image. 4. System prompts:  “Are you sure you want to delete this item? ” 5. Actor confirms deletion. 6. System removes the image from storage and updates the database. 7. Success message is displayed:  “ Deleted successfully.” | | |
| Alternative Sequences/Flows | 1. **Deletion Error:**  * System displays: “Failed to delete image. Please try again.” | | |

#### 2.4.27 View Court Image by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to view images. The court image provides visual confirmation and allows the court Owner to be able to upload the latest court image of the field to the user visually. | | |
| Preconditions | * The court exists in the system. * The system must be allow Owner have permission to upload image about court | | |
| Postconditions | * The court images are displayed successfully. * Owner can browse through all uploaded images for the court. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Image** page. 2. Owner selects a specific court to view more details. 3. System retrieves all uploaded images for the selected court. 4. System displays images in a gallery or carousel view:  * Thumbnails * Full-size images (on click/tap)  1. Owner browses through available court images. | | |
| Alternative Sequences/Flows | 1. **No Images Uploaded for Court:**  * System displays placeholder image and message:  “No images available for this court yet.”  1. **Image Load Failure:**  * System shows error message: “Unable to load court images. Please try again later.” | | |

#### 2.4.28 View Court Image by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to view images. The court image provides visual confirmation and allows the court Manager to be able to upload the latest court image of the field to the user visually. | | |
| Preconditions | * The court exists in the system. * The system must be allow Manager have permission to upload image about court | | |
| Postconditions | * The court images are displayed successfully. * Manager can browse through all uploaded images for the court. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Court Image** page. 2. Manager selects a specific court to view more details. 3. System retrieves all uploaded images for the selected court. 4. System displays images in a gallery or carousel view:  * Thumbnails * Full-size images (on click/tap)  1. Manager browses through available court images. | | |
| Alternative Sequences/Flows | * **No Images Uploaded for Court:** * System displays placeholder image and message:  “No images available for this court yet.” * **Image Load Failure:** * System shows error message: “Unable to load court images. Please try again later.” | | |

#### 2.4.29 View Court Details

| Primary Actors | User/Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows User/Guest to view detailed information about a pickleball court, including its specifications (e.g., type, location, availability), user reviews, and images, to help them make an informed decision before booking. | | |
| Preconditions | * The system must have at least one court registered with available details, reviews, or images. * The User/Guest must have access to the court listing section of the system (guest and user can additional features like saved preferences). | | |
| Postconditions | The User/Guest has viewed the court’s details, reviews, and images, and can proceed to book the court or explore other options. | | |
| Normal Sequence/Flow | 1. **Access Court Listings**: The User/Guest has viewed the court’s details, reviews, and images, and can proceed to book the court or explore other options. 2. **Select a Court**: The User/Guest selects a specific court from the available list. 3. **Display Court Details**: The system displays detailed information about the selected court, location, pricing by time slot, and availability 4. **Show Reviews**: The system presents user-generated reviews and ratings for the court (if available). 5. **Show Images**: The system displays images of the court (e.g., uploaded by the owner or staff) for the user to view. 6. **Interact with Content**: The user scrolls through the details, and view images to evaluate the court. | | |
| Alternative Sequences/Flows | 1. **No Images Available**:  * At step 5, if no images have been uploaded for the court:   + - The system displays a placeholder image or a message like "No images available at this time."     - The use case continues with step 6.  1. **Court Not Found**:  * At step 2, if the User/Guest selects a court that no longer exists or is unavailable:   + - The system displays an error message (e.g., "This court is not available. Please select another.")     - The user is redirected to the court listings to choose another court, returning to step 1.  1. **User Proceeds to Booking**:  * After step 6, if the User/Guest decides to book the court:   + - The system transitions the user to the booking process (a separate use case).     - The current use case ends.  1. **User Exits Without Action**:  * At any point, if the User/Guest chooses to exit the court details view:   + - The system returns the user to the court listings or homepage.     - The use case ends. | | |

#### 

#### 2.4.30 View Court

| Primary Actors | User, Guest, Owner, Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case allows **Users** and **Guests** to view available courts for booking, along with relevant details such as court type, location, available time slots, price, and court images. **Owners** and **Managers** can view all courts with additional management details, including maintenance status and booking statistics. | | |
| Preconditions | * The system has at least one court registered and marked as **active**. * The actor is either a guest, user, manager, or owner. * The court list or detail view is accessible from the home screen or management dashboard. | | |
| Postconditions | * The actor can view court information. * Court status (e.g., available, under maintenance) is shown. * The user may proceed to **booking**, while admin may go to **edit/delete**. | | |
| Normal Sequence/Flow | **For User/Guest:**   1. Actor accesses **Home Page** or **Court Listing**. 2. System displays available courts with:  * Court name and type (indoor/outdoor) * Location with map link * Price per slot/hour * Available time slots * Description and photos * “Book Now” or “View Details” button  1. Actor clicks on a specific court for more info. 2. System shows full details and booking calendar.   **For Owner/Manager:**   1. Actor logs into admin dashboard. 2. Navigates to **Court Management → Court List**. 3. System shows all courts, including:  * Booking stats * Status (active/inactive, under maintenance) * Edit/Delete buttons * Maintenance logs (if any) | | |
| Alternative Sequences/Flows | 1. **No Active Courts Found:**  * System displays: “No courts currently available.”  1. **Court Info Load Failure:**  * System shows an error message and retry option. | | |

#### 

#### 2.4.31 View Court Slot

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Users/Guests** to view available **court time slots** for booking. Time slots show detailed scheduling, availability, and pricing information. | | |
| Preconditions | * At least one court exists with assigned time slots. * **User/Guest** viewing available slots for booking | | |
| Postconditions | A list or calendar view of available (and optionally unavailable) court slots is displayed. | | |
| Normal Sequence/Flow | 1. Navigates to **Home → Select Court → View Schedule**. 2. System shows:  * Date picker or calendar view * List of available time slots (e.g., 06:00–07:00, 07:00–08:00) * Status (Available / Booked / Maintenance) * Price per slot  1. User may click on a slot to begin booking (if available). | | |
| Alternative Sequences/Flows | **No Slots Defined for Court:**   * System shows: “No time slots available for this court.”   **Slot Load Failure (System Error):**   * Error message is shown and retry option provided. | | |

#### 2.4.32 View Court Slot by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** can view the full list of time slots associated with each court, including inactive, booked, or under maintenance slots. Time slots show detailed scheduling, availability, and pricing information. | | |
| Preconditions | **Owner** accessing time slot configuration for a court | | |
| Postconditions | Actor may proceed to booking, editing, or analysis | | |
| Normal Sequence/Flow | 1. Logs into **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. System displays:  * Full list of all defined slots * Status: Active / Inactive / Fully booked * Price, recurrence pattern (e.g., every Monday) * Action buttons (Edit / Deactivate / Delete) | | |
| Alternative Sequences/Flows | **No Slots Defined for Court:**   * System shows: “No time slots available for this court.”   **Slot Load Failure (System Error):**   * Error message is shown and retry option provided. | | |

#### 2.4.33 View Court Slot by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** can view the full list of time slots associated with each court, including inactive, booked, or under maintenance slots. Time slots show detailed scheduling, availability, and pricing information under the Owner's authority | | |
| Preconditions | **Manager** accessing time slot configuration for a court | | |
| Postconditions | Actor may proceed to booking, editing, or analysis | | |
| Normal Sequence/Flow | 1. Logs into **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. System displays:  * Full list of all defined slots * Status: Active / Inactive / Fully booked * Price, recurrence pattern (e.g., every Monday) * Action buttons (Edit / Deactivate / Delete) | | |
| Alternative Sequences/Flows | **No Slots Defined for Court:**   * System shows: “No time slots available for this court.”   **Slot Load Failure (System Error):**   * Error message is shown and retry option provided. | | |

#### 

### *2.5 Service Management*

#### 2.5.1 Create Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to create new services that will be available for users to purchase. Services can include food and drinks, equipment rental, or coaching sessions. Each service includes a name, description, price, status, and optional images or categories. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The court exists in the system and is currently marked as **Active**. * The court is not tied to any pending or ongoing bookings. | | |
| Postconditions | * The court status is set to **Inactive**. * The court is hidden from the user booking interface. * The court remains accessible in admin view for editing or reactivation. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks on the **“Thêm dịch vụ mới”**. 4. Fills in:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”** 2. System displays:  “Tạo mới thành công .” | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Action:**  * No changes are made.  1. **System Error During Create:**  * Error is shown and action is rolled back. | | |

#### 2.5.2 Create Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to create new services that will be available for users to purchase. Services can include food and drinks, equipment rental, or coaching sessions. Each service includes a name, description, price, status, and optional images or categories under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * The court exists in the system and is currently marked as **Active**. * The court is not tied to any pending or ongoing bookings. | | |
| Postconditions | * The court status is set to **Inactive**. * The court is hidden from the user booking interface. * The court remains accessible in admin view for editing or reactivation. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks on the **“Thêm dịch vụ mới”**. 4. Fills in:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”** 2. System displays:  “Tạo mới thành công .” | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Action:**  * No changes are made.  1. **System Error During Create:**  * Error is shown and action is rolled back. | | |

#### 2.5.3 Active Status Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **activate a service** (such as water, snack, paddle rental, etc.) so that it becomes available for users to view and purchase through the booking system or service order screens. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The service exists in the system and is currently in an **Inactive** status. * Service information (name, price, description) is fully configured. | | |
| Postconditions | * The selected service is set to **Active** status. * Users and guests can now view and purchase the service during booking or through service ordering. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Service Management → Service List**. 2. User filters or searches for **Inactive Services**. 3. User selects a service they wish to activate. 4. System displays service details for review. 5. Owner clicks the **Activate** button. 6. System updates the service status to **Active**. 7. Service becomes visible and purchasable for users immediately. 8. System logs the action for audit purposes. | | |
| Alternative Sequences/Flows | 1. **Service Validation Fails (Missing Critical Data):**  * System prevents activation and displays an error:  “Cannot activate service. Please complete all required information (e.g., pricing, description).”  1. **Database Error on Update:**  * System displays: “Activation failed. Please try again later.” | | |

#### 

#### 

#### 2.5.4 Active Status Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Manager** to **activate a service** (such as water, snack, paddle rental, etc.) so that it becomes available for users to view and purchase through the booking system or service order screens under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The service exists in the system and is currently in an **Inactive** status. * Service information (name, price, description) is fully configured. | | |
| Postconditions | * The selected service is set to **Active** status. * Users and guests can now view and purchase the service during booking or through service ordering. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Service Management → Service List**. 2. User filters or searches for **Inactive Services**. 3. User selects a service they wish to activate. 4. System displays service details for review. 5. Manager clicks the **Activate** button. 6. System updates the service status to **Active**. 7. Service becomes visible and purchasable for users immediately. 8. System logs the action for audit purposes. | | |
| Alternative Sequences/Flows | 1. **Service Validation Fails (Missing Critical Data):**  * System prevents activation and displays an error:  “Cannot activate service. Please complete all required information (e.g., pricing, description).”  1. **Database Error on Update:**  * System displays: “Activation failed. Please try again later.” | | |

#### 2.5.5 Deactive Status Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **deactivate a service** (such as bottled water, paddle rental, snacks, etc.) so that it is temporarily **unavailable** for users to view and purchase. Deactivation may be needed during out-of-stock situations, maintenance periods, or promotional adjustments. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The service currently exists in the system and is in an **Active** status. * No critical transactions are relying on the immediate availability of the service (e.g., pending service orders can still be fulfilled). | | |
| Postconditions | * The selected service is updated to **Inactive** status. * The service is hidden from users and cannot be selected for new bookings or service orders. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Service Management → Service List**. 2. User filters or searches for **Active Services**. 3. User selects the service to deactivate. 4. System displays the service’s current details. 5. Owner clicks the **Deactivate** button. 6. System updates the service status to **Inactive**. 7. Service is removed from the user-facing booking and ordering interfaces. 8. System logs the deactivation action for auditing. | | |
| Alternative Sequences/Flows | 1. **Service in Use:**  * If the service is currently included in active unpaid orders, system warns the user:  “This service is included in ongoing transactions. Please confirm before deactivation.”  1. **Database Save Error:**  * System displays: “Unable to deactivate service. Please try again later.**”** | | |

#### 2.5.6 Deactive Status Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** or **Manager** to **deactivate a service** (such as bottled water, paddle rental, snacks, etc.) so that it is temporarily **unavailable** for users to view and purchase. Deactivation may be needed during out-of-stock situations, maintenance periods, or promotional adjustments under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The service currently exists in the system and is in an **Active** status. * No critical transactions are relying on the immediate availability of the service (e.g., pending service orders can still be fulfilled). | | |
| Postconditions | * The selected service is updated to **Inactive** status. * The service is hidden from users and cannot be selected for new bookings or service orders. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Service Management → Service List**. 2. User filters or searches for **Active Services**. 3. User selects the service to deactivate. 4. System displays the service’s current details. 5. Manager clicks the **Deactivate** button. 6. System updates the service status to **Inactive**. 7. Service is removed from the user-facing booking and ordering interfaces. 8. System logs the deactivation action for auditing. | | |
| Alternative Sequences/Flows | 1. **Service in Use:**  * If the service is currently included in active unpaid orders, system warns the user:  “This service is included in ongoing transactions. Please confirm before deactivation.”  1. **Database Save Error:**  * System displays: “Unable to deactivate service. Please try again later.**”** | | |

#### 2.5.7 Update Service by Owner

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Manager** to edit details of an existing service (e.g., equipment rental, snacks, water, coaching). Updates can include changes to the name, price, description, availability status, category, or associated image. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The service exists in the system. * The actor has permission to manage services. | | |
| Postconditions | * The selected service is updated with the new information. * Changes are saved to the database and visible to users if the service is active. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks **“Edit”** next to the desired service. 4. System displays a form pre-filled with current service details. 5. Actor modifies one or more fields:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”**. 2. System validates changes and updates the service record. 3. Confirmation message is shown:  “Cập nhật thành công” | | |
| Alternative Sequences/Flows | 1. **Invalid Input (e.g., empty name or invalid price):**  * System highlights error and requests correction.  1. **System Error on Save:**  * Displays error message and logs failure. | | |

#### 

#### 

#### 2.5.8 Update Service by Manager

#### 

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Manager** to edit details of an existing service (e.g., equipment rental, snacks, water, coaching). Updates can include changes to the name, price, description, availability status, category, or associated image under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * The service exists in the system. * The actor has permission to manage services. | | |
| Postconditions | * The selected service is updated with the new information. * Changes are saved to the database and visible to users if the service is active. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks **“Edit”** next to the desired service. 4. System displays a form pre-filled with current service details. 5. Actor modifies one or more fields:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”**. 2. System validates changes and updates the service record. 3. Confirmation message is shown:  “Cập nhật thành công” | | |
| Alternative Sequences/Flows | 1. **Invalid Input (e.g., empty name or invalid price):**  * System highlights error and requests correction.  1. **System Error on Save:**  * Displays error message and logs failure. | | |

#### 2.5.9 View List of Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | **Owners** can view all services in the system, including inactive or archived ones, along with management data like status, price, and edit options. | | |
| Preconditions | * Services have been created and saved in the system. * All services (active/inactive) are accessible. | | |
| Postconditions | * A list of services is displayed. * Actors can view full service details. * Owner may **edit/delete/deactivate**. | | |
| Normal Sequence/Flow | Actor logs into the **admin dashboard**.   1. Navigates to **Service Management**. 2. System shows:  * Full service list (Active/Inactive) * Name, Price, Status, Edit/Delete buttons * Sorting, search, and filtering by category or status  1. Actor clicks on a service to see detailed view or modify it. | | |
| Alternative Sequences/Flows | * **No Services Available:** * System shows: “No services currently available.” * **Service Not Found or Deleted:** * System displays error message: “Service does not exist.” | | |

#### 

#### 2.5.10 View List of Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | **Manager** can view all services in the system, including inactive or archived ones, along with management data like status, price, and edit options under the Owner's authority | | |
| Preconditions | * Services have been created and saved in the system. * All services (active/inactive) are accessible. | | |
| Postconditions | * A list of services is displayed. * Actors can view full service details. * Manager may **edit/delete/deactivate**. | | |
| Normal Sequence/Flow | Actor logs into the **admin dashboard**.   1. Navigates to **Service Management**. 2. System shows:  * Full service list (Active/Inactive) * Name, Price, Status, Edit/Delete buttons * Sorting, search, and filtering by category or status  1. Actor clicks on a service to see detailed view or change it. | | |
| Alternative Sequences/Flows | * **No Services Available:** * System shows: “No services currently available.” * **Service Not Found or Deleted:** * System displays error message: “Service does not exist.” | | |

### *2.6 Staff Management*

#### 2.6.1 View List Order for Staff

#### 

| Primary Actors | Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Staff** to view a complete list of **court and service orders** created by users, including booking details, order status, payment status, and time. Staff can use this list to manage on-site service preparation, verify arrivals, and prepare resources for each time slot. | | |
| Preconditions | * Staff is authenticated and authorized to access the order module. * Orders have been created by users through the system. * The system’s order tracking module is active. | | |
| Postconditions | * A table of user orders (court bookings and services) is displayed. * Staff can filter, sort, and view detailed information. * Staff may optionally update order status (e.g., "Checked-in"). | | |
| Normal Sequence/Flow | 1. Staff logs into the **staff dashboard**. 2. Navigates to **Orders → View List**. 3. System displays:  * Court bookings (with time, court ID, user name, status) * Service orders (e.g., water, rentals, with quantity, status) * Columns:   + Order ID   + User name   + Order type (Court/Service)   + Date & time   + Status (Paid, Unpaid, Completed, Canceled)   + Action: View Details  1. Staff filters orders by:  * Date * Status * Type (Court / Service)  1. Staff clicks **“View”** to open full order detail. | | |
| Alternative Sequences/Flows | 1. **No Orders Found:**  * System shows: “No orders found for selected criteria.”  1. **Failed to Load Orders:**  * Error is shown: “Unable to load order list. Please try again.” | | |

#### 2.6.2 Perform Check-In

#### 

| Primary Actors | Staff | Secondary Actors |  |
| --- | --- | --- | --- |
| Description | This use case allows **Staff** to mark a booking or service order as **"Checked-in"** when the user arrives at the facility. This helps ensure proper scheduling, prepare the court, and confirm attendance for reporting and billing purposes. | | |
| Preconditions | * Staff is logged into the system with the appropriate role. * The booking/order has been created and is scheduled for the current or upcoming time. * Payment has been completed or deposit made (depending on policy). | | |
| Postconditions | * The booking/service order status is updated to **Checked-in**. * The check-in time is recorded in the system. * Optionally, a notification is sent to the user or management (if configured). | | |
| Normal Sequence/Flow | 1. Staff accesses the **order list** or **schedule view**. 2. Searches for or selects a **specific booking/service order**. 3. Confirms the user's identity (e.g., name, phone, or QR code). 4. Clicks **“Check-In”**. 5. System updates the order status to **Checked-in**. 6. A success message | | |
| Alternative Sequences/Flows | 1. **Order Not Found:**  * System shows: “Booking not found or not scheduled for today.”  1. **User Information Mismatch:**  * System blocks check-in and shows: “User identity could not be verified.”  1. **System Error During Update:**  * System displays: “Unable to perform check-in. Please try again later.” | | |

#### 

#### 2.6.3 Create Booking for Customer

#### 

| Primary Actors | Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Staff** to manually create a court booking on behalf of a **Customer**. This is useful when the customer books directly at the facility, via phone call, or doesn’t have access to the website or app. Staff can select courts, time slots, and input basic customer details. | | |
| Preconditions | * Staff is authenticated and has access to booking functions. * At least one court and available time slot exist. * The booking date/time is valid (in the future, within booking window). | | |
| Postconditions | * A booking is created in the system and linked to either:    + A registered user (if known)   + Or as a **guest booking** with phone number or name * The system optionally generates a payment QR or confirms deposit. | | |
| Normal Sequence/Flow | 1. Staff logs into the **staff dashboard**. 2. Navigates to **Bookings → Create Booking for Customer**. 3. Inputs customer details:  * Name * Phone number or email (optional)  1. Selects:  * Court * Date * Time slot * Booking type (One-time, Fixed, Optimized)  1. Confirms pricing and adds notes if necessary. 2. Clicks **“Confirm Booking”**. 3. System creates the booking and shows:  * Booking summary * Optional: Generate payment QR or mark as deposit received  1. Staff confirms with the customer and (optionally) sends email summary. | | |
| Alternative Sequences/Flows | 1. **Payment Pending:**  * System creates booking with **Pending Payment** status and shows QR.  1. **Customer Cancels Midway:**  * Staff discards form; no booking created. | | |

#### 2.6.4 View Staff Dashboard

#### 

| Primary Actors | Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Staff** to access the centralized dashboard after logging in. The dashboard provides a summary of daily bookings, service orders, check-in status, upcoming schedules, and quick actions such as creating bookings or processing payments. | | |
| Preconditions | * The actor is authenticated as **Staff**. * The staff dashboard module is enabled. * Booking, service, and report data are available in the system. | | |
| Postconditions | * The staff member sees an overview of relevant operational data. * From the dashboard, the staff can navigate to booking, order, and check-in functions. | | |
| Normal Sequence/Flow | 1. Staff logs into the system using valid credentials. 2. System redirects to the **Staff Dashboard**. 3. Dashboard displays:  * **Date Bookings** (list of court reservations with status) * Quick buttons:   + “Create Booking for Customer”   + “Sell Service”   + “View Calendar”  1. Staff clicks on a module (e.g., “Today’s Bookings”) to view details or take action. | | |
| Alternative Sequences/Flows | 1. **Dashboard Load Fails:**  * System displays: “Trang bạn đang tìm kiếm có thể đã bị xóa, tên trang đã được đổi hoặc tạm thời không khả dụng.”  1. **No Bookings/Orders Today:**  * System shows: “No data.” | | |

### *2.7 Payment System*

#### 2.7.1 Generate QR for Payment

#### 

#### 

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Users** or **Guests** to generate a QR code linked to their booking or service order. The QR code contains transaction details and can be scanned using supported banking or e-wallet apps to complete payment or deposit. | | |
| Preconditions | * A booking or service order exists in **pending payment** or **deposit required** status. * The payment gateway integration is active and functioning. * The user is on a device that supports QR generation/display (web or mobile). | | |
| Postconditions | * A QR code is generated and displayed to the user. * The code links to the correct payment amount and transaction ID. * Upon scanning and payment confirmation, the order is updated to **Paid** or **Deposit Received**. | | |
| Normal Sequence/Flow | 1. User completes a court booking or service order. 2. On the **payment page**, clicks **“Pay with QR Code”**. 3. System generates a secure QR code containing:  * Order ID * Amount due * Payment reference/token * Payment provider URL or payload  1. QR code is displayed on screen. 2. User scans the QR code with a bank or e-wallet app. 3. Upon successful payment, the payment gateway confirms the transaction. 4. System updates the order status to **Paid**. 5. Success message is displayed:  “Payment successful. Your booking is confirmed!” | | |
| Alternative Sequences/Flows | 1. **QR Generation Failure:**  * System shows error: “Unable to generate QR code. Please try another payment method.”  1. **Payment Not Confirmed in Time:**  * System cancels pending payment after timeout and shows message:  “Payment not received. Please try again or choose another method.”  1. **User Cancels or Closes Window:**  * No status change; order remains in **Pending Payment**. | | |

#### 

#### 2.7.2 Online Payment Processing

| Primary Actors | User | Secondary Actors | Payment Gateway |
| --- | --- | --- | --- |
| Description | This use case covers the process of securely handling online payments for court bookings or service orders via QR code or other integrated payment gateways. After an order is created (court or service), the system generates a payment request and processes confirmation upon successful transaction. | | |
| Preconditions | * The user has selected to pay for a court or service order. * A valid order with total amount exists in the system. * The payment gateway (e.g., VNPay, Momo, etc.) is connected and operational. | | |
| Postconditions | * The system marks the order as **Paid**. * A transaction record is created and linked to the order. * The booking/service is confirmed, and user receives a receipt. | | |
| Normal Sequence/Flow | 1. User confirms their booking or service order. 2. System generates a payment request with total amount. 3. System creates and displays a **QR code** or redirects to the payment gateway. 4. User scans QR or completes payment via banking app. 5. Payment gateway notifies the system of success. 6. System updates:  * Order status → **Paid** * Booking status → **Confirmed** * Transaction record with:   + Amount   + Time   + Payment method   + Order ID  1. Confirmation screen is shown. 2. System sends email/notification receipt (optional). | | |
| Alternative Sequences/Flows | 1. **3A. QR Code Fails to Load:**  * System displays error and retries or allows manual link.  1. **5A. Payment Fails or Timeout:**  * System shows “Payment failed. Please try again.” * Order remains in “Pending” status.  1. **6A. Delay in Payment Notification:**  * System periodically polls payment gateway for status.  1. **7A. User Closes Payment Window Early:**  * System allows retry or shows pending status in order history. | | |

#### 

#### 

### *2.8 Notification Management*

#### 2.8.1 Receive Notifications

| Primary Actors | User, Guest, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows all actors (**User, Guest, Staff, Manager, Owner**) to **receive notifications** triggered by system activities such as bookings, payments, refunds, schedule changes, maintenance updates, or promotional announcements. Notifications help users stay informed about important events related to their activities in the system. | | |
| Preconditions | * The actor is authenticated in the system or has an active session (Guest). * Notification generation service is active and configured. * Relevant trigger events occur (e.g., successful booking, refund processed, maintenance scheduled). | | |
| Postconditions | * The actor receives the corresponding notification. * Notifications are stored in the system for future reference. * Users can view unread and read notifications separately. | | |
| Normal Sequence/Flow | 1. A trigger event happens in the system (e.g., booking created, payment confirmed). 2. System generates a notification based on event type and actor role. 3. System delivers the notification to:     * + - In-app Notification Center        - Optionally, Email or SMS if configured 4. User/Actor logs in or is already logged in. 5. User/Actor views notification list, sees a badge or alert icon. 6. User/Actor clicks to open and read full notification details. 7. Notification is marked as read once opened. | | |
| Alternative Sequences/Flows | 1. **Notification Delivery Failure (External):**  * If email or SMS fails, the notification remains available in-app only. * System logs the delivery failure for admin review.  1. **User Disables Some Notification Types:**  * Only selected types are pushed to the user based on their preferences.  1. **Trigger Event Error:**  * If trigger event fails (e.g., failed payment), system skips sending success notifications. | | |

### *2.9 Financial Management*

#### 2.9.1 View All Transaction History by Owner

| Primary Actors | Owner | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to access and review the full list of all payment transactions that occurred in the system, including those related to court bookings, service purchases, deposits, and refunds. Transactions include payment method, status, amount, and are filterable for financial reporting and auditing. | | |
| Preconditions | * The actor is authenticated with an **Manager** role. * Transactions have been recorded in the system. * The transaction log module is enabled and accessible from the dashboard. | | |
| Postconditions | * A list of all system-wide financial transactions is displayed. * The actor can filter, sort, and click to view specific transaction details. * Optional: Export the transaction list to Excel/PDF. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin panel. 2. Navigates to **Transaction**. 3. System loads a list of all transactions, each showing:  * Transaction ID * Bill Code * Order ID * Payment status * Amount * Timestamp  1. Actor filters by:  * Date range * Payment status * Order ID  1. Optionally, clicks **“Export”** to download the list. | | |
| Alternative Sequences/Flows | 1. **No Transactions Found:**  * System shows: “No data.”  1. **Filter Yields No Results:**  * System shows: “No data.” | | |

#### 

#### 2.9.2 View All Transaction History by Manager

| Primary Actors | Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to access and review the full list of all payment transactions that occurred in the system, including those related to court bookings, service purchases, deposits, and refunds. Transactions include payment method, status, amount, and are filterable for financial reporting and auditing under the Owner's authority | | |
| Preconditions | * The actor is authenticated with an **Manager** role. * Transactions have been recorded in the system. * The transaction log module is enabled and accessible from the dashboard. | | |
| Postconditions | * A list of all system-wide financial transactions is displayed. * The actor can filter, sort, and click to view specific transaction details. * Optional: Export the transaction list to Excel/PDF. | | |
| Normal Sequence/Flow | 1. Manager logs into the admin panel. 2. Navigates to **Transaction**. 3. System loads a list of all transactions, each showing:  * Transaction ID * Bill Code * Order ID * Payment status * Amount * Timestamp  1. Actor filters by:  * Date range * Payment status * Order ID  1. Optionally, clicks **“Export”** to download the list. | | |
| Alternative Sequences/Flows | 1. **No Transactions Found:**  * System shows: “No data.”  1. **Filter Yields No Results:**  * System shows: “No data.” | | |

#### 

#### 2.9.3 View Revenue Report by Owner

| Primary Actors | Owner | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to access a visual and/or tabular summary of **total revenue** generated from court bookings, service purchases, and other transactions within a selected time frame. The revenue report helps monitor business performance, identify trends, and support financial decision-making. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * Revenue data exists for the selected time range. * The reporting/analytics module is active. | | |
| Postconditions | * A visual revenue summary (charts, graphs, totals) is displayed. * Data is filterable by date, type (court/service), and source (QR/cash/etc.). * Optional export/download (Excel/PDF) may be available. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Statistic**. 3. Selects a time range. 4. System retrieves revenue data from bookings. 5. System displays:  * Total revenue * Revenue breakdown by type (court, deposit,..) * Bar Chart, line Chart.  1. Actor may filter by:  * By court, by day, by month * Date range * Payment status * Order status,..  1. Optionally, clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Available:**  * System displays: “No data ”  1. **Analytics Module Error:**  * System shows fallback text view or error message and retry option.  1. **Export Fails:**  * System display:” Xuất file không thành công”. | | |

#### 2.9.4 View Revenue Report by Manager

| Primary Actors | Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to access a visual and/or tabular summary of **total revenue** generated from court bookings, service purchases, and other transactions within a selected time frame. The revenue report helps monitor business performance, identify trends, and support financial decision-making under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * Revenue data exists for the selected time range. * The reporting/analytics module is active. | | |
| Postconditions | * A visual revenue summary (charts, graphs, totals) is displayed. * Data is filterable by date, type (court/service), and source (QR/cash/etc.). * Optional export/download (Excel/PDF) may be available. | | |
| Normal Sequence/Flow | 1. Manager logs into the admin dashboard. 2. Navigates to **Statistic**. 3. Selects a time range. 4. System retrieves revenue data from bookings. 5. System displays:  * Total revenue * Revenue breakdown by type (court, deposit,..) * Bar Chart, line Chart.  1. Actor may filter by:  * By court, by day, by month * Date range * Payment status * Order status,..  1. Optionally, clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Available:**  * System displays: “No data ”  1. **Analytics Module Error:**  * System shows fallback text view or error message and retry option.  1. **Export Fails:**  * System display:” Xuất file không thành công”. | | |

#### 2.9.5 Export Revenue Report by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to export a detailed revenue report summarizing income from court bookings, service purchases, and other transactions within a specified time range. The report is exported as an Excel (.xlsx) or PDF file for analysis, recordkeeping, or financial auditing. | | |
| Preconditions | * The actor is logged in with **Owner** role. * Revenue data exists in the system for the selected date range. * The system has access to the reporting module and database. | | |
| Postconditions | * A structured financial report file (Excel) is generated. * The file includes totals, breakdowns, and optional transaction logs. * The action may be logged for security/audit tracking. | | |
| Normal Sequence/Flow | 1. Owner navigates to the **Reports** or **Revenue Dashboard**. 2. Selects:  * Date range * Payment status * Order status * Court * By Court, by month, by day * Order Type  1. Selects **“Xuất Báo Cáo”**. 2. System retrieves all relevant financial data. 3. System generates a report including:  * Total Revenue * Total Deposit * Total Refund * Total Paid,...  1. System prepares the file and download. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Found:**  * System displays all fields as 0.  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công” | | |

#### 

#### 2.9.6 Export Revenue Report by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Managerto export a detailed revenue report summarizing income from court bookings, service purchases, and other transactions within a specified time range. The report is exported as an Excel (.xlsx) or PDF file for analysis, recordkeeping, or financial auditing. | | |
| Preconditions | * The actor is logged in with **Manager** role. * Revenue data exists in the system for the selected date range. * The system has access to the reporting module and database. | | |
| Postconditions | * A structured financial report file (Excel or PDF) is generated. * The file includes totals, breakdowns, and optional transaction logs. * The action may be logged for security/audit tracking. | | |
| Normal Sequence/Flow | 1. Manager navigates to the **Reports** or **Revenue Dashboard**. 2. Selects:  * Date range * Payment status * Order status * Court * By Court, by month, by day * Order Type  1. Selects **“Xuất Báo Cáo”**. 2. System retrieves all relevant financial data. 3. System generates a report including:  * Total Revenue * Total Deposit * Total Refund * Total Paid,...  1. System prepares the file and download. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Found:**  * System displays all fields as 0.  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công” | | |

#### 2.9.7 Export Booking Orders by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables the Owner to export a report of all purchase orders made by users, including court bookings and service orders, within a specific time range. The exported data supports operational tracking, accounting, and reconciliation processes, and is generated in Excel (.xlsx) or PDF format. | | |
| Preconditions | * The actor is authenticated with appropriate permissions (Owner). * The system contains at least one order within the selected timeframe. * The reporting module is active and integrated with the database. | | |
| Postconditions | * A structured export file is generated with order information. * The file is successfully downloaded by the actor. * The export may be logged for administrative purposes. | | |
| Normal Sequence/Flow | 1. Owner logs in and navigates to the **Booking Management** or **Reports** section. 2. Selects:  * Date range * Payment status * Order status * Court name  1. Clicks on **“Xuất Excel”**. 2. System retrieves matching orders from the database. 3. System generates a file containing:  * Order ID * Court ID * Custome name * Phone Number * Order Type * Order Status * Payment Status * Total Amount * Deposit Amount * Disscount Amount,...  1. System prepares the file for download. 2. Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | 1. **No Orders Found:**  * System displays: “No data.”  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công.” | | |

#### 

#### 2.9.8 Export Booking Orders by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables the Manager to export a report of all purchase orders made by users, including court bookings and service orders, within a specific time range. The exported data supports operational tracking, accounting, and reconciliation processes, and is generated in Excel (.xlsx) or PDF format under the Owner's authority | | |
| Preconditions | * The actor is authenticated with appropriate permissions ( Manager). * The system contains at least one order within the selected timeframe. * The reporting module is active and integrated with the database. | | |
| Postconditions | * A structured export file is generated with order information. * The file is successfully downloaded by the actor. * The export may be logged for administrative purposes. | | |
| Normal Sequence/Flow | 1. Manager logs in and navigates to the **Booking Management** or **Reports** section. 2. Selects:  * Date range * Payment status * Order status * Court name  1. Clicks on **“Xuất Excel”**. 2. System retrieves matching orders from the database. 3. System generates a file containing:  * Order ID * Court ID * Custome name * Phone Number * Order Type * Order Status * Payment Status * Total Amount * Deposit Amount * Disscount Amount,...  1. System prepares the file for download. 2. Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | 1. **No Orders Found:**  * System displays: “No data.”  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công.” | | |

#### 

#### 

#### 2.9.9 Export Transaction by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to export a list of financial transactions recorded in the system, including payments for court bookings, services, and other activities. The transaction report can be filtered by date, payment status, or payment method, and is exported as an Excel (.xlsx) for accounting and auditing purposes. | | |
| Preconditions | * The actor is authenticated with the **Owner** role. * The system contains transaction records for the time frame, or by the selected court. * The reporting module is active and has access to transaction logs. | | |
| Postconditions | * A transaction report is generated and ready for download. * The file includes accurate transaction data based on the selected filters. | | |
| Normal Sequence/Flow | Owner login to the admin dashboard.  Navigates to the **Transaction** section.  Selects:   * Date range * Payment status * By day * Order status * Order type * Court Name   Click on **“Xuất báo cáo”**.  System queries the transaction database.  System generates a file including:   * Period * Court ID * Court Name * Total Revenue * Deposit Amount * Paid Amount * Refund Amount   System prepares the file for download.  Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | * **No Transactions Found:** * System shows: “No data.” * **Download Failed:** * System shows error: “Xuất file không thành công”. | | |

## 

#### 2.9.10 Export Transaction by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to export a list of financial transactions recorded in the system, including payments for court bookings, services, and other activities. The transaction report can be filtered by date, payment status, or payment method, and is exported as an Excel (.xlsx) for accounting and auditing purposes under the Owner's authority | | |
| Preconditions | * The actor is authenticated with the **Manager** role. * The system contains transaction records for the time frame, or by the selected court. * The reporting module is active and has access to transaction logs. | | |
| Postconditions | * A transaction report is generated and ready for download. * The file includes accurate transaction data based on the selected filters. | | |
| Normal Sequence/Flow | Manager login to the admin dashboard.  Navigates to the **Transaction** section.  Selects:   * Date range * Payment status * By day * Order status * Order type * Court Name   Click on **“Xuất báo cáo”**.  System queries the transaction database.  System generates a file including:   * Period * Court ID * Court Name * Total Revenue * Deposit Amount * Paid Amount * Refund Amount   System prepares the file for download.  Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | * **No Transactions Found:** * System shows: “No data.” * **Download Failed:** * System shows error: “Xuất file không thành công”. | | |

#### 2.9.11 Forgot Password

| Primary Actors | User, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | When users forget the password and want to get the password back to log in | | |
| Preconditions | * Users must have an associated account with valid email | | |
| Postconditions | * Users get the password back via email | | |
| Normal Sequence/Flow | 1. Users on Home page 2. Users click on "Forget Password" 3. User username/email/phonenumber to get a password via email 4. Users press the "Reset Password" button | | |
| Alternative Sequences/Flows | **N/A** | | |

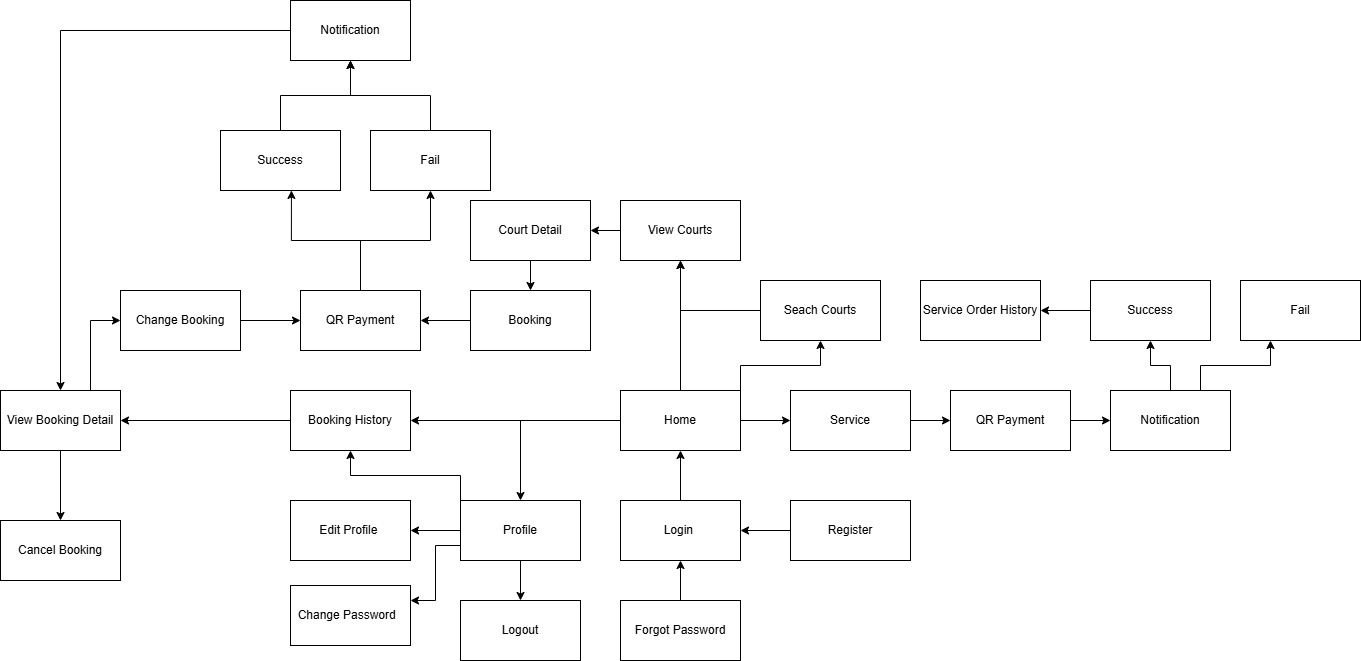
## 

***Figure 4.6****: UseCase for Owner*

### 1.4 System Functionalities

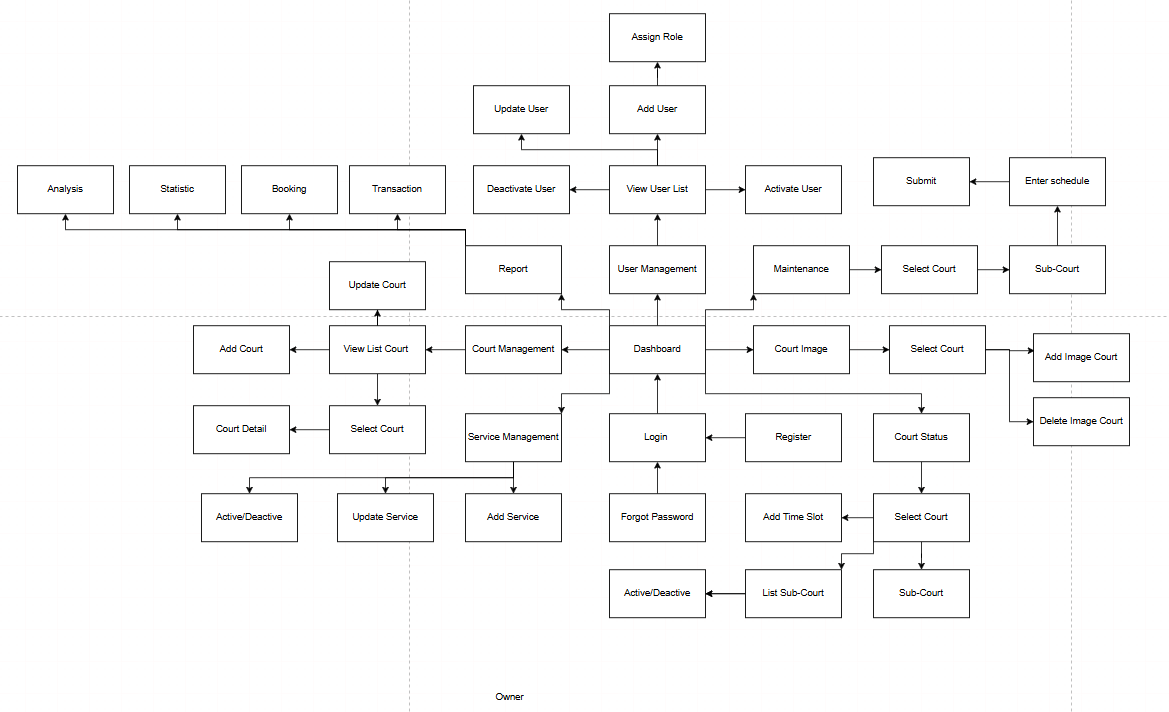
#### 1.4.1 Screens Flow

##### 1.4.1.1 Screens Flow User

**

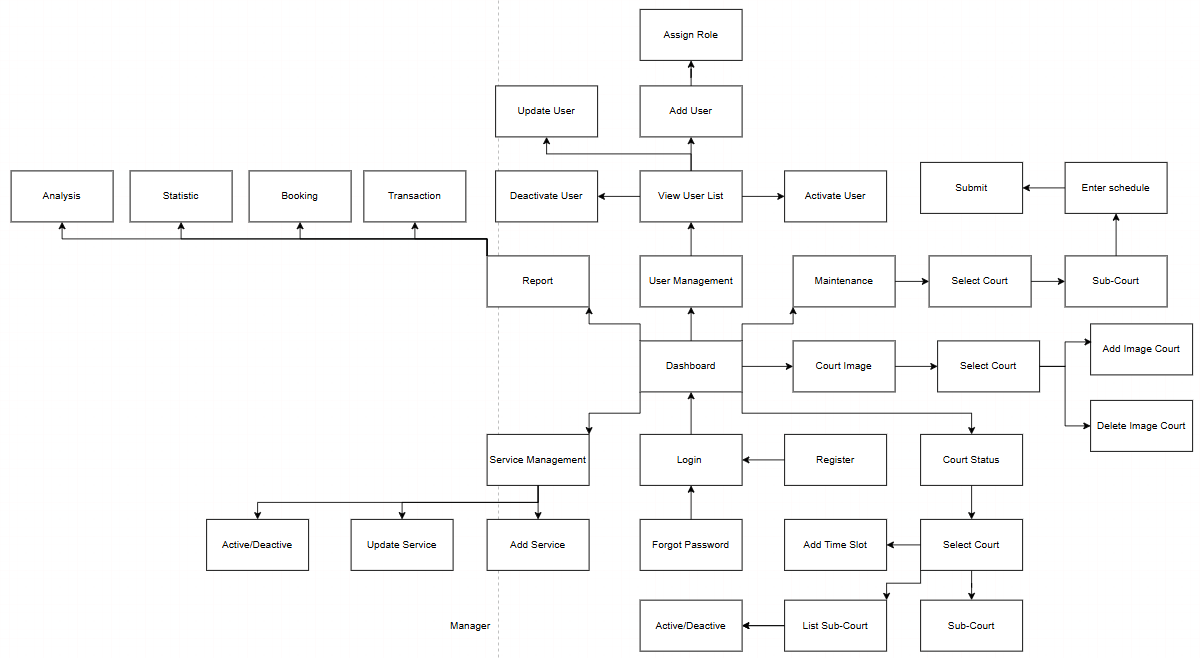
***Figure 4.7****: Screens Flow User*

##### 1.4.1.2 Screens Flow Owner



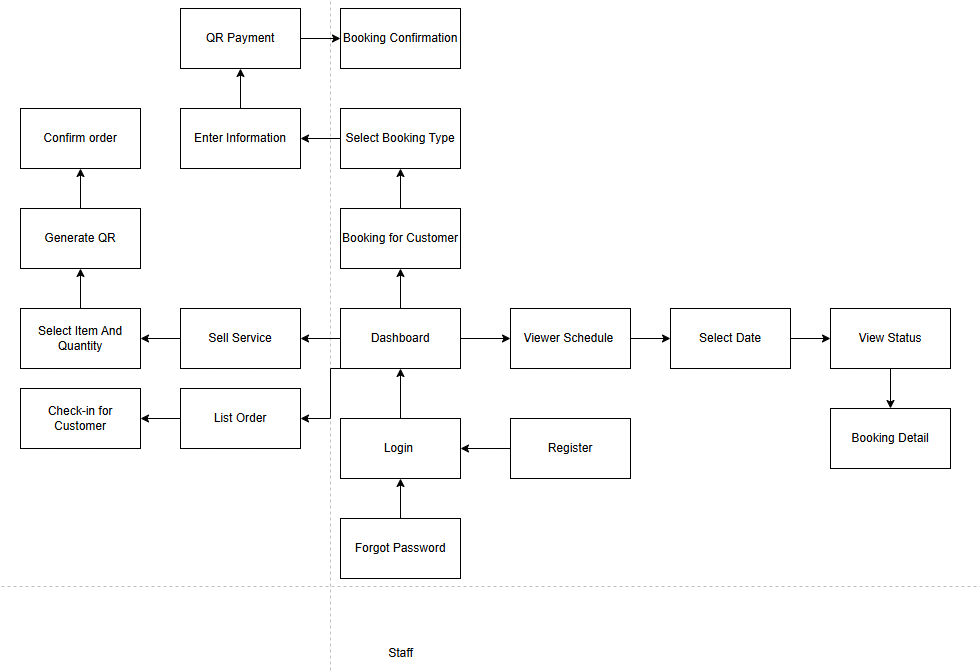
***Figure 4.8****: Screens Flow Owner*

##### 1.4.1.3 Screens Flow Manager

**

***Figure 4.9****: Screens Flow Manager*

##### 1.4.1.4 Screens Flow Staff

**

***Figure 5.0****: Screens Flow Staff*

#### 1.4.2 Screen Authorization

| **Screen** | **Owner** | **Staff** | **Manager** | **User** | **Guest** |
| --- | --- | --- | --- | --- | --- |
| Home | X | X | X | X | X |
| Login | X | X | X | X | X |
| Register |  |  |  |  | X |
| Forgot Password | X | X | X | X |  |
| Court Detail | X | X | X | X | X |
| Booking Screen | X | X | X | X | X |
| View Booking History | X |  | X |  |  |
| Change Booking | X | X | X | X | X |
| Cancel Booking | X | X | X | X | X |
| Payment (QR Generation) | X | X | X | X | X |
| Service Screen | X | X | X | X | X |
| Sell Service |  | X |  |  |  |
| Personal / Profile Screen | X | X | X | X |  |
| Edit Profile | X | X | X | X |  |
| Change Password | X | X | X | X |  |
| Language Switch | X | X | X | X | X |
| Notification Center | X | X | X | X | X |
| Report / Analytics | X |  | X |  |  |
| Court Management | X |  | X |  |  |
| Add/Edit Court | X |  | X |  |  |
| Court Zone & Time Slot Management | X |  | X |  |  |
| Add/Edit Time Slot | X |  | X |  |  |
| Service Management | X |  | X |  |  |
| Image Management (Court images) | X |  | X |  |  |
| Maintenance Scheduling | X |  | X |  |  |
| Add/Edit Role & Permissions | X |  |  |  |  |
| Staff Management | X |  | X |  |  |
| User Management | X |  | X |  |  |
| Register Student | X | X | X |  | X |

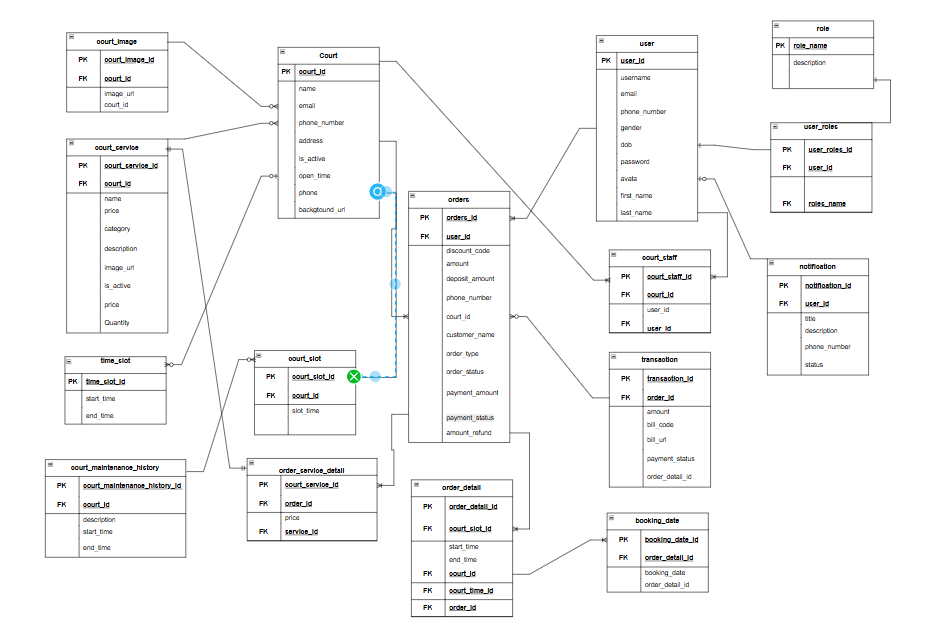
***Table 4.2****: Screen Authorization*

#### 1.4.3 Non-UI Functions

| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | User Management | User Role Assignment Service | Assigns roles to users and updates their permissions accordingly. |
| 2 | Authentication | Login Service | Handles user authentication and token generation. |
| 3 | Data Processing | User Data Sync Batch | A scheduled batch job that synchronizes user data across different services. |
| 4 | Notifications | Notification Service | Sends system notifications based on user actions and scheduled events. |
| 5 | Reporting | User Activity Report API | Retrieves user activity logs for Owneristrative review. |
| 6 | Access Control | Permission Validation API | Validates user permissions before granting access to restricted areas. |

***Table 4.3****: Non-UI Functions*

### 1.5 Entity Relationship Diagram

******

***Entities Description***

| **#** | **Entity** | **Description** |
| --- | --- | --- |
| 1 | Court | Information about sports courts, including name, address, email, phone number, description, background image, and activity status. |
| 2 | Court\_Service | Additional services provided by the court (e.g., equipment rental, beverages, coaching). |
| 3 | Court\_Slot | Time slots available for court booking. |
|  | Court\_Price | Pricing details for court bookings depending on the day (weekday, weekend) and customer type (student, adult). |
|  | Court\_Image | Images related to each court (a court can have multiple images). |
|  | Court\_Maintenance\_History | Maintenance history of courts, recording the start time, end time, and description of the maintenance work. |
|  | Time\_Slot | Standard defined time ranges (e.g., 7 AM - 8 AM, 8 AM - 9 AM). |
|  | User | Information about system users, including email, phone number, password, date of birth, gender, and avatar. |
|  | Role | Different user roles in the system (e.g., Admin, Staff, User, Manager). |
|  | User\_Roles | Relationship linking users and their assigned roles (one user can have multiple roles). |
|  | Court\_Staff | Court staff information, linking users who are employees of a specific court. |
|  | Orders | Court booking orders, capturing customer information, total amount, discount code, and payment method. |
|  | Order\_Detail | Details of court booking orders: which court, which time slot, start and end times. |
|  | Order\_Service\_Detail | Additional services booked within an order (e.g., ball rental, coach rental). |
|  | Booking\_Date | Specific booking dates associated with an order detail. |
|  | Notification | Notifications sent to users regarding orders, payments, or other events. |
|  | Transaction | Payment transaction details for an order, including amount, bill code, and payment status. |

## 2. Use Case Specifications

### 2.1 Common Feature

###### **2.1 Common Feature**

#### 2.1.1 Register Student Account

| Primary Actors | Guest | Secondary Actors | Staff |
| --- | --- | --- | --- |
| Description | This use case allows a student to create a user account by providing required information such as a valid university email (e.g., ending with **@edu.vn**) and phone number. The system verifies the information to confirm the student’s eligibility before allowing access to student-specific features (e.g., discounted pricing, student promotions). | | |
| Preconditions | * The student has not previously registered with the same email or phone number. * The system is connected to the student verification mechanism (e.g., email check, token validation). * The registration feature is enabled and accessible. | | |
| Postconditions | * A new student account is successfully created and marked with the **student role**. * The student receives confirmation via email or notification. * The student is granted access to benefits such as student pricing. | | |
| Normal Sequence/Flow | 1. Student navigates to the **Register** page. 2. Select “Register as Student”. 3. Inputs required information:  * First Name * Last Name * Student email (e.g., abc@university.edu) * Phone number * Password * Date of birth * Gender  1. System validates:  * Email domain (e.g., must end in edu.vn) * Unique email/phone  1. If valid, the system creates a **student user account** and assigns appropriate roles. 2. System sends confirmation messages via email. 3. Student is logged in and redirected to the home screen. | | |
| Alternative Sequences/Flows | * **Invalid Email Format:** * System shows an error: “Please use a valid university email address.” * **Duplicate Email or Phone:** * System shows: “This email/phone is already in use.” | | |

#### 2.1.2 Login

| Primary Actors | User/Guest, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | Allows users to log in and access the system. The system will authenticate and authorize the user. | | |
| Preconditions | User account has been created & authorized. | | |
| Postconditions | * User logs in the system successfully * The system tracked successful login into the Activity Log | | |
| Normal Sequence/Flow | 1. User clicks on the link to navigate to the login page. 2. User enter username/gmail/phone number and password and click on “Đăng nhập” (A-F\_1) 3. System redirect user to the home page | | |
| Alternative Sequences/Flows | **A-F\_1**. **System can’t authenticate the user**  Users can not login if:   1. User leaves the email/username/phone number or password field blank (MSG01) 2. User enters the invalid email/username/phone number (MSG02) 3. User input wrong password format (MSG03, MSG04, MSG05) 4. The account is not registered with the system or is blocked (MSG07) | | |

#### 2.1.3 Logout

| Primary Actors | User, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | Users can securely terminate their active sessions | | |
| Preconditions | User has account and logged in | | |
| Postconditions | User logout successfully | | |
| Normal Sequence/Flow | 1. User click the avatar and click button “Log out” 2. System logs out the user | | |
| Alternative Sequences/Flows | If the user's token times out, the user will be automatically logged out. | | |

#### 2.1.4 Change Password

| Primary Actors | User, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the user to reset the password. | | |
| Preconditions | * User has account and logged in * User is on profile page | | |
| Postconditions | User resets the password for the account. | | |
| Normal Sequence/Flow | 1. User get on to the profile page and click on “Đặt lại mật khẩu” button 2. System display Change Password form 3. User enter old password, new password and re-password and click on “Save” (A-F\_1) 4. System display success message (MSG07) | | |
| Alternative Sequences/Flows | **A-F\_1: System cannot change password**  User cannot change to new password if:   1. User input wrong password format (MSG03, MSG04, MSG05) 2. User input re-password not match the password (MSG06) | | |

#### 2.1.5 Register New User

#### 

| Primary Actors | Guest | Secondary Actors |  |
| --- | --- | --- | --- |
| Description | As a new user, I want to register an account so that I can book pickleball courts, and manage my profile within the system. | | |
| Preconditions | The user must have access to a valid email or phone number for verification. | | |
| Postconditions | 1. The new user account is successfully created. 2. The user can log in and access the system based on their assigned role. | | |
| Normal Sequence/Flow | 1. The system prompts the user to enter required information (e.g., name, email, phone, password). 2. The user is redirected to the login page with a success message. | | |
| Alternative Sequences/Flows | **AF1: Email already exists in the system**   * The system notifies the user that the email is already in use and prompts them to log in or recover their account.   **AF2: System or network failure during registration**   * The system displays an error message and prompts the user to try again later. | | |

### 2.2 User Management

#### 2.2.1 Update User Profiles

| Primary Actors | User | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | As a registered user, I want to update and manage my profile details so that I can ensure my information is accurate and up to date. | | |
| Preconditions | 1. The user must be logged into the system. 2. The system must have existing user profile data. | | |
| Postconditions | 1. The user’s profile information is successfully updated. 2. If information is changed, a confirmation notification is sent. | | |
| Normal Sequence/Flow | 1. The user navigates to the profile management page. 2. The system displays the current profile information. 3. The user updates one or more profile fields (e.g., name, contact info, password). 4. The user submits the changes. 5. The system validates the input data and updates the profile. 6. A success message is displayed, and a notification is sent if necessary. | | |
| Alternative Sequences/Flows | **AF1: User enters an invalid email or phone number**   * The system displays an error message and asks for a valid input.   **AF2: System failure during update**   * The system logs the error and prompts the user to try again later. | | |

#### 

#### 2.2.2. Update Role User by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Owner to update the role of an existing user (e.g., Staff, Student, Manager, User) changes affect what functionalities the user can access in the system. | | |
| Preconditions | * The actor is logged in as an Owner * The user must already exist in the system. * The new role must be valid and active in the system. | | |
| Postconditions | * The selected user has a new role assigned. * Their access rights are updated based on the new role. * The update is logged for auditing purposes (if applicable). | | |
| Normal Sequence/Flow | 1. Owner logs into the system. 2. Navigates to **User Management** section. 3. Searches for and selects the user whose role needs to be updated. 4. Clicks the **"Edit"** button next to the user’s profile. 5. Selects the new role from a dropdown list (e.g., User, Staff, Manager). 6. Clicks **"Save"** . 7. System validates the role change and updates the user’s profile. 8. A confirmation message is shown (e.g., "Success"). 9. The new role permissions take effect immediately on the next login or page refresh. | | |
| Alternative Sequences/Flows | 1. **Invalid Role Selected:**  * System displays an error if the role is inactive or unauthorized for assignment.  1. **Update Fails (e.g., network/server issue):**  * System displays an error message and logs the issue for retry. | | |

#### 2.2.3 Update Role User by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Manager to update the role of an existing user (e.g., Staff, Student, Manager, User) changes affect what functionalities the user can access in the system | | |
| Preconditions | * The actor is logged in as an Manager * The user must already exist in the system. * The new role must be valid and active in the system. | | |
| Postconditions | * The selected user has a new role assigned. * Their access rights are updated based on the new role. * The update is logged for auditing purposes (if applicable). | | |
| Normal Sequence/Flow | 1. Owner logs into the system. 2. Navigates to **User Management** section. 3. Searches for and selects the user whose role needs to be updated. 4. Clicks the **"Edit"** button next to the user’s profile. 5. Selects the new role from a dropdown list (e.g., User, Staff, Manager). 6. Clicks **"Save"** . 7. System validates the role change and updates the user’s profile. 8. A confirmation message is shown (e.g., "Success"). 9. The new role permissions take effect immediately on the next login or page refresh. | | |
| Alternative Sequences/Flows | * **Invalid Role Selected:** * System displays an error if the role is inactive or unauthorized for assignment. * **Update Fails (e.g., network/server issue):** * System displays an error message and logs the issue for retry. | | |

#### 

#### 2.2.4 Create Role

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to define and create a **new role** in the system, such as "Coach", "Customer Support", or any custom role. A role defines the level of access a user has to specific features and modules. After creation, the role can be assigned to users and have permissions configured accordingly. | | |
| Preconditions | 1. The actor is authenticated as **Owner**. 2. The system is in a state that allows configuration changes. 3. The role name to be created is unique. | | |
| Postconditions | 1. A new role is successfully created in the system. 2. The role is visible in the role management module. 3. Permissions can now be assigned to the new role. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Role Management**. 3. Clicks **“Create New Role”**. 4. Enters:  * Role name (e.g., "Staff Assistant") * Optional description  1. Clicks **“Add”**. 2. System checks for name uniqueness and validates input. 3. System creates the new role and confirms success. 4. Owner is redirected to assign permissions to the new role (optional next step). | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * System display: “Error” | | |

#### 2.2.5 Delete Role

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to delete a custom-defined role from the system. The system will ensure that the role is **not currently assigned** to any user before allowing deletion. This helps maintain role management cleanliness and prevent unused roles from cluttering the system. | | |
| Preconditions | * The actor is authenticated as an **Owner**. * The role to be deleted is **not a system default role** (e.g., User, Manager). * The role is **not assigned** to any active users. | | |
| Postconditions | * The selected user has a new role assigned. * Their access rights are updated based on the new role. * The update is logged for auditing purposes (if applicable). | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to  **Role Management**. 3. Sees a list of roles with action buttons. 4. Clicks **“Delete”** next to a custom role. 5. System checks if the role is:  * Custom-defined (not system default) * Not assigned to any users  1. System deletes the role and shows success message. | | |
| Alternative Sequences/Flows | 1. **Owner Cancels Confirmation:**  * No action is taken; role remains.  1. **System Error During Deletion:**  * System displays error , try again again. | | |

#### 2.2.6 View Role

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to view a list of all defined user roles in the system, including system default roles (e.g., User, Guest, Staff) and custom roles created by the Owner. The role view includes details such as the role name, description, assigned permissions, and the number of users currently using the role. | | |
| Preconditions | * The actor is authenticated as **Owner** * The role management module is enabled. * Roles have already been defined in the system. | | |
| Postconditions | * The system displays a list of all roles. * The actor may view role details, including associated permissions and user count. * (Optional) The actor can navigate to **Edit**, **Assign**, or **Delete** role actions. | | |
| Normal Sequence/Flow | 1. Actor logs into the admin dashboard. 2. Navigates to **Role Management → View Roles**. 3. System displays a table with the following info per role:  * Role name * Role description  1. Actor clicks on a specific role to view details. 2. System shows:  * Full description | | |
| Alternative Sequences/Flows | 1. **No Roles Exist:**  * System displays: “No data”  1. **Role Data Fails to Load:**  * System shows error:”Failed to load data” | | |

#### 

#### 2.2.7 View List of Users

| Primary Actors | Owner, Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | The "View User List" use case allows the Owner, Manager to view a list of all registered users in the Pickleball Court Management System. The list includes filtering options, allowing the Owner to refine the users displayed based on criteria such as login name, phone number, email, role. | | |
| Preconditions | * The Owner, Manager must be logged into the system using valid credentials. * The Owner, Manager must have appropriate permissions to access user information. | | |
| Postconditions | * The system displays a list of registered users, reflecting any filters applied by the Owner. | | |
| Normal Sequence/Flow | 1. **Access User Management:** The Owner navigates to the user management section of the system (e.g., via the menu or dashboard). 2. **Show All Users**: The system displays a list of all registered users, including details such as username, role, email, and phone number. 3. **Apply Filters**: The Owner selects filtering options (e.g., filter by role, phone number, or username). 4. **Update List**: The system refreshes the list to display only users that match the selected filters. 5. **View Details**: The Owner can select a user from the list to view additional details. | | |
| Alternative Sequences/Flows | 1. **No Matching Users**:  * If no users match the applied filters in step 4:   + - The system displays empty list.     - The Owner can modify or clear the filters to see the full list again.  1. **No Filters Used**:  * If the Owner skips step 3 and does not apply filters:   + - The system continues to show the complete list of all registered users.     - The Owner may proceed to step 5 to view individual user details.  1. **Data Retrieval Error**:  * If the system fails to load the user list in step 2 or step 4:   + - The system displays an error message (e.g., "Fail.").     - The Owner is prompted to retry.  1. **Exit Use Case**:  * At any point, if the Owner chooses to leave the user list view:   + - The system returns the Owner to the previous screen (e.g., dashboard).     - The use case ends. | | |

#### 2.2.8 Create New User by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables Owners to create new user accounts in the Pickleball Court Management System. Owners can input essential user details such as name, email, phone number, and assign an appropriate role (e.g., user, staff, manager) to the new user. | | |
| Preconditions | * The Owners must be logged into the system with valid credentials and possess the necessary permissions to create user accounts. * The email address provided for the new user must not already exist in the system. * The phone number provided for the new user must not already exist in the system. * The username provided for the new user must not already exist in the system. | | |
| Postconditions | * A new user account is successfully created with the provided details and assigned role. | | |
| Normal Sequence/Flow | 1. **Owner initiates user creation**: The Owner navigates to the user management section of the system and selects the option to create a new user. 2. **System provides a form**: A user creation form appears, containing fields for details such as name, email, phone number, and role,... 3. **Owner inputs user information**: The Owner enters the required details for the new user and selects a role from a predefined list (e.g., user, staff, owner). 4. **Owner submits the form**: After reviewing the entered data, The Owner submits the form. 5. **System validates the input**: The system checks the data for completeness and accuracy (e.g., ensuring the email format is valid and the email is unique). 6. **System creates the account**: Upon successful validation, the system generates a new user account with the provided details and assigned role. 7. **System confirms success**: The Owner receives a confirmation message indicating that the new user account has been successfully created. | | |
| Alternative Sequences/Flows | 1. Invalid Data Submission 2. **Scenario**: At step 5, if the system identifies invalid data (e.g., missing fields or incorrect email format):    * The system displays an error message detailing the issues.    * The Owner corrects the errors and resubmits the form.    * The process resumes at step 4. 3. Email, Phone number, Username Already in Use 4. **Scenario**: At step 5, if the email address, phone number, username is already registered:    * The system shows an error message stating the email, phone number, username is in use and prompts for a different email.    * The Owner provides a new email, phone number, username and resubmits the form.    * The process resumes at step 4. 5. Role Assignment Error 6. **Scenario**: At step 3, if The Owner and Manager selects an unavailable or restricted role:    * The system displays an error message indicating the role cannot be assigned.    * The Owner chooses a valid role and proceeds.    * The process resumes at step 4. 7. Notification Failure 8. **Scenario**: At step 7, if the notification fails to send (e.g., due to an invalid email or technical issue):    * The system logs the error and warns The Owner that the account was created but the notification failed.    * The Owner may manually notify the user or retry sending the notification.    * The process continues to step 8. 9. The Owner Cancels the Process 10. **Scenario**: After step 3, if The Owner decides not to proceed:     * The system discards the entered data and returns The Owner to the user management screen.     * The use case ends without creating an account. | | |

#### 2.2.9 Create New User by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to create a new user account in the Pickleball Court Management System and assign one of the roles: Staff, User, or Student | | |
| Preconditions | * The Manager must be logged into the system with valid credentials and possess the necessary permissions to create user accounts. * The email address provided for the new user must not already exist in the system. * The phone number provided for the new user must not already exist in the system. * The username provided for the new user must not already exist in the system. | | |
| Postconditions | * A new user account is successfully created with the provided details and assigned role. | | |
| Normal Sequence/Flow | 1. **Manager initiates user creation**: The Manager navigates to the user management section of the system and selects the option to create a new user. 2. **System provides a form**: A user creation form appears, containing fields for details such as name, email, phone number, and role,... 3. **Manage inputs user information**: The Manager enters the required details for the new user and selects a role (dropdown: Staff, User, Student) 4. **Manage submits the form**: After reviewing the entered data, The Owner and Manager submits the form. 5. **System validates the input**: The system checks the data for completeness and accuracy (e.g., ensuring the email format is valid and the email is unique). 6. **System creates the account**: Upon successful validation, the system generates a new user account with the provided details and assigned role. 7. **System confirms success**: The Manager receives a confirmation message indicating that the new user account has been successfully created. | | |
| Alternative Sequences/Flows | 1. Invalid Data Submission 2. **Scenario**: At step 5, if the system identifies invalid data (e.g., missing fields or incorrect email format):    * The system displays an error message detailing the issues.    * The Manager corrects the errors and resubmits the form.    * The process resumes at step 4. 3. Email, Phone number, Username Already in Use 4. **Scenario**: At step 5, if the email address, phone number, username is already registered:    * The system shows an error message stating the email, phone number, username is in use and prompts for a different email.    * The Manager provides a new email, phone number, username and resubmits the form.    * The process resumes at step 4. 5. Role Assignment Error 6. **Scenario**: At step 3, if The Manager selects an unavailable or restricted role:    * The system displays an error message indicating the role cannot be assigned.    * The Manager chooses a valid role and proceeds.    * The process resumes at step 4. 7. Notification Failure 8. **Scenario**: At step 7, if the notification fails to send (e.g., due to an invalid email or technical issue):    * The system logs the error and warns The Manager that the account was created but the notification failed.    * The Manager may manually notify the user or retry sending the notification.    * The process continues to step 8. 9. The Manager Cancels the Process 10. **Scenario**: After step 3, if The Manager decides not to proceed:     * The system discards the entered data and returns The Manager to the user management screen.     * The use case ends without creating an account. | | |

#### 2.2.10 Deactivate User

| Primary Actors | Owner | Secondary Actors | User |
| --- | --- | --- | --- |
| Description | This use case enables an Owner to deactivate a user’s account in the Pickleball Court Management System. Once deactivated, the user can no longer log in or access any system features, but their data remains stored for potential reactivation later. | | |
| Preconditions | * The Owner must be logged into the system with valid credentials. * The Owner must have sufficient permissions to manage user accounts. * The user account targeted for deactivation must exist in the system and be currently active. | | |
| Postconditions | * The user’s account is marked as deactivated, blocking their ability to log in or use the system. * The user’s data is retained in the system for possible future reactivation. | | |
| Normal Sequence/Flow | 1. **Access User Management:** The Owner navigates to the user management section of the system web interface. 2. **View list users:** The system displays a list of user with status. 3. **Select a user:** The Owner selects the activated user account from the list. 4. **Start Activation:** The Owner selects the activation button for the selected user 5. **Update Account Status:** The system changes the user's status from "Activated" to "Deactive". 6. **Show notification:** The system displays “ Cập nhật trạng thái thành công” . | | |
| Alternative Sequences/Flows | 1. **System Error During Activation**:  * In step 5, if the system encounters an error while updating the account status:   The system displays an error message ("Cập nhật trạng thái thất bại").  If the retry is successful, the use case continues from step 5; otherwise, it ends. | | |

#### 2.2.11 Activate User

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to activate a previously deactivated user account within the Pickleball Court Management System. Once activated, the user regains access to their account and associated functionalities (e.g., booking courts, viewing orders). | | |
| Preconditions | * The Owner must be logged into the system with valid credentials and have permissions to manage user accounts. * The user account to be activated must exist in the system and be in a deactivated state. | | |
| Postconditions | The selected user account is activated, and the user can log in and use the system as per their assigned role and permissions. | | |
| Normal Sequence/Flow | 1. **Access User Management:** The Owner navigates to the user management section of the system web interface. 2. **View list users:** The system displays a list of user with status. 3. **Select a user:** The Owner selects the deactivated user account from the list. 4. **Start Activation:** The Owner selects the activation button for the selected user 5. **Update Account Status:** The system changes the user's status from "Deactivated" to "Active". 6. **Show notification:** The system displays “ Cập nhật trạng thái thành công” . | | |
| Alternative Sequences/Flows | 1. **System Error During Activation**:  * In step 5, if the system encounters an error while updating the account status:   The system displays an error message ("Cập nhật trạng thái thất bại").  If the retry is successful, the use case continues from step 5; otherwise, it ends. | | |

#### 

#### 

### *2.3 Booking Management*

#### 2.3.1 Booking Service On Mobile

| Primary Actors | User, Guest | Secondary Actors | System Notification |
| --- | --- | --- | --- |
| Description | This use case allows users to purchase additional services via the mobile app (e.g. drinks, snacks,... ) instead of having to buy them directly at the counter. Users can select services, make payment through a generated QR code, and once the payment is successful, the order is sent to the staff for processing and fulfillment. | | |
| Preconditions | * The user is logged in (or is a guest). * There are available services listed in the system. * The system is connected to a valid payment processor for QR generation. | | |
| Postconditions | * The number of services is reduced as the user completes payment * A QR code is generated and scanned to complete payment. * The order status is updated to "Đã thanh toán" or "Đặt dịch vụ tại sân". * The service order appears in the user’s order history. | | |
| Normal Sequence/Flow | 1. User/Guest navigates to the Services section. 2. System displays available services (e.g., drinks, snacks). 3. User/Guest selects one or more services để tính tiền 4. User/Guest xem lại chi tiết dịch vụ và tổng tiền ứng với dịch vụ đã chọn. 5. System generates a QR code showing the total amount. 6. User/Guest scans the QR code with their banking/payment app. 7. System confirms payment. 8. The order is marked as paid and sent to staff (if applicable). 9. User/Guest receives a notification "Đặt dịch vụ tại sân thành công". | | |
| Alternative Sequences/Flows | 1. **No services available:**  The system displays “No services available at this time” 2. **User fails to scan QR code in time:**  QR code expires, and the user can no longer make the payment 3. **Order processing fails:**  The system will refund the money to the bank account used for the payment 4. **User places an order but hasn’t paid:**  The user can cancel the order or proceed with the payment 5. **Service sold out:**  The service will display a quantity of 0 | | |

#### 2.3.2 Change Booking

| Primary Actors | User | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a user to change an existing court reservation. The system supports one-time rescheduling, allowing users to change their court, date, or time within the allowed policy. The system handles payment differences automatically (e.g., generate new QR if extra cost, refund if lower). | | |
| Preconditions | * The user is authenticated (or a guest with an active booking session). * The user has at least one active booking that is eligible for modification. * The modification is made within the system’s allowed timeframe. * Only one update per booking is allowed. | | |
| Postconditions | * The booking is updated with new details (court/time/date). * If the new court costs more, a new QR code is generated for the additional amount. * If the new court costs less, the system initiates an automatic refund. | | |
| Normal Sequence/Flow | 1. User logs in and navigates to **Booking History**. 2. Selects a booking and clicks **“Change Booking”**. 3. System checks if the booking is eligible for modification (not expired, not modified before). 4. User selects new court, date, or time. 5. System compares the new booking price to the original:  * If equal: update is applied immediately. * If higher: system generates new QR for the additional amount. * If lower: system processes an automatic refund.  1. User confirms the changes. 2. System updates the booking and sends a confirmation message. | | |
| Alternative Sequences/Flows | 1. **Booking Already Modified Once:**  * System shows a message: “This booking has already been modified and cannot be updated again.”  1. **New Slot Unavailable:**  * System prevents selection and prompts the user to choose another time.  1. **Additional Payment Required:**  * QR code is generated. System waits for successful payment before confirming update.  1. **Price Reduction:**  * System notifies user that the difference will be refunded automatically.  1. **User Cancels Before Confirming:**  * No changes are saved; system returns to booking details.  1. **7A. Update Fails (Network/Server Issue):**  * Displays error message and allows retry. | | |

#### 2.3.3 View Booking History User by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to view the entire booking history of a user, for example, within a specific time period, by venue, by day, by month, etc. This information helps the owner track the customer's booking behavior and provide better services | | |
| Preconditions | * The Owner must be logged into the system using valid credentials. * The Owner must have appropriate permissions to access user information. | | |
| Postconditions | The system displays all booking history of the User, including details of previous and upcoming bookings as well as their payment information. | | |
| Normal Sequence/Flow | 1. Owner logs into the system. 2. Navigates to Order section. 3. Displays all booking history of the User | | |
| Alternative Sequences/Flows | 1. **No Booking History**:  * The system no displays data   2. **User Does Not Exist:**   * The system checks the database but does not find the corresponding User.   + The system displays an error message: “No user found. Please check” | | |

#### 

#### 2.3.4 View Booking History User by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to view the entire booking history of a user, for example, within a specific time period, by venue, by day, by month, etc. This information helps the owner track the customer's booking behavior and provide better services under the Owner's authority | | |
| Preconditions | * The Manager must be logged into the system using valid credentials. * The Manager must have appropriate permissions to access user information. | | |
| Postconditions | The system displays all booking history of the User, including details of previous and upcoming bookings as well as their payment information. | | |
| Normal Sequence/Flow | 1. Manager logs into the system. 2. Navigates to Order section. 3. Displays all booking history of the User under the Owner's authority | | |
| Alternative Sequences/Flows | 1. **No Booking History**:  * The system no displays data   2. **User Does Not Exist:**   * The system checks the database but does not find the corresponding User.   + The system displays an error message: “No user found. Please check” | | |

#### 

#### 2.3.5 Cancel Booking

| Primary Actors | User | Secondary Actors | System Notification Service |
| --- | --- | --- | --- |
| Description | This use case allows users to cancel their existing court bookings in the Pickleball Court Management System. Depending on the system's cancellation policy, canceling a booking may result in the loss of the deposit paid by the user at the time of booking. | | |
| Preconditions | * The user must have an active booking in the system. * The user must be logged in (for registered users) or provide booking details (for guests) to access the booking. | | |
| Postconditions | * The booking is canceled in the system, and the court becomes available for others to book. * If applicable, the user’s deposit is forfeited as per the cancellation policy. * The user receives a confirmation of the cancellation. | | |
| Normal Sequence/Flow | 1. **Access Booking Details**: The user navigates to the booking management section (e.g., “My Bookings” for registered users or a booking lookup for guests). 2. **Select Booking**: The system displays the user’s active bookings, and the user selects the specific booking to cancel. 3. **Initiate Cancellation**: The user chooses the option to cancel the selected booking. 4. **Display Cancellation Policy**: The system informs the user of the cancellation policy, including potential deposit loss (e.g., “Canceling this booking will result in the loss of your deposit.”). 5. **Confirm Cancellation**: The user confirms their intent to cancel the booking. 6. **Process Cancellation**: The system cancels the booking, updates the court availability, and marks the deposit as forfeited (if applicable). 7. **Notify User**: The system sends a confirmation to the user via email (e.g., “Your booking has been canceled. Your deposit has been forfeited.”). | | |
| Alternative Sequences/Flows | 1. **Cancellation Within Grace Period**:  * If the cancellation occurs within a predefined grace period (e.g., 24 hours before the booking time) where no deposit loss applies:   + - At step 4, the system informs the user that no deposit will be lost.     - At step 6, the deposit is refunded or not deducted, and the booking is canceled.     - The use case continues to step 7 with a notification (e.g., “Your booking has been canceled, and your deposit is safe.”).  1. **User Cancels Cancellation**:  * After step 4, if the user decides not to proceed with the cancellation:   + - The user selects an option to abort the cancellation (e.g., “Keep Booking”).     - The system returns the user to the booking management screen, and the booking remains active.     - The use case ends.  1. **Booking Not Found**:  * At step 2, if the selected booking cannot be located (e.g., already canceled or expired):   + - The system displays an error message (e.g., “Booking not found or already canceled.”).     - The user is prompted to select a different booking or refresh the list.     - The use case returns to step 1.  1. **Notification Failure**:  * At step 7, if the system fails to send the cancellation confirmation (e.g., due to an email service error):   + - The cancellation is still processed, and the system logs the failure.     - The user can view the cancellation status in the app, but no external notification is sent. | | |

#### 2.3.6 View Booking Personal

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to view a list of their own court bookings, including information such as booking date, time slot, court name, booking status, and payment status. Users can access and review both upcoming and past bookings in their personal dashboard. | | |
| Preconditions | * User or Guest is authenticated (or recognized by session token for guest booking). * There are existing bookings linked to the user’s account or session. | | |
| Postconditions | * The user successfully views a list of their bookings. * The system displays updated statuses (e.g., confirmed, canceled, completed). | | |
| Normal Sequence/Flow | 1. User/Guest logs into the system (or is already authenticated). 2. User/Guest navigates to **Personal Dashboard → My Bookings**. 3. System retrieves all bookings linked to the user/guest account. 4. System displays booking information including:  * Booking ID / reference number * Court name and location * Date and time slot * Booking status (Pending / Confirmed / Canceled / Completed) * Payment status (Paid / Deposit / Pending)  1. User can filter bookings by:  * Status * Date range  1. User clicks on a booking to view detailed information (optional). | | |
| Alternative Sequences/Flows | 1. **No Bookings Found:**  * System shows message: “No bookings found. Start booking your first court!”  1. **Booking Data Retrieval Error:**  * System shows error message: “Unable to load bookings. Please try again later.” | | |

#### 2.3.7 View Booking Detail Personal

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to view the detailed information of a specific court booking that they have made. The detail view includes comprehensive information about the court, booking time, services added (if any), payment status, and other relevant booking actions (e.g., eligible for cancellation or rescheduling). | | |
| Preconditions | * User or Guest is authenticated (or session is valid). * The booking record exists and is linked to the user or session. | | |
| Postconditions | * The system displays full details of the selected booking. * User can take available actions based on booking status (view only, cancel, reschedule if permitted). | | |
| Normal Sequence/Flow | 1. User/Guest navigates to **My Bookings**. 2. User/Guest selects a booking from the list. 3. System retrieves detailed information for the selected booking. 4. System displays booking details including:  * Booking ID / Reference Code * Court Name and Location * Booking Date and Time Slot * Booking Type (Daily, Regular, Optimized) * Booking Status (Pending, Confirmed, Completed, Canceled) * Payment Details (Amount paid, Deposit, Payment method, Payment status) * Services or Equipment Ordered (if any) * Cancellation/Rescheduling Options (if applicable)  1. User/Guest reviews the information. | | |
| Alternative Sequences/Flows | 1. **Booking Not Found:**  * System displays: “Booking not found or already removed.”  1. **Unauthorized Access:**  * System prevents access if the user tries to view a booking that does not belong to them. | | |

#### 2.3.8 Confirm Booking

| Primary Actors | User, Guest, Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest (or Staff on behalf of a customer) to confirm a court booking after initial creation. Confirmation can happen automatically after successful payment or manually if staff intervention is required. The confirmation secures the booking, reserves the court, and updates the booking status from "Pending" to "Confirmed." | | |
| Preconditions | * A booking has been created and is in a **Pending** status. * Required payment has been made (deposit or full payment as required by booking rules). * The court and time slot are still available. | | |
| Postconditions | * The booking status is updated to **Confirmed**. * The court time slot is locked for that user (unavailable to others). * A booking confirmation notification (email/SMS/in-app) is sent to the user. | | |
| Normal Sequence/Flow | 1. User completes the booking process and initiates payment (if not already done). 2. System verifies:  * Payment is successful (or deposit received). * Court/time slot availability remains unchanged.  1. System updates booking status to **Confirmed**. 2. System generates a booking confirmation message with booking details. 3. System sends a notification to the user (email/SMS/app). 4. User views confirmation in the **Booking Personal** or **Booking Detail** screens. | | |
| Alternative Sequences/Flows | 1. **Payment Fails:**  * System keeps the booking in **Pending** status and prompts user to retry payment.  1. **Court/Slot Becomes Unavailable Before Confirmation:**  * System informs user: “Selected court and time slot are no longer available.”  1. **Staff-initiated Confirmation:**  * Staff manually confirms a booking (e.g., walk-in customer) without requiring online payment (if applicable). | | |

#### 

#### 2.3.9 Deposit Booking

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a **User** or **Guest** to place a **deposit** to temporarily secure a court booking. By paying a predefined deposit amount instead of the full price, the system locks the booking until the remaining balance is paid (either before check-in or within a specified deadline). | | |
| Preconditions | * A booking has been created and is in **Pending** status. * Court and time slot are still available. * Deposit policy and amount are defined in the system. | | |
| Postconditions | * A deposit payment is successfully made. * The booking status is updated to **Deposit Paid** or **Partial Payment**. * The system records the deposit transaction and updates booking history. | | |
| Normal Sequence/Flow | 1. User/Guest creates a booking and selects **Deposit Payment** option. 2. System calculates and displays the required deposit amount (e.g., 30% of total booking value). 3. User/Guest confirms and proceeds to the payment gateway (QR code/payment link). 4. User completes the deposit payment. 5. System verifies payment success with the payment gateway. 6. System updates booking status to **Deposit Paid**. 7. System sends a confirmation notification to the user. 8. System schedules a reminder for full payment deadline (if applicable). | | |
| Alternative Sequences/Flows | 1. **Payment Failure:**  * System displays error: “Deposit payment failed. Please try again.”  1. **Payment Timeout:**  * If payment is not completed within X minutes, system cancels the booking.  1. **Deposit Option Disabled:**  * If deposit payment is not allowed for a particular booking (e.g., close to play time), system only allows full payment. | | |

#### 

#### 2.3.10 Payment Booking

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to pay for a court booking, either by completing the full amount at once or completing the remaining balance after a deposit. Successful payment finalizes the booking process, secures the court, and transitions the booking status to Confirmed. | | |
| Preconditions | * A booking exists with status **Pending** or **Deposit Paid**. * Court and time slot are still available. * Payment gateway is integrated and operational. | | |
| Postconditions | * Payment is successfully processed. * Booking status is updated to **Confirmed**. * System generates an invoice and transaction record. * Confirmation notification is sent to the user. | | |
| Normal Sequence/Flow | 1. User/Guest accesses My Bookings or completes the initial booking creation. 2. System prompts the user to Pay Now (full amount or remaining balance if deposit already paid). 3. User reviews payment details (amount, court name, booking date, time slot). 4. User selects payment method (e.g., QR Code, Bank Transfer, E-Wallet). 5. System redirects to the payment gateway. 6. User completes the payment. 7. System verifies payment success. 8. System updates booking status to Confirmed. 9. System generates and saves the payment invoice. 10. System sends confirmation notification (email/SMS/app) to user. | | |
| Alternative Sequences/Flows | 1. **Payment Failure:**  * System displays error: “Payment failed. Please try again.” * Booking remains in **Pending** or **Deposit Paid** status.  1. **Payment Timeout:**  * If no payment after X minutes, booking may expire or user is reminded to retry.  1. **Already Fully Paid:**  * If booking is already paid in full, system prevents duplicate payments. | | |

#### 2.3.11 Refund on cancellation

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to receive a refund after canceling a booking, based on the system's cancellation and refund policy. The refund amount may vary depending on whether the user paid in full or only made a deposit, and how close to the booking time the cancellation occurs. | | |
| Preconditions | * The booking was already **paid** (full payment or deposit). * The user initiates cancellation **within the allowable cancellation window** (if any). * System has the payment and banking information available to process the refund. | | |
| Postconditions | * The refund (full, partial, or none) is calculated and processed. * System updates the booking status to **Canceled**. * Refund transaction is recorded and a notification is sent to the user. | | |
| Normal Sequence/Flow | 1. User/Guest accesses **My Bookings** and selects a booking to cancel. 2. User clicks **Cancel Booking**. 3. System checks:  * Payment status * Cancellation policy (full refund, partial refund, deposit forfeited)  1. System calculates the eligible refund amount:  * **If full payment:** Refund = Total payment – Deposit (if deposit is non-refundable) * **If deposit only:** No refund or partial refund (depending on timing)  1. System sends refund request to the Bank API/Payment Gateway. 2. Bank processes the refund and returns confirmation. 3. System updates booking status to **Canceled**. 4. System logs the refund transaction. 5. User receives notification about refund processing status. | | |
| Alternative Sequences/Flows | 1. **Bank API Error/Failure:**   System retries or flags refund for manual processing.   1. **Cancellation Not Allowed (e.g., too close to booking time):**  * System informs the user: “Cancellation not eligible for refund at this time.”  1. **No Refund Eligible:**  * System shows message: “This booking is not eligible for a refund based on the policy.” | | |

#### 2.3.12 Refund for rescheduling

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to receive a partial refund when they reschedule a booking to a cheaper court or time slot. The system automatically calculates the price difference and initiates the refund process while retaining the original deposit amount (if applicable). | | |
| Preconditions | * The booking has been paid in full or deposit paid. * User successfully reschedules the booking to a cheaper court or cheaper time slot. * Payment and refund integration is active (Bank API or Payment Gateway). | | |
| Postconditions | * Refund for the price difference (excluding deposit) is processed automatically. * Booking is updated with new schedule details. * Refund transaction is logged and user is notified. | | |
| Normal Sequence/Flow | 1. User/Guest selects a booking and chooses the **Reschedule** option. 2. User selects a new court or time slot with **lower pricing**. 3. System calculates:  * Original Total Price * New Total Price * Refund Amount = (Original Total – New Total) – (Deposit, if deposit is non-refundable)  1. System initiates refund through the Bank API/Payment Gateway. 2. Refund is processed and confirmed. 3. System updates the booking with new details (court, time). 4. System logs the refund transaction. 5. User receives confirmation notification about the new booking and refund. | | |
| Alternative Sequences/Flows | 1. **New Court/Slot Has Higher or Equal Price:**  * No refund is processed. * If new price > old price, system requests additional payment.  1. **Refund Failure at Bank API:**  * System retries or flags for manual refund processing.  1. **New Slot No Longer Available:**  * System prompts the user to select another time or cancel reschedule request. | | |

#### 2.3.13 Refund without check in

#### 

| Primary Actors | User (indirectly), Admin (monitor only) | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables the system to **automatically detect** fully paid bookings where the user **did not check in** and the play time has passed. In such cases, the system **retains the deposit** and **refunds the remaining amount** via **MB Bank API**. This ensures fair handling of no-shows while maintaining deposit policy. | | |
| Preconditions | * Booking status = **Paid** * Check-in status = **Not Checked-in** * Current time > booking end time * MB Bank integration is configured * Booking is within eligible refund window for no-show policy | | |
| Postconditions | * Refund of (total payment - deposit) is processed via MB Bank * Booking is marked as **No-show / Refunded** * Deposit is retained as revenue * Admin and user are notified (optional) | | |
| Normal Sequence/Flow | 1. System runs **scheduled job** (e.g., every hour). 2. It filters for bookings where:  * status = paid * check-in status = not checked-in * current time > booking end time * refund\_status = pending  1. System calculates:  refund\_amount = total\_paid - deposit\_amount 2. If refund > 0, system sends refund request to **MB Bank API** with:  * User account * Amount * Booking reference  1. MB Bank returns status:  * Success → system sets refund\_status = completed * Fail → system sets refund\_status = failed and logs error  1. Booking is updated to **No-show / Refunded**, and notifications are sent. | | |
| Alternative Sequences/Flows | 1. **Refund Amount ≤ 0:**  * System logs: “No refund due – deposit retained.”  1. **MB Bank API Timeout/Error:**  * System retries; if still fails, marks status as **Refund Failed** and flags for manual review.  1. **Check-in Logged Late (After Slot):**  * System skips refund if check-in was eventually marked. | | |

#### 

#### 2.3.14 Book Court Daily

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User or Guest to book a court for a specific date and time (single session booking). This is the most common type of booking for casual players who do not need a recurring reservation. The system displays available courts, times, and corresponding pricing, and allows the user to complete the reservation by paying a deposit or full amount. | | |
| Preconditions | * The user or guest is authenticated (or identified via session for Guest). * Courts are available for booking. * Booking settings for daily reservations are active. | | |
| Postconditions | * Court is reserved for the selected date and time. * Booking status is created as **Pending** until payment is completed. * Upon successful payment, booking status changes to **Confirmed**. | | |
| Normal Sequence/Flow | 1. User/Guest navigates to **Book a Court** screen. 2. User selects:  * Date * Time Slot * Court (optional filters: indoor/outdoor, location, price)  1. System shows available courts based on selection. 2. User selects a court and reviews booking details (court, date, time, price). 3. User chooses payment method (Deposit or Full Payment). 4. System generates a booking record with **Pending** status. 5. User proceeds with payment. 6. System verifies payment:  * If successful → Booking status changes to **Confirmed**. * If failed → Booking remains Pending or is canceled after timeout.  1. System sends booking confirmation notification. | | |
| Alternative Sequences/Flows | 1. **No Courts Available:**  * System displays: “No courts available for the selected date and time.”  1. **Payment Failure:**  * System prompts user to retry payment or cancel booking.  1. **Session Timeout Before Payment:**  * System automatically cancels the pending booking if payment is not completed within a configured time window (e.g., 15 minutes). | | |

#### 2.3.15 Book Court Regular

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User, Guest to book a court for multiple fixed dates and times in a recurring manner (e.g., every Monday and Wednesday at 7:00 PM for 4 weeks). It is designed for players or groups who want a consistent weekly schedule without manually booking each session. | | |
| Preconditions | * User is authenticated (Guest cannot book regular recurring slots). * Court has availability for all selected recurrence dates. * Recurring booking feature is enabled in the system settings. | | |
| Postconditions | * A series of court bookings are created based on the user's selected schedule. * Each individual booking is recorded with its own ID and linked to a main recurring group ID. * System reserves the court for all requested dates and times. | | |
| Normal Sequence/Flow | 1. User navigates to **Book a Court** and selects **Regular Booking** option. 2. User specifies:  * Start date * Days of the week (e.g., Monday, Wednesday, Friday) * Time slot (e.g., 19:00 – 20:30) * Duration (number of weeks)  1. System checks court availability for all requested dates and time slots. 2. System displays:  * Summary of selected schedule * Total price (if paying full) or deposit amount  1. User confirms booking and proceeds to payment (Deposit or Full Payment). 2. System creates multiple linked booking entries:  * Each booking assigned a unique ID. * All bookings linked under a regular booking group.  1. Payment is processed and verified. 2. System sends booking confirmation for all reserved sessions. 3. Booking status for each session is updated accordingly. | | |
| Alternative Sequences/Flows | 1. **Some Slots Unavailable:**  * System notifies the user which dates are unavailable and offers options:    + Skip unavailable dates.   + Choose alternative slots.   + Cancel the recurring booking process.  1. **Payment Failure:**  * System keeps all bookings in Pending state until payment is completed.  1. **User Cancels During Setup:**  * Booking setup is canceled without creating any entries. | | |

#### 2.3.16 Book Court Optimal

| Primary Actors | User/Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows a User/Guest to request a multi-day or recurring booking where the system will automatically find and assign the most optimal available courts if the User/Guest preferred court is not fully available across all dates. The system ensures that even if a specific court is fully booked on some dates, it substitutes with a similar available court so that the User/Guest can maintain a consistent playing schedule without manual rebooking. | | |
| Preconditions | * User selects the desired schedule across multiple dates. * Optimal booking feature is enabled in the system. | | |
| Postconditions | * A combination of bookings is created across different courts (if necessary). * Courts are reserved for all requested dates without manual intervention. * User/Guest is notified of any changes from their preferred court selections. | | |
| Normal Sequence/Flow | 1. User/Guest navigates to **Book a Court** and selects **Optimal Booking** option. 2. User/Guest specifies:  * Preferred court (optional) * Date range or recurring schedule (e.g., every Monday and Thursday for 6 weeks) * Preferred time slot  1. System attempts to reserve the preferred court for all selected dates. 2. If any conflicts occur:  * System automatically finds the most similar available courts based on criteria like location, indoor/outdoor, price range.  1. System generates a booking summary showing:  * Dates and courts assigned. * Indication of which dates had substitutions.  1. User/Guest reviews and confirms the booking schedule. 2. User/Guest proceeds to payment (deposit or full payment). 3. System verifies payment. 4. System finalizes and confirms all booking entries. 5. Notification is sent detailing full booking results. | | |
| Alternative Sequences/Flows | 1. **No Suitable Alternative Courts Found:**  * System alerts User/Guest and allows them to:    + Adjust schedule.   + Try a different time slot.   + Cancel the booking process.  1. **Payment Failure:**  * Bookings are held temporarily but not finalized until payment is completed.  1. **User Rejects Alternative Proposal:**  * User/Guest can cancel the entire optimal booking attempt. | | |

#### 2.3.17 Hold Court While Booking

| Primary Actors | User/Guest | Secondary Actors | Payment Gateway |
| --- | --- | --- | --- |
| Description | This use case allows the system to temporarily hold a pickleball court for a User/Guest during the booking process, ensuring the court remains reserved while the user completes payment. The hold prevents other users from booking the same court until the payment is confirmed or the hold expires. | | |
| Preconditions | * The User/Guest must have selected a specific court, date, and time slot for booking. * The selected court and time slot must be available at the time of initiating the booking. * The system must support a predefined hold duration (e.g., 10 minutes). | | |
| Postconditions | * If payment is completed successfully within the hold duration, the court is officially booked for the User/Guest . * If payment is not completed within the hold duration, the court hold is released, and the court becomes available again for other User/Guest . | | |
| Normal Sequence/Flow | 1. **Initiate Booking**: The User/Guestselects a court, date, and time slot from the available options and chooses to proceed with booking. 2. **System Holds Court**: The system temporarily reserves the selected court and time slot, marking it as "on hold" and unavailable to other User/Guest. 3. **Display Payment Prompt**: The system redirects the User/Guest to the payment interface and informs them of the hold duration (e.g., "Court held for 10 minutes"). 4. **User Completes Payment**: The user enters payment details (e.g., via an integrated payment method like MB Bank) and submits the payment. 5. **Payment Gateway Confirms**: The payment gateway processes the transaction and sends a confirmation to the system. 6. **Finalize Booking**: Upon successful payment confirmation, the system converts the temporary hold into a confirmed booking and notifies the user (e.g., via email ). 7. **Release Hold (if applicable)**: If the payment is completed within the hold duration, the process ends with the court booked. | | |
| Alternative Sequences/Flows | 1. **Payment Timeout**:  * If the user does not complete payment within the hold duration (e.g., 10 minutes) at step 4:   + - The system releases the hold on the court, making it available again for other User/Guest .     - The User/Guest receives a notification (e.g., "Your court hold has expired. Please try booking again.").     - The use case ends without a confirmed booking.  1. **Payment Failure**:  * At step 5, if the payment gateway returns a failure (e.g., insufficient funds or invalid card):   + - The system informs the User/Guest of the payment failure and provides an option to retry with a different payment method.     - If the hold duration has not expired, the User/Guest can attempt payment again, returning to step 4.     - If the hold duration expires during retries, the system releases the court, and the user is notified to start over.  1. **User Cancels Booking**:  * At any point before payment confirmation (e.g., during step 4), if the user chooses to cancel the booking:   + - The system immediately releases the court hold.     - The user is returned to the court selection screen, and the use case ends.  1. **System Error During Hold**:  * If the system fails to hold the court in step 2 (e.g., due to a technical glitch):   + - The system displays an error message (e.g., "Unable to hold the court. Please try again.").     - The user is prompted to retry the booking process from step 1. | | |

#### 

#### 

#### 2.3.19 View Reservation

| Primary Actors | User, Guest, Owner, Staff, Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case describes the process in which a user selects a specific pickleball court to view more information or proceed to booking. This is the first step in the booking process, where the user chooses a court based on location, availability, or preference. | | |
| Preconditions | * The user is logged into the system (or is a guest with limited access). * The system has active courts listed and available to view. * The user has access to the Home or Court List screen. | | |
| Postconditions | * The user is navigated to the Court Detail screen. * The selected court information is loaded and displayed. | | |
| Normal Sequence/Flow | 1. User opens the **Home Screen** or **Court Listing Page**. 2. System displays a list of available courts with filters (by name, location, availability, favorite). 3. User browses through the list and selects a specific court. 4. System navigates the user to the **Court Detail Page**. 5. System displays:  * Court name, address, phone number * Images and description * Services offered * Map link * Available booking slots | | |
| Alternative Sequences/Flows | 1. **No Courts Available:**  * System displays a message: “No courts found. Try adjusting your filters.”  1. **Court Unavailable (disabled or under maintenance):**  * System shows status (e.g., “Under Maintenance”) and prevents booking but still allows viewing.  1. **User Not Logged In (Guest):**  * System allows viewing court details but prompts login when proceeding to book.  1. **System Error / Timeout:**  * Displays an error message and offers a retry or reload option. | | |

#### 

#### 

#### 

#### 

#### 

#### 

### *2.4 Court Management*

#### 

#### 2.4.1 View Schedule occupancy analysis by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Owner to view analysis and reports on court occupancy rates based on schedules. It provides insights into how efficiently the courts are being utilized over time (by day, by time slot, or by specific courts) to help optimize operations and maximize revenue. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * Historical booking data exists in the system. * Analytics module is active and updated. | | |
| Postconditions | * Owner/Manager successfully views occupancy data visualizations. * System displays occupancy trends by court, by date, and by time slot. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Analytics → Occupancy Analysis** section. 2. User selects:  * Date range (e.g., last 7 days, this month) * Specific court(s) or all courts * Time frames (optional filter: morning, afternoon, evening)  1. System retrieves and processes booking data. 2. System displays:  * Occupancy percentage per court and per time slot. * Trends over days/weeks. * Visual charts (bar graphs, heat maps).  1. Owner reviews the report and optionally exports it (Excel, PDF). | | |
| Alternative Sequences/Flows | 1. **No Data Available:**  * System displays: “No occupancy data found for the selected period.”  1. **Export Failure:**  * System shows error: “Unable to export report. Please try again later.” | | |

#### 2.4.2 View Schedule occupancy analysis by Manager

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an Manager to view analysis and reports on court occupancy rates based on schedules. It provides insights into how efficiently the courts are being utilized over time (by day, by time slot, or by specific courts) to help optimize operations and maximize revenue under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * Historical booking data exists in the system. * Analytics module is active and updated. | | |
| Postconditions | * Manager successfully views occupancy data visualizations. * System displays occupancy trends by court, by date, and by time slot. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Analytics → Occupancy Analysis** section. 2. User selects:  * Date range (e.g., last 7 days, this month) * Specific court(s) or all courts * Time frames (optional filter: morning, afternoon, evening)  1. System retrieves and processes booking data. 2. System displays:  * Occupancy percentage per court and per time slot. * Trends over days/weeks. * Visual charts (bar graphs, heat maps).  1. Manager reviews the report and optionally exports it (Excel, PDF). | | |
| Alternative Sequences/Flows | 1. **No Data Available:**  * System displays: “No occupancy data found for the selected period.”  1. **Export Failure:**  * System shows error: “Unable to export report. Please try again later.” | | |

#### 

#### 2.4.3 View Peak hours analysis by Owner

| Primary Actors | Owner | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to analyze **peak booking hours** across all courts or specific locations. The system provides visual insights such as bar charts to show which hours of the day (and which days of the week) have the highest demand, helping facilities optimize pricing, staff allocation, and slot availability. | | |
| Preconditions | * The actor is authenticated as **Owner** * Booking data is available for the selected timeframe. * The analytics module is functioning properly. | | |
| Postconditions | * A visual analysis of peak hours is shown. * Actors can identify busiest time slots by day/hour/court. * The data can be exported for further planning. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Analytics**. 3. Selects filters:  * Time range * Court Name  1. System processes booking data. 2. Displays analytics including:  * Hourly booking frequency * Most and least booked time slots * Bar Chart by day/hour * Comparison with previous periods (optional)  1. Actor uses this data to:  * Optimize pricing (e.g., dynamic pricing for peak hours) * Reallocate staff during busy times * Offer promotions for off-peak hours  1. Optionally clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Booking Data:**  * System displays: “No data ”  1. **Chart Generation Error:**  * System falls back to table view and logs the error.  1. **No Distinct Peak Patterns:**  * System highlights low variation and suggests broader time range. | | |

#### 2.4.4 View Peak hours analysis by Manager

| Primary Actors | Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to analyze **peak booking hours** across all courts or specific locations. The system provides visual insights such as bar charts to show which hours of the day (and which days of the week) have the highest demand, helping facilities optimize pricing, staff allocation, and slot availability under the Owner's authority | | |
| Preconditions | * The actor is authenticated as **Manager**. * Booking data is available for the selected timeframe. * The analytics module is functioning properly. | | |
| Postconditions | * A visual analysis of peak hours is shown. * Actor can identify busiest time slots by day/hour/court. * The data can be exported for further planning. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Analytics**. 3. Selects filters:  * Time range * Court Name  1. System processes booking data. 2. Displays analytics including:  * Hourly booking frequency * Most and least booked time slots * Bar Chart by day/hour * Comparison with previous periods (optional)  1. Actor uses this data to:  * Optimize pricing (e.g., dynamic pricing for peak hours) * Reallocate staff during busy times * Offer promotions for off-peak hours  1. Optionally clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Booking Data:**  * System displays: “No data ”  1. **Chart Generation Error:**  * System falls back to table view and logs the error.  1. **No Distinct Peak Patterns:**  * System highlights low variation and suggests broader time range. | | |

#### 2.4.5 Create Court

| Primary Actors | Owner | Secondary Actors | System |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to add a new court to the system, including details such as court name location and email, location, website, open time. The court will then be available for booking by users. | | |
| Preconditions | * The actor is logged in with the **Owner** role. * The court management module is active and accessible. * Required fields (e.g., court name, type) are filled in. | | |
| Postconditions | * A new court is added to the database. * The court is visible in the **booking interface** for users. * The court appears in the **court management dashboard** for future edits or updates. | | |
| Normal Sequence/Flow | 1. Owner/Manager logs in to the admin dashboard. 2. Navigates to **Court Management → Add New Court**. 3. Fills in required details:  * Court Name * Address * Phone Number * Open Time * Email  1. Defines:  * Status (Active/Inactive)  1. Clicks **“Save”**. 2. System validates input and creates court record. 3. Confirmation message is shown: "Success." | | |
| Alternative Sequences/Flows | 1. **Required Field Missing:**  * System prompts: "Please complete all required fields."  1. **System Error on Save:**  * Error message displayed; user may retry. | | |

#### 2.4.6 Update Court

| Primary Actors | Owner | Secondary Actors | System |
| --- | --- | --- | --- |
| Description | This use case allows an Owner to update the information of an existing court, including details like court name, type (indoor/outdoor), surface type, pricing, operational status, and maintenance notes. Keeping court information up to date ensures accurate booking management and player communication. | | |
| Preconditions | * The user is authenticated and has the role of **Owner** * The court to be updated already exists in the system. * Court update feature is available and the court is not locked by active critical transactions (e.g., booking modification in process). | | |
| Postconditions | * The selected court’s information is updated in the system. * Changes are reflected immediately in court listings and booking screens. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Management → Court List**. 2. User selects a specific court to update. 3. System displays current court details (name, type, description, pricing, availability, etc.). 4. Owner edits the desired fields (e.g., updates court type to indoor, changes pricing). 5. User saves the changes. 6. System validates the updated information. 7. System updates court record in the database. 8. System confirms the update with a success message. | | |
| Alternative Sequences/Flows | 1. **Validation Error (Missing Fields/Invalid Data):**  * System highlights errors and requests correction (e.g., invalid price format, missing court name).  1. **Update Failure (Database Error):**  * System displays error message: “Unable to save changes. Please try again later.”  1. **Court Not Found or Locked:**  * System notifies: “This court cannot be updated at the moment.” | | |

#### 2.4.7 Active Court

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to activate a court that was previously set as **inactive**, making it visible and available for booking by users. This is commonly used after court maintenance or administrative review. | | |
| Preconditions | * The actor is authenticated as **Owner**. * The court exists in the system and is currently set to **Inactive**. * The court has required configurations (e.g., time slots, pricing). | | |
| Postconditions | * The court status is changed to **Active**. * The court becomes visible in the user-facing interface. * The court is now bookable. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Court Management** 3. Clicks on the **"Edit"** button. 4. Clicks on the **"Activate"** button for a specific court. 5. Clicks **“Save”**. 6. System validates changes and updates the database. 7. Success message is shown: "Success." | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Confirmation:**  * No changes are made.  1. **System Error on Status Update:**  * Error message is shown: “Fail.” | | |

#### 

#### 

#### 

#### 2.4.8 Deactivate Court

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to deactivate a court, making it unavailable for user bookings. This action is typically taken when the court is under maintenance, unavailable due to events, or retired from regular use. Deactivated courts are not visible to users but remain in the system for editing or future reactivation. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The court exists in the system and is currently marked as **Active**. * The court is not tied to any pending or ongoing bookings. | | |
| Postconditions | * The court status is set to **Inactive**. * The court is hidden from the user booking interface. * The court remains accessible in admin view for editing or reactivation. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Court Management** 3. Clicks on the **"Edit"** button. 4. Clicks on the **"Deactivate"** button for a specific court. 5. Clicks **“Save”**. 6. System validates changes and updates the database. 7. Success message is shown: "Success." | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Action:**  * No changes are made.  1. **System Error During Update:**  * Error message is shown: “Fail.” | | |

#### 

#### 2.4.9 Create Court Slot by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Owner** to define available **time slots** for each court, including start/end times, duration, and availability rules (e.g., day of week, custom pricing). Once created, these time slots appear in the booking calendar for users. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The time slot module is active and accessible. | | |
| Postconditions | * New time slots are associated with a court and stored in the database. * These time slots appear on the **booking interface** for users. * Slot availability is ready for pricing and status updates. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Sub**-**Court Management** Clicks **“Add Time Slot”**. 3. Fills in:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Add”**. 2. System validates time range and overlap. 3. Slot is saved and visible in the time slot list for that court. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.10 Create Court Slot by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Manager** to define available **time slots** for each court, including start/end times, duration, and availability rules (e.g., day of week, custom pricing). Once created, these time slots appear in the booking calendar for users under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * The time slot module is active and accessible. | | |
| Postconditions | * New time slots are associated with a court and stored in the database. * These time slots appear on the **booking interface** for users. * Slot availability is ready for pricing and status updates. | | |
| Normal Sequence/Flow | 1. Manager logs into the admin dashboard. 2. Navigates to **Sub**-**Court Management** Clicks **“Add Time Slot”**. 3. Fills in:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Add”**. 2. System validates time range and overlap. 3. Slot is saved and visible in the time slot list for that court. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.11 Update Court Slot by Owner

| Primary Actors | Owner, | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to update information related to a specific court slot, including its start time, end time, price, availability status, or applicable days. Updates are reflected immediately and can affect upcoming booking availability. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * At least one court and one time slot already exist in the system. * The slot to be updated is not locked by an active or confirmed booking (depending on system policy). | | |
| Postconditions | * The selected time slot is updated with new configurations. * The updated slot is reflected in the booking interface (if still active). * Booking conflicts (if any) are handled according to system rules. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. Selects an existing slot and clicks **“Edit”**. 4. Updates one or more fields:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Update”**. 2. System validates updated time range and potential overlaps. 3. Changes are saved and confirmation is displayed. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.12 Update Court Slot by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to update information related to a specific court slot, including its start time, end time, price, availability status, or applicable days. Updates are reflected immediately and can affect upcoming booking availability under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * At least one court and one time slot already exist in the system. * The slot to be updated is not locked by an active or confirmed booking (depending on system policy). | | |
| Postconditions | * The selected time slot is updated with new configurations. * The updated slot is reflected in the booking interface (if still active). * Booking conflicts (if any) are handled according to system rules. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. Selects an existing slot and clicks **“Edit”**. 4. Updates one or more fields:  * Slot start time (e.g., 06:00 AM) * Slot end time (e.g., 07:00 AM) * Repetition (daily/weekday/weekend)  1. Clicks **“Update”**. 2. System validates updated time range and potential overlaps. 3. Changes are saved and confirmation is displayed. | | |
| Alternative Sequences/Flows | 1. **System Error During Save:**  * Displays error and logs the issue. | | |

#### 2.4.13 Active Sub-Court by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **activate a sub-court** (a smaller unit or division of a main court) to make it available for booking. A sub-court may be deactivated temporarily for maintenance or scheduling purposes, and activating it again restores its visibility and bookability for users. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The sub-court exists in the system and is currently in an **Inactive** state. * No conflicts exist (e.g., sub-court not under ongoing maintenance or reserved for exclusive events). | | |
| Postconditions | * The sub-court is now marked as **Active**. * The sub-court becomes available for booking immediately or based on an effective date. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Inactive Sub-Courts**. 3. User selects a sub-court to activate. 4. System displays the current status and related information. 5. User clicks **Activate** button. 6. System updates the sub-court status to **Active**. 7. Sub-court becomes available on the booking platform. 8. System logs the activation action. | | |
| Alternative Sequences/Flows | 1. **Validation Error:**  * System prevents activation if maintenance is ongoing or scheduled. * System shows warning: “Sub-court cannot be activated due to ongoing maintenance.”  1. **Update Failure (Database Issue):**  * System displays: “Activation failed. Please try again later.” | | |

#### 2.4.14 Active Sub-Court by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Manager** to **activate a sub-court** (a smaller unit or division of a main court) to make it available for booking. A sub-court may be deactivated temporarily for maintenance or scheduling purposes, and activating it again restores its visibility and bookability for users under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The sub-court exists in the system and is currently in an **Inactive** state. * No conflicts exist (e.g., sub-court not under ongoing maintenance or reserved for exclusive events). | | |
| Postconditions | * The sub-court is now marked as **Active**. * The sub-court becomes available for booking immediately or based on an effective date. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Inactive Sub-Courts**. 3. User selects a sub-court to activate. 4. System displays the current status and related information. 5. User clicks **Activate** button. 6. System updates the sub-court status to **Active**. 7. Sub-court becomes available on the booking platform. 8. System logs the activation action. | | |
| Alternative Sequences/Flows | 1. **Validation Error:**  * System prevents activation if maintenance is ongoing or scheduled. * System shows warning: “Sub-court cannot be activated due to ongoing maintenance.”  1. **Update Failure (Database Issue):**  * System displays: “Activation failed. Please try again later.” | | |

#### 2.4.15 Deactivate Sub-Court by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **deactivate a sub-court**, temporarily removing it from the booking system. This can be necessary for reasons such as scheduled maintenance, operational adjustments, or special events requiring full-court usage. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The sub-court exists in the system and is currently in an **Active** state. * No active bookings conflict with the deactivation timeframe (or a rescheduling/cancellation policy is applied). | | |
| Postconditions | * The selected sub-court is updated to **Inactive** status. * The sub-court is no longer available for users to book. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Active Sub-Courts**. 3. User selects the sub-court to deactivate. 4. System displays the current status and sub-court details. 5. User clicks the **Deactivate** button and confirms the action. 6. System checks for upcoming bookings:  * If none → Proceed. * If existing bookings → Prompt user to handle them (cancel or reschedule).  1. System updates sub-court status to **Inactive**. 2. System logs the deactivation action. 3. System sends notification if needed (to affected users, if applicable). | | |
| Alternative Sequences/Flows | 1. **Existing Bookings Conflict:**  * System blocks deactivation and shows warning:  “Sub-court has active bookings. Please resolve bookings before deactivation.”  1. **Deactivation Aborted by User:**  * If user cancels at confirmation step, no changes are made.  1. **Database Error:**  * System displays: “Unable to deactivate sub-court. Please try again later.” | | |

#### 2.4.16 Deactivate Sub-Court by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Manager** to **deactivate a sub-court**, temporarily removing it from the booking system. This can be necessary for reasons such as scheduled maintenance, operational adjustments, or special events requiring full-court usage under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The sub-court exists in the system and is currently in an **Active** state. * No active bookings conflict with the deactivation timeframe (or a rescheduling/cancellation policy is applied). | | |
| Postconditions | * The selected sub-court is updated to **Inactive** status. * The sub-court is no longer available for users to book. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Court Management → Sub-Court List**. 2. User filters or searches for **Active Sub-Courts**. 3. User selects the sub-court to deactivate. 4. System displays the current status and sub-court details. 5. User clicks the **Deactivate** button and confirms the action. 6. System checks for upcoming bookings:  * If none → Proceed. * If existing bookings → Prompt user to handle them (cancel or reschedule).  1. System updates sub-court status to **Inactive**. 2. System logs the deactivation action. 3. System sends notification if needed (to affected users, if applicable). | | |
| Alternative Sequences/Flows | 1. **Existing Bookings Conflict:**  * System blocks deactivation and shows warning:  “Sub-court has active bookings. Please resolve bookings before deactivation.”  1. **Deactivation Aborted by User:**  * If user cancels at confirmation step, no changes are made.  1. **Database Error:**  * System displays: “Unable to deactivate sub-court. Please try again later.” | | |

#### 2.4.17 Create Maintenance by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to schedule maintenance periods for specific courts or court areas. During the maintenance window, the court will be **unavailable for booking**. The maintenance record includes a description, duration, and affected court(s), and ensures proper scheduling and communication to users and staff. | | |
| Preconditions | * The actor is authenticated as **Owner**. * At least one court exists in the system. * The maintenance scheduling module is available. | | |
| Postconditions | * A maintenance entry is created and associated with one or more courts. * The court becomes unavailable for booking during the maintenance window. * Users are prevented from booking and (optionally) receive notification of closure. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Maintenance**. 3. Select  * Select court * Select sub-court  1. Fills in the maintenance form:  * Start date & time * End date & time * Description  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Confirmation message is shown: “Tạo lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **System Error on Save:**  * System shows: “Tạo lịch bảo trì thất bại.” | | |

#### 2.4.18 Create Maintenance by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to schedule maintenance periods for specific courts or court areas. During the maintenance window, the court will be **unavailable for booking**. The maintenance record includes a description, duration, and affected court(s), and ensures proper scheduling and communication to users and staff under the Owner's authority | | |
| Preconditions | * The actor is authenticated as **Manager**. * At least one court exists in the system. * The maintenance scheduling module is available. | | |
| Postconditions | * A maintenance entry is created and associated with one or more courts. * The court becomes unavailable for booking during the maintenance window. * Users are prevented from booking and (optionally) receive notification of closure. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Maintenance**. 3. Select  * Select court * Select sub-court  1. Fills in the maintenance form:  * Start date & time * End date & time * Description  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Confirmation message is shown: “Tạo lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **System Error on Save:**  * System shows: “Tạo lịch bảo trì thất bại.” | | |

#### 2.4.19 Update Maintenance by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to modify an existing maintenance record for a court, including changes to time, description, or affected court area. Updating maintenance ensures accuracy and allows for changes due to scheduling conflicts or work delays. | | |
| Preconditions | * The actor is authenticated as **Owner** * At least one maintenance entry exists in the system. * The maintenance record has not yet expired (optional policy). | | |
| Postconditions | * The maintenance entry is updated with new details. * The booking system reflects the new maintenance window. * Users are blocked from booking during the updated period. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → Maintenance List**. 3. Select  * Select court * Select sub-court  1. Click”Edit” 2. Fills in the maintenance form:  * Start date & time * End date & time * Description * Status  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Success message is shown: “Cập nhật lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **Save Error:**  * System displays error message and allows retry. | | |

#### 2.4.20 Update Maintenance by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to modify an existing maintenance record for a court, including changes to time, description, or affected court area. Updating maintenance ensures accuracy and allows for changes due to scheduling conflicts or work delays under the Owner's authority | | |
| Preconditions | * The actor is authenticated as **Manager**. * At least one maintenance entry exists in the system. * The maintenance record has not yet expired (optional policy). | | |
| Postconditions | * The maintenance entry is updated with new details. * The booking system reflects the new maintenance window. * Users are blocked from booking during the updated period. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → Maintenance List**. 3. Select  * Select court * Select sub-court  1. Click”Edit” 2. Fills in the maintenance form:  * Start date & time * End date & time * Description * Status  1. Clicks **“Ok”**. 2. System validates that the maintenance window does not conflict with existing bookings. 3. Success message is shown: “Cập nhật lịch bảo trì thành công” | | |
| Alternative Sequences/Flows | 1. **Save Error:**  * System displays error message and allows retry. | | |

#### 2.4.21 View Maintenance by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to view a list of all scheduled and completed maintenance records for courts or court areas. Maintenance entries include time period, status, description, and affected courts. This supports operational visibility and planning. | | |
| Preconditions | * The actor is logged in with **Owner** role. * At least one court maintenance record exists in the system. | | |
| Postconditions | * A list of maintenance records is displayed. * Each record can be filtered, searched, and optionally expanded for detail. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → View Maintenance**. 3. System displays a list or table of maintenance entries:  * Select court * Select sub-court | | |
| Alternative Sequences/Flows | 1. **No Maintenance Found:**  * System shows: “Trống.”  1. **Entry Load Failure:**  * Error message shown with retry option. | | |

#### 2.4.22 View Maintenance by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to view a list of all scheduled and completed maintenance records for courts or court areas. Maintenance entries include time period, status, description, and affected courts. This supports operational visibility and planning under the Owner's authority | | |
| Preconditions | * The actor is logged in with **Manager** role. * At least one court maintenance record exists in the system. | | |
| Postconditions | * A list of maintenance records is displayed. * Each record can be filtered, searched, and optionally expanded for detail. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Maintenance → View Maintenance**. 3. System displays a list or table of maintenance entries:  * Select court * Select sub-court | | |
| Alternative Sequences/Flows | 1. **No Maintenance Found:**  * System shows: “Trống.”  1. **Entry Load Failure:**  * Error message shown with retry option. | | |

#### 2.4.23 Create Court Image by Owner

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to upload and assign one or more images to a specific court. These images will be displayed on the booking interface, giving users a visual preview of the court (e.g., surface, lighting, surroundings). | | |
| Preconditions | * The actor is authenticated with the **Owner** role. * The court already exists in the system. * The image file meets system format and size requirements. | | |
| Postconditions | * One or more images are associated with a court. * Images are visible to users when viewing or booking the court. * Images can be managed (edited or deleted) later. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court → Manage Images**. 3. Clicks **“Select File”**. 4. Selects image file(s) from local device. 5. Clicks **“Upload”**. 6. System validates the file (format, size) and associates it with the selected court. 7. Confirmation message is displayed:  “Create Successfully” | | |
| Alternative Sequences/Flows | 1. **Upload Failure:**  * System logs error and shows: “Fail.” | | |

#### 

#### 2.4.24 Create Court Image by Manager

#### 

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to upload and assign one or more images to a specific court. These images will be displayed on the booking interface, giving users a visual preview of the court (e.g., surface, lighting, surroundings) under the Owner's authority | | |
| Preconditions | * The actor is authenticated with the **Manager** role. * The court already exists in the system. * The image file meets system format and size requirements. | | |
| Postconditions | * One or more images are associated with a court. * Images are visible to users when viewing or booking the court. * Images can be managed (edited or deleted) later. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court → Manage Images**. 3. Clicks **“Select File”**. 4. Selects image file(s) from local device. 5. Clicks **“Upload”**. 6. System validates the file (format, size) and associates it with the selected court. 7. Confirmation message is displayed:  “Create Successfully” | | |
| Alternative Sequences/Flows | 1. **Upload Failure:**  * System logs error and shows: “Fail.” | | |

#### 2.4.25 Delete Court Image by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts. | | |
| Preconditions | * The actor is authenticated as **Owner** * The image exists and is currently associated with a valid court. * The image is **not the only image** remaining if court requires at least one display image. | | |
| Postconditions | * The selected image is deleted from the system. * The court no longer displays the deleted image in the booking interface. * If the image was marked as default, the system may assign another default automatically. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court** 3. Views image gallery and clicks **“Delete”** on the chosen image. 4. System prompts:  “Are you sure you want to delete this item? ” 5. Actor confirms deletion. 6. System removes the image from storage and updates the database. 7. Success message is displayed:  “ Deleted successfully.” | | |
| Alternative Sequences/Flows | 1. **Deletion Error:**  * System displays: “Failed to delete image. Please try again.” | | |

#### 2.4.26 Delete Court Image by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to delete one or more images associated with a specific court. Removing outdated or irrelevant images ensures the court’s visual representation is accurate and up to date for users browsing or booking courts. | | |
| Preconditions | * The actor is authenticated as **Manager**. * The image exists and is currently associated with a valid court. * The image is **not the only image** remaining if court requires at least one display image. | | |
| Postconditions | * The selected image is deleted from the system. * The court no longer displays the deleted image in the booking interface. * If the image was marked as default, the system may assign another default automatically. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Court Management → Select Court** 3. Views image gallery and clicks **“Delete”** on the chosen image. 4. System prompts:  “Are you sure you want to delete this item? ” 5. Actor confirms deletion. 6. System removes the image from storage and updates the database. 7. Success message is displayed:  “ Deleted successfully.” | | |
| Alternative Sequences/Flows | 1. **Deletion Error:**  * System displays: “Failed to delete image. Please try again.” | | |

#### 2.4.27 View Court Image by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to view images. The court image provides visual confirmation and allows the court Owner to be able to upload the latest court image of the field to the user visually. | | |
| Preconditions | * The court exists in the system. * The system must be allow Owner have permission to upload image about court | | |
| Postconditions | * The court images are displayed successfully. * Owner can browse through all uploaded images for the court. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Court Image** page. 2. Owner selects a specific court to view more details. 3. System retrieves all uploaded images for the selected court. 4. System displays images in a gallery or carousel view:  * Thumbnails * Full-size images (on click/tap)  1. Owner browses through available court images. | | |
| Alternative Sequences/Flows | 1. **No Images Uploaded for Court:**  * System displays placeholder image and message:  “No images available for this court yet.”  1. **Image Load Failure:**  * System shows error message: “Unable to load court images. Please try again later.” | | |

#### 2.4.28 View Court Image by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to view images. The court image provides visual confirmation and allows the court Manager to be able to upload the latest court image of the field to the user visually. | | |
| Preconditions | * The court exists in the system. * The system must be allow Manager have permission to upload image about court | | |
| Postconditions | * The court images are displayed successfully. * Manager can browse through all uploaded images for the court. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Court Image** page. 2. Manager selects a specific court to view more details. 3. System retrieves all uploaded images for the selected court. 4. System displays images in a gallery or carousel view:  * Thumbnails * Full-size images (on click/tap)  1. Manager browses through available court images. | | |
| Alternative Sequences/Flows | * **No Images Uploaded for Court:** * System displays placeholder image and message:  “No images available for this court yet.” * **Image Load Failure:** * System shows error message: “Unable to load court images. Please try again later.” | | |

#### 2.4.29 View Court Details

| Primary Actors | User/Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows User/Guest to view detailed information about a pickleball court, including its specifications (e.g., type, location, availability), user reviews, and images, to help them make an informed decision before booking. | | |
| Preconditions | * The system must have at least one court registered with available details, reviews, or images. * The User/Guest must have access to the court listing section of the system (guest and user can additional features like saved preferences). | | |
| Postconditions | The User/Guest has viewed the court’s details, reviews, and images, and can proceed to book the court or explore other options. | | |
| Normal Sequence/Flow | 1. **Access Court Listings**: The User/Guest has viewed the court’s details, reviews, and images, and can proceed to book the court or explore other options. 2. **Select a Court**: The User/Guest selects a specific court from the available list. 3. **Display Court Details**: The system displays detailed information about the selected court, location, pricing by time slot, and availability 4. **Show Reviews**: The system presents user-generated reviews and ratings for the court (if available). 5. **Show Images**: The system displays images of the court (e.g., uploaded by the owner or staff) for the user to view. 6. **Interact with Content**: The user scrolls through the details, and view images to evaluate the court. | | |
| Alternative Sequences/Flows | 1. **No Images Available**:  * At step 5, if no images have been uploaded for the court:   + - The system displays a placeholder image or a message like "No images available at this time."     - The use case continues with step 6.  1. **Court Not Found**:  * At step 2, if the User/Guest selects a court that no longer exists or is unavailable:   + - The system displays an error message (e.g., "This court is not available. Please select another.")     - The user is redirected to the court listings to choose another court, returning to step 1.  1. **User Proceeds to Booking**:  * After step 6, if the User/Guest decides to book the court:   + - The system transitions the user to the booking process (a separate use case).     - The current use case ends.  1. **User Exits Without Action**:  * At any point, if the User/Guest chooses to exit the court details view:   + - The system returns the user to the court listings or homepage.     - The use case ends. | | |

#### 

#### 2.4.30 View Court

| Primary Actors | User, Guest, Owner, Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case allows **Users** and **Guests** to view available courts for booking, along with relevant details such as court type, location, available time slots, price, and court images. **Owners** and **Managers** can view all courts with additional management details, including maintenance status and booking statistics. | | |
| Preconditions | * The system has at least one court registered and marked as **active**. * The actor is either a guest, user, manager, or owner. * The court list or detail view is accessible from the home screen or management dashboard. | | |
| Postconditions | * The actor can view court information. * Court status (e.g., available, under maintenance) is shown. * The user may proceed to **booking**, while admin may go to **edit/delete**. | | |
| Normal Sequence/Flow | **For User/Guest:**   1. Actor accesses **Home Page** or **Court Listing**. 2. System displays available courts with:  * Court name and type (indoor/outdoor) * Location with map link * Price per slot/hour * Available time slots * Description and photos * “Book Now” or “View Details” button  1. Actor clicks on a specific court for more info. 2. System shows full details and booking calendar.   **For Owner/Manager:**   1. Actor logs into admin dashboard. 2. Navigates to **Court Management → Court List**. 3. System shows all courts, including:  * Booking stats * Status (active/inactive, under maintenance) * Edit/Delete buttons * Maintenance logs (if any) | | |
| Alternative Sequences/Flows | 1. **No Active Courts Found:**  * System displays: “No courts currently available.”  1. **Court Info Load Failure:**  * System shows an error message and retry option. | | |

#### 

#### 2.4.31 View Court Slot

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Users/Guests** to view available **court time slots** for booking. Time slots show detailed scheduling, availability, and pricing information. | | |
| Preconditions | * At least one court exists with assigned time slots. * **User/Guest** viewing available slots for booking | | |
| Postconditions | A list or calendar view of available (and optionally unavailable) court slots is displayed. | | |
| Normal Sequence/Flow | 1. Navigates to **Home → Select Court → View Schedule**. 2. System shows:  * Date picker or calendar view * List of available time slots (e.g., 06:00–07:00, 07:00–08:00) * Status (Available / Booked / Maintenance) * Price per slot  1. User may click on a slot to begin booking (if available). | | |
| Alternative Sequences/Flows | **No Slots Defined for Court:**   * System shows: “No time slots available for this court.”   **Slot Load Failure (System Error):**   * Error message is shown and retry option provided. | | |

#### 2.4.32 View Court Slot by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** can view the full list of time slots associated with each court, including inactive, booked, or under maintenance slots. Time slots show detailed scheduling, availability, and pricing information. | | |
| Preconditions | **Owner** accessing time slot configuration for a court | | |
| Postconditions | Actor may proceed to booking, editing, or analysis | | |
| Normal Sequence/Flow | 1. Logs into **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. System displays:  * Full list of all defined slots * Status: Active / Inactive / Fully booked * Price, recurrence pattern (e.g., every Monday) * Action buttons (Edit / Deactivate / Delete) | | |
| Alternative Sequences/Flows | **No Slots Defined for Court:**   * System shows: “No time slots available for this court.”   **Slot Load Failure (System Error):**   * Error message is shown and retry option provided. | | |

#### 2.4.33 View Court Slot by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** can view the full list of time slots associated with each court, including inactive, booked, or under maintenance slots. Time slots show detailed scheduling, availability, and pricing information under the Owner's authority | | |
| Preconditions | **Manager** accessing time slot configuration for a court | | |
| Postconditions | Actor may proceed to booking, editing, or analysis | | |
| Normal Sequence/Flow | 1. Logs into **admin dashboard**. 2. Navigates to **Sub**-**Court Management**. 3. System displays:  * Full list of all defined slots * Status: Active / Inactive / Fully booked * Price, recurrence pattern (e.g., every Monday) * Action buttons (Edit / Deactivate / Delete) | | |
| Alternative Sequences/Flows | **No Slots Defined for Court:**   * System shows: “No time slots available for this court.”   **Slot Load Failure (System Error):**   * Error message is shown and retry option provided. | | |

#### 

### *2.5 Service Management*

#### 2.5.1 Create Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to create new services that will be available for users to purchase. Services can include food and drinks, equipment rental, or coaching sessions. Each service includes a name, description, price, status, and optional images or categories. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The court exists in the system and is currently marked as **Active**. * The court is not tied to any pending or ongoing bookings. | | |
| Postconditions | * The court status is set to **Inactive**. * The court is hidden from the user booking interface. * The court remains accessible in admin view for editing or reactivation. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks on the **“Thêm dịch vụ mới”**. 4. Fills in:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”** 2. System displays:  “Tạo mới thành công .” | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Action:**  * No changes are made.  1. **System Error During Create:**  * Error is shown and action is rolled back. | | |

#### 2.5.2 Create Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to create new services that will be available for users to purchase. Services can include food and drinks, equipment rental, or coaching sessions. Each service includes a name, description, price, status, and optional images or categories under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * The court exists in the system and is currently marked as **Active**. * The court is not tied to any pending or ongoing bookings. | | |
| Postconditions | * The court status is set to **Inactive**. * The court is hidden from the user booking interface. * The court remains accessible in admin view for editing or reactivation. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks on the **“Thêm dịch vụ mới”**. 4. Fills in:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”** 2. System displays:  “Tạo mới thành công .” | | |
| Alternative Sequences/Flows | 1. **Actor Cancels Action:**  * No changes are made.  1. **System Error During Create:**  * Error is shown and action is rolled back. | | |

#### 2.5.3 Active Status Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **activate a service** (such as water, snack, paddle rental, etc.) so that it becomes available for users to view and purchase through the booking system or service order screens. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The service exists in the system and is currently in an **Inactive** status. * Service information (name, price, description) is fully configured. | | |
| Postconditions | * The selected service is set to **Active** status. * Users and guests can now view and purchase the service during booking or through service ordering. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Service Management → Service List**. 2. User filters or searches for **Inactive Services**. 3. User selects a service they wish to activate. 4. System displays service details for review. 5. Owner clicks the **Activate** button. 6. System updates the service status to **Active**. 7. Service becomes visible and purchasable for users immediately. 8. System logs the action for audit purposes. | | |
| Alternative Sequences/Flows | 1. **Service Validation Fails (Missing Critical Data):**  * System prevents activation and displays an error:  “Cannot activate service. Please complete all required information (e.g., pricing, description).”  1. **Database Error on Update:**  * System displays: “Activation failed. Please try again later.” | | |

#### 

#### 

#### 2.5.4 Active Status Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Manager** to **activate a service** (such as water, snack, paddle rental, etc.) so that it becomes available for users to view and purchase through the booking system or service order screens under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The service exists in the system and is currently in an **Inactive** status. * Service information (name, price, description) is fully configured. | | |
| Postconditions | * The selected service is set to **Active** status. * Users and guests can now view and purchase the service during booking or through service ordering. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Service Management → Service List**. 2. User filters or searches for **Inactive Services**. 3. User selects a service they wish to activate. 4. System displays service details for review. 5. Manager clicks the **Activate** button. 6. System updates the service status to **Active**. 7. Service becomes visible and purchasable for users immediately. 8. System logs the action for audit purposes. | | |
| Alternative Sequences/Flows | 1. **Service Validation Fails (Missing Critical Data):**  * System prevents activation and displays an error:  “Cannot activate service. Please complete all required information (e.g., pricing, description).”  1. **Database Error on Update:**  * System displays: “Activation failed. Please try again later.” | | |

#### 2.5.5 Deactive Status Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** to **deactivate a service** (such as bottled water, paddle rental, snacks, etc.) so that it is temporarily **unavailable** for users to view and purchase. Deactivation may be needed during out-of-stock situations, maintenance periods, or promotional adjustments. | | |
| Preconditions | * The user is authenticated with the role of **Owner** * The service currently exists in the system and is in an **Active** status. * No critical transactions are relying on the immediate availability of the service (e.g., pending service orders can still be fulfilled). | | |
| Postconditions | * The selected service is updated to **Inactive** status. * The service is hidden from users and cannot be selected for new bookings or service orders. | | |
| Normal Sequence/Flow | 1. Owner navigates to **Service Management → Service List**. 2. User filters or searches for **Active Services**. 3. User selects the service to deactivate. 4. System displays the service’s current details. 5. Owner clicks the **Deactivate** button. 6. System updates the service status to **Inactive**. 7. Service is removed from the user-facing booking and ordering interfaces. 8. System logs the deactivation action for auditing. | | |
| Alternative Sequences/Flows | 1. **Service in Use:**  * If the service is currently included in active unpaid orders, system warns the user:  “This service is included in ongoing transactions. Please confirm before deactivation.”  1. **Database Save Error:**  * System displays: “Unable to deactivate service. Please try again later.**”** | | |

#### 2.5.6 Deactive Status Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows an **Owner** or **Manager** to **deactivate a service** (such as bottled water, paddle rental, snacks, etc.) so that it is temporarily **unavailable** for users to view and purchase. Deactivation may be needed during out-of-stock situations, maintenance periods, or promotional adjustments under the Owner's authority | | |
| Preconditions | * The user is authenticated with the role of **Manager**. * The service currently exists in the system and is in an **Active** status. * No critical transactions are relying on the immediate availability of the service (e.g., pending service orders can still be fulfilled). | | |
| Postconditions | * The selected service is updated to **Inactive** status. * The service is hidden from users and cannot be selected for new bookings or service orders. | | |
| Normal Sequence/Flow | 1. Manager navigates to **Service Management → Service List**. 2. User filters or searches for **Active Services**. 3. User selects the service to deactivate. 4. System displays the service’s current details. 5. Manager clicks the **Deactivate** button. 6. System updates the service status to **Inactive**. 7. Service is removed from the user-facing booking and ordering interfaces. 8. System logs the deactivation action for auditing. | | |
| Alternative Sequences/Flows | 1. **Service in Use:**  * If the service is currently included in active unpaid orders, system warns the user:  “This service is included in ongoing transactions. Please confirm before deactivation.”  1. **Database Save Error:**  * System displays: “Unable to deactivate service. Please try again later.**”** | | |

#### 2.5.7 Update Service by Owner

#### 

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Manager** to edit details of an existing service (e.g., equipment rental, snacks, water, coaching). Updates can include changes to the name, price, description, availability status, category, or associated image. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * The service exists in the system. * The actor has permission to manage services. | | |
| Postconditions | * The selected service is updated with the new information. * Changes are saved to the database and visible to users if the service is active. | | |
| Normal Sequence/Flow | 1. Owner logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks **“Edit”** next to the desired service. 4. System displays a form pre-filled with current service details. 5. Actor modifies one or more fields:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”**. 2. System validates changes and updates the service record. 3. Confirmation message is shown:  “Cập nhật thành công” | | |
| Alternative Sequences/Flows | 1. **Invalid Input (e.g., empty name or invalid price):**  * System highlights error and requests correction.  1. **System Error on Save:**  * Displays error message and logs failure. | | |

#### 

#### 

#### 2.5.8 Update Service by Manager

#### 

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the **Manager** to edit details of an existing service (e.g., equipment rental, snacks, water, coaching). Updates can include changes to the name, price, description, availability status, category, or associated image under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * The service exists in the system. * The actor has permission to manage services. | | |
| Postconditions | * The selected service is updated with the new information. * Changes are saved to the database and visible to users if the service is active. | | |
| Normal Sequence/Flow | 1. Manager logs into the **admin dashboard**. 2. Navigates to **Service Management → Service List**. 3. Clicks **“Edit”** next to the desired service. 4. System displays a form pre-filled with current service details. 5. Actor modifies one or more fields:  * Danh mục * Tên dịch vụ * Giá * Số lượng * Đơn vị * Mô tả  1. Clicks **“Ok”**. 2. System validates changes and updates the service record. 3. Confirmation message is shown:  “Cập nhật thành công” | | |
| Alternative Sequences/Flows | 1. **Invalid Input (e.g., empty name or invalid price):**  * System highlights error and requests correction.  1. **System Error on Save:**  * Displays error message and logs failure. | | |

#### 2.5.9 View List of Service by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | **Owners** can view all services in the system, including inactive or archived ones, along with management data like status, price, and edit options. | | |
| Preconditions | * Services have been created and saved in the system. * All services (active/inactive) are accessible. | | |
| Postconditions | * A list of services is displayed. * Actors can view full service details. * Owner may **edit/delete/deactivate**. | | |
| Normal Sequence/Flow | Actor logs into the **admin dashboard**.   1. Navigates to **Service Management**. 2. System shows:  * Full service list (Active/Inactive) * Name, Price, Status, Edit/Delete buttons * Sorting, search, and filtering by category or status  1. Actor clicks on a service to see detailed view or modify it. | | |
| Alternative Sequences/Flows | * **No Services Available:** * System shows: “No services currently available.” * **Service Not Found or Deleted:** * System displays error message: “Service does not exist.” | | |

#### 

#### 2.5.10 View List of Service by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | **Manager** can view all services in the system, including inactive or archived ones, along with management data like status, price, and edit options under the Owner's authority | | |
| Preconditions | * Services have been created and saved in the system. * All services (active/inactive) are accessible. | | |
| Postconditions | * A list of services is displayed. * Actors can view full service details. * Manager may **edit/delete/deactivate**. | | |
| Normal Sequence/Flow | Actor logs into the **admin dashboard**.   1. Navigates to **Service Management**. 2. System shows:  * Full service list (Active/Inactive) * Name, Price, Status, Edit/Delete buttons * Sorting, search, and filtering by category or status  1. Actor clicks on a service to see detailed view or change it. | | |
| Alternative Sequences/Flows | * **No Services Available:** * System shows: “No services currently available.” * **Service Not Found or Deleted:** * System displays error message: “Service does not exist.” | | |

### *2.6 Staff Management*

#### 2.6.1 View List Order for Staff

#### 

| Primary Actors | Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Staff** to view a complete list of **court and service orders** created by users, including booking details, order status, payment status, and time. Staff can use this list to manage on-site service preparation, verify arrivals, and prepare resources for each time slot. | | |
| Preconditions | * Staff is authenticated and authorized to access the order module. * Orders have been created by users through the system. * The system’s order tracking module is active. | | |
| Postconditions | * A table of user orders (court bookings and services) is displayed. * Staff can filter, sort, and view detailed information. * Staff may optionally update order status (e.g., "Checked-in"). | | |
| Normal Sequence/Flow | 1. Staff logs into the **staff dashboard**. 2. Navigates to **Orders → View List**. 3. System displays:  * Court bookings (with time, court ID, user name, status) * Service orders (e.g., water, rentals, with quantity, status) * Columns:   + Order ID   + User name   + Order type (Court/Service)   + Date & time   + Status (Paid, Unpaid, Completed, Canceled)   + Action: View Details  1. Staff filters orders by:  * Date * Status * Type (Court / Service)  1. Staff clicks **“View”** to open full order detail. | | |
| Alternative Sequences/Flows | 1. **No Orders Found:**  * System shows: “No orders found for selected criteria.”  1. **Failed to Load Orders:**  * Error is shown: “Unable to load order list. Please try again.” | | |

#### 2.6.2 Perform Check-In

#### 

| Primary Actors | Staff | Secondary Actors |  |
| --- | --- | --- | --- |
| Description | This use case allows **Staff** to mark a booking or service order as **"Checked-in"** when the user arrives at the facility. This helps ensure proper scheduling, prepare the court, and confirm attendance for reporting and billing purposes. | | |
| Preconditions | * Staff is logged into the system with the appropriate role. * The booking/order has been created and is scheduled for the current or upcoming time. * Payment has been completed or deposit made (depending on policy). | | |
| Postconditions | * The booking/service order status is updated to **Checked-in**. * The check-in time is recorded in the system. * Optionally, a notification is sent to the user or management (if configured). | | |
| Normal Sequence/Flow | 1. Staff accesses the **order list** or **schedule view**. 2. Searches for or selects a **specific booking/service order**. 3. Confirms the user's identity (e.g., name, phone, or QR code). 4. Clicks **“Check-In”**. 5. System updates the order status to **Checked-in**. 6. A success message | | |
| Alternative Sequences/Flows | 1. **Order Not Found:**  * System shows: “Booking not found or not scheduled for today.”  1. **User Information Mismatch:**  * System blocks check-in and shows: “User identity could not be verified.”  1. **System Error During Update:**  * System displays: “Unable to perform check-in. Please try again later.” | | |

#### 

#### 2.6.3 Create Booking for Customer

#### 

| Primary Actors | Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables **Staff** to manually create a court booking on behalf of a **Customer**. This is useful when the customer books directly at the facility, via phone call, or doesn’t have access to the website or app. Staff can select courts, time slots, and input basic customer details. | | |
| Preconditions | * Staff is authenticated and has access to booking functions. * At least one court and available time slot exist. * The booking date/time is valid (in the future, within booking window). | | |
| Postconditions | * A booking is created in the system and linked to either:    + A registered user (if known)   + Or as a **guest booking** with phone number or name * The system optionally generates a payment QR or confirms deposit. | | |
| Normal Sequence/Flow | 1. Staff logs into the **staff dashboard**. 2. Navigates to **Bookings → Create Booking for Customer**. 3. Inputs customer details:  * Name * Phone number or email (optional)  1. Selects:  * Court * Date * Time slot * Booking type (One-time, Fixed, Optimized)  1. Confirms pricing and adds notes if necessary. 2. Clicks **“Confirm Booking”**. 3. System creates the booking and shows:  * Booking summary * Optional: Generate payment QR or mark as deposit received  1. Staff confirms with the customer and (optionally) sends email summary. | | |
| Alternative Sequences/Flows | 1. **Payment Pending:**  * System creates booking with **Pending Payment** status and shows QR.  1. **Customer Cancels Midway:**  * Staff discards form; no booking created. | | |

#### 2.6.4 View Staff Dashboard

#### 

| Primary Actors | Staff | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Staff** to access the centralized dashboard after logging in. The dashboard provides a summary of daily bookings, service orders, check-in status, upcoming schedules, and quick actions such as creating bookings or processing payments. | | |
| Preconditions | * The actor is authenticated as **Staff**. * The staff dashboard module is enabled. * Booking, service, and report data are available in the system. | | |
| Postconditions | * The staff member sees an overview of relevant operational data. * From the dashboard, the staff can navigate to booking, order, and check-in functions. | | |
| Normal Sequence/Flow | 1. Staff logs into the system using valid credentials. 2. System redirects to the **Staff Dashboard**. 3. Dashboard displays:  * **Date Bookings** (list of court reservations with status) * Quick buttons:   + “Create Booking for Customer”   + “Sell Service”   + “View Calendar”  1. Staff clicks on a module (e.g., “Today’s Bookings”) to view details or take action. | | |
| Alternative Sequences/Flows | 1. **Dashboard Load Fails:**  * System displays: “Trang bạn đang tìm kiếm có thể đã bị xóa, tên trang đã được đổi hoặc tạm thời không khả dụng.”  1. **No Bookings/Orders Today:**  * System shows: “No data.” | | |

### *2.7 Payment System*

#### 2.7.1 Generate QR for Payment

#### 

| Primary Actors | User, Guest | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows **Users** or **Guests** to generate a QR code linked to their booking or service order. The QR code contains transaction details and can be scanned using supported banking or e-wallet apps to complete payment or deposit. | | |
| Preconditions | * A booking or service order exists in **pending payment** or **deposit required** status. * The payment gateway integration is active and functioning. * The user is on a device that supports QR generation/display (web or mobile). | | |
| Postconditions | * A QR code is generated and displayed to the user. * The code links to the correct payment amount and transaction ID. * Upon scanning and payment confirmation, the order is updated to **Paid** or **Deposit Received**. | | |
| Normal Sequence/Flow | 1. User completes a court booking or service order. 2. On the **payment page**, clicks **“Pay with QR Code”**. 3. System generates a secure QR code containing:  * Order ID * Amount due * Payment reference/token * Payment provider URL or payload  1. QR code is displayed on screen. 2. User scans the QR code with a bank or e-wallet app. 3. Upon successful payment, the payment gateway confirms the transaction. 4. System updates the order status to **Paid**. 5. Success message is displayed:  “Payment successful. Your booking is confirmed!” | | |
| Alternative Sequences/Flows | 1. **QR Generation Failure:**  * System shows error: “Unable to generate QR code. Please try another payment method.”  1. **Payment Not Confirmed in Time:**  * System cancels pending payment after timeout and shows message:  “Payment not received. Please try again or choose another method.”  1. **User Cancels or Closes Window:**  * No status change; order remains in **Pending Payment**. | | |

#### 

#### 2.7.2 Online Payment Processing

| Primary Actors | User | Secondary Actors | Payment Gateway |
| --- | --- | --- | --- |
| Description | This use case covers the process of securely handling online payments for court bookings or service orders via QR code or other integrated payment gateways. After an order is created (court or service), the system generates a payment request and processes confirmation upon successful transaction. | | |
| Preconditions | * The user has selected to pay for a court or service order. * A valid order with total amount exists in the system. * The payment gateway (e.g., VNPay, Momo, etc.) is connected and operational. | | |
| Postconditions | * The system marks the order as **Paid**. * A transaction record is created and linked to the order. * The booking/service is confirmed, and user receives a receipt. | | |
| Normal Sequence/Flow | 1. User confirms their booking or service order. 2. System generates a payment request with total amount. 3. System creates and displays a **QR code** or redirects to the payment gateway. 4. User scans QR or completes payment via banking app. 5. Payment gateway notifies the system of success. 6. System updates:  * Order status → **Paid** * Booking status → **Confirmed** * Transaction record with:   + Amount   + Time   + Payment method   + Order ID  1. Confirmation screen is shown. 2. System sends email/notification receipt (optional). | | |
| Alternative Sequences/Flows | 1. **3A. QR Code Fails to Load:**  * System displays error and retries or allows manual link.  1. **5A. Payment Fails or Timeout:**  * System shows “Payment failed. Please try again.” * Order remains in “Pending” status.  1. **6A. Delay in Payment Notification:**  * System periodically polls payment gateway for status.  1. **7A. User Closes Payment Window Early:**  * System allows retry or shows pending status in order history. | | |

#### 

#### 

### *2.8 Notification Management*

#### 2.8.1 Receive Notifications

| Primary Actors | User, Guest, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows all actors (**User, Guest, Staff, Manager, Owner**) to **receive notifications** triggered by system activities such as bookings, payments, refunds, schedule changes, maintenance updates, or promotional announcements. Notifications help users stay informed about important events related to their activities in the system. | | |
| Preconditions | * The actor is authenticated in the system or has an active session (Guest). * Notification generation service is active and configured. * Relevant trigger events occur (e.g., successful booking, refund processed, maintenance scheduled). | | |
| Postconditions | * The actor receives the corresponding notification. * Notifications are stored in the system for future reference. * Users can view unread and read notifications separately. | | |
| Normal Sequence/Flow | 1. A trigger event happens in the system (e.g., booking created, payment confirmed). 2. System generates a notification based on event type and actor role. 3. System delivers the notification to:     * + - In-app Notification Center        - Optionally, Email or SMS if configured 4. User/Actor logs in or is already logged in. 5. User/Actor views notification list, sees a badge or alert icon. 6. User/Actor clicks to open and read full notification details. 7. Notification is marked as read once opened. | | |
| Alternative Sequences/Flows | 1. **Notification Delivery Failure (External):**  * If email or SMS fails, the notification remains available in-app only. * System logs the delivery failure for admin review.  1. **User Disables Some Notification Types:**  * Only selected types are pushed to the user based on their preferences.  1. **Trigger Event Error:**  * If trigger event fails (e.g., failed payment), system skips sending success notifications. | | |

### *2.9 Financial Management*

#### 2.9.1 View All Transaction History by Owner

| Primary Actors | Owner | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case allows **Owners** to access and review the full list of all payment transactions that occurred in the system, including those related to court bookings, service purchases, deposits, and refunds. Transactions include payment method, status, amount, and are filterable for financial reporting and auditing. | | |
| Preconditions | * The actor is authenticated with an **Manager** role. * Transactions have been recorded in the system. * The transaction log module is enabled and accessible from the dashboard. | | |
| Postconditions | * A list of all system-wide financial transactions is displayed. * The actor can filter, sort, and click to view specific transaction details. * Optional: Export the transaction list to Excel/PDF. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin panel. 2. Navigates to **Transaction**. 3. System loads a list of all transactions, each showing:  * Transaction ID * Bill Code * Order ID * Payment status * Amount * Timestamp  1. Actor filters by:  * Date range * Payment status * Order ID  1. Optionally, clicks **“Export”** to download the list. | | |
| Alternative Sequences/Flows | 1. **No Transactions Found:**  * System shows: “No data.”  1. **Filter Yields No Results:**  * System shows: “No data.” | | |

#### 

#### 2.9.2 View All Transaction History by Manager

| Primary Actors | Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case allows **Managers** to access and review the full list of all payment transactions that occurred in the system, including those related to court bookings, service purchases, deposits, and refunds. Transactions include payment method, status, amount, and are filterable for financial reporting and auditing under the Owner's authority | | |
| Preconditions | * The actor is authenticated with an **Manager** role. * Transactions have been recorded in the system. * The transaction log module is enabled and accessible from the dashboard. | | |
| Postconditions | * A list of all system-wide financial transactions is displayed. * The actor can filter, sort, and click to view specific transaction details. * Optional: Export the transaction list to Excel/PDF. | | |
| Normal Sequence/Flow | 1. Manager logs into the admin panel. 2. Navigates to **Transaction**. 3. System loads a list of all transactions, each showing:  * Transaction ID * Bill Code * Order ID * Payment status * Amount * Timestamp  1. Actor filters by:  * Date range * Payment status * Order ID  1. Optionally, clicks **“Export”** to download the list. | | |
| Alternative Sequences/Flows | 1. **No Transactions Found:**  * System shows: “No data.”  1. **Filter Yields No Results:**  * System shows: “No data.” | | |

#### 

#### 2.9.3 View Revenue Report by Owner

| Primary Actors | Owner | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Owners** to access a visual and/or tabular summary of **total revenue** generated from court bookings, service purchases, and other transactions within a selected time frame. The revenue report helps monitor business performance, identify trends, and support financial decision-making. | | |
| Preconditions | * The actor is authenticated with **Owner** role. * Revenue data exists for the selected time range. * The reporting/analytics module is active. | | |
| Postconditions | * A visual revenue summary (charts, graphs, totals) is displayed. * Data is filterable by date, type (court/service), and source (QR/cash/etc.). * Optional export/download (Excel/PDF) may be available. | | |
| Normal Sequence/Flow | 1. Owner logs into the admin dashboard. 2. Navigates to **Statistic**. 3. Selects a time range. 4. System retrieves revenue data from bookings. 5. System displays:  * Total revenue * Revenue breakdown by type (court, deposit,..) * Bar Chart, line Chart.  1. Actor may filter by:  * By court, by day, by month * Date range * Payment status * Order status,..  1. Optionally, clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Available:**  * System displays: “No data ”  1. **Analytics Module Error:**  * System shows fallback text view or error message and retry option.  1. **Export Fails:**  * System display:” Xuất file không thành công”. | | |

#### 2.9.4 View Revenue Report by Manager

| Primary Actors | Manager | Secondary Actors | Payment System |
| --- | --- | --- | --- |
| Description | This use case enables **Managers** to access a visual and/or tabular summary of **total revenue** generated from court bookings, service purchases, and other transactions within a selected time frame. The revenue report helps monitor business performance, identify trends, and support financial decision-making under the Owner's authority | | |
| Preconditions | * The actor is authenticated with **Manager** role. * Revenue data exists for the selected time range. * The reporting/analytics module is active. | | |
| Postconditions | * A visual revenue summary (charts, graphs, totals) is displayed. * Data is filterable by date, type (court/service), and source (QR/cash/etc.). * Optional export/download (Excel/PDF) may be available. | | |
| Normal Sequence/Flow | 1. Manager logs into the admin dashboard. 2. Navigates to **Statistic**. 3. Selects a time range. 4. System retrieves revenue data from bookings. 5. System displays:  * Total revenue * Revenue breakdown by type (court, deposit,..) * Bar Chart, line Chart.  1. Actor may filter by:  * By court, by day, by month * Date range * Payment status * Order status,..  1. Optionally, clicks **“Export”** to download the report. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Available:**  * System displays: “No data ”  1. **Analytics Module Error:**  * System shows fallback text view or error message and retry option.  1. **Export Fails:**  * System display:” Xuất file không thành công”. | | |

#### 2.9.5 Export Revenue Report by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to export a detailed revenue report summarizing income from court bookings, service purchases, and other transactions within a specified time range. The report is exported as an Excel (.xlsx) or PDF file for analysis, recordkeeping, or financial auditing. | | |
| Preconditions | * The actor is logged in with **Owner** role. * Revenue data exists in the system for the selected date range. * The system has access to the reporting module and database. | | |
| Postconditions | * A structured financial report file (Excel) is generated. * The file includes totals, breakdowns, and optional transaction logs. * The action may be logged for security/audit tracking. | | |
| Normal Sequence/Flow | 1. Owner navigates to the **Reports** or **Revenue Dashboard**. 2. Selects:  * Date range * Payment status * Order status * Court * By Court, by month, by day * Order Type  1. Selects **“Xuất Báo Cáo”**. 2. System retrieves all relevant financial data. 3. System generates a report including:  * Total Revenue * Total Deposit * Total Refund * Total Paid,...  1. System prepares the file and download. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Found:**  * System displays all fields as 0.  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công” | | |

#### 

#### 2.9.6 Export Revenue Report by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Managerto export a detailed revenue report summarizing income from court bookings, service purchases, and other transactions within a specified time range. The report is exported as an Excel (.xlsx) or PDF file for analysis, recordkeeping, or financial auditing. | | |
| Preconditions | * The actor is logged in with **Manager** role. * Revenue data exists in the system for the selected date range. * The system has access to the reporting module and database. | | |
| Postconditions | * A structured financial report file (Excel or PDF) is generated. * The file includes totals, breakdowns, and optional transaction logs. * The action may be logged for security/audit tracking. | | |
| Normal Sequence/Flow | 1. Manager navigates to the **Reports** or **Revenue Dashboard**. 2. Selects:  * Date range * Payment status * Order status * Court * By Court, by month, by day * Order Type  1. Selects **“Xuất Báo Cáo”**. 2. System retrieves all relevant financial data. 3. System generates a report including:  * Total Revenue * Total Deposit * Total Refund * Total Paid,...  1. System prepares the file and download. | | |
| Alternative Sequences/Flows | 1. **No Revenue Data Found:**  * System displays all fields as 0.  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công” | | |

#### 2.9.7 Export Booking Orders by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables the Owner to export a report of all purchase orders made by users, including court bookings and service orders, within a specific time range. The exported data supports operational tracking, accounting, and reconciliation processes, and is generated in Excel (.xlsx) or PDF format. | | |
| Preconditions | * The actor is authenticated with appropriate permissions (Owner). * The system contains at least one order within the selected timeframe. * The reporting module is active and integrated with the database. | | |
| Postconditions | * A structured export file is generated with order information. * The file is successfully downloaded by the actor. * The export may be logged for administrative purposes. | | |
| Normal Sequence/Flow | 1. Owner logs in and navigates to the **Booking Management** or **Reports** section. 2. Selects:  * Date range * Payment status * Order status * Court name  1. Clicks on **“Xuất Excel”**. 2. System retrieves matching orders from the database. 3. System generates a file containing:  * Order ID * Court ID * Custome name * Phone Number * Order Type * Order Status * Payment Status * Total Amount * Deposit Amount * Disscount Amount,...  1. System prepares the file for download. 2. Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | 1. **No Orders Found:**  * System displays: “No data.”  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công.” | | |

#### 

#### 2.9.8 Export Booking Orders by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case enables the Manager to export a report of all purchase orders made by users, including court bookings and service orders, within a specific time range. The exported data supports operational tracking, accounting, and reconciliation processes, and is generated in Excel (.xlsx) or PDF format under the Owner's authority | | |
| Preconditions | * The actor is authenticated with appropriate permissions ( Manager). * The system contains at least one order within the selected timeframe. * The reporting module is active and integrated with the database. | | |
| Postconditions | * A structured export file is generated with order information. * The file is successfully downloaded by the actor. * The export may be logged for administrative purposes. | | |
| Normal Sequence/Flow | 1. Manager logs in and navigates to the **Booking Management** or **Reports** section. 2. Selects:  * Date range * Payment status * Order status * Court name  1. Clicks on **“Xuất Excel”**. 2. System retrieves matching orders from the database. 3. System generates a file containing:  * Order ID * Court ID * Custome name * Phone Number * Order Type * Order Status * Payment Status * Total Amount * Deposit Amount * Disscount Amount,...  1. System prepares the file for download. 2. Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | 1. **No Orders Found:**  * System displays: “No data.”  1. **Download Error or Permission Issue:**  * System displays: “Xuất file không thành công.” | | |

#### 

#### 

#### 2.9.9 Export Transaction by Owner

| Primary Actors | Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Owner to export a list of financial transactions recorded in the system, including payments for court bookings, services, and other activities. The transaction report can be filtered by date, payment status, or payment method, and is exported as an Excel (.xlsx) for accounting and auditing purposes. | | |
| Preconditions | * The actor is authenticated with the **Owner** role. * The system contains transaction records for the time frame, or by the selected court. * The reporting module is active and has access to transaction logs. | | |
| Postconditions | * A transaction report is generated and ready for download. * The file includes accurate transaction data based on the selected filters. | | |
| Normal Sequence/Flow | Owner login to the admin dashboard.  Navigates to the **Transaction** section.  Selects:   * Date range * Payment status * By day * Order status * Order type * Court Name   Click on **“Xuất báo cáo”**.  System queries the transaction database.  System generates a file including:   * Period * Court ID * Court Name * Total Revenue * Deposit Amount * Paid Amount * Refund Amount   System prepares the file for download.  Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | * **No Transactions Found:** * System shows: “No data.” * **Download Failed:** * System shows error: “Xuất file không thành công”. | | |

## 

#### 2.9.10 Export Transaction by Manager

| Primary Actors | Manager | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | This use case allows the Manager to export a list of financial transactions recorded in the system, including payments for court bookings, services, and other activities. The transaction report can be filtered by date, payment status, or payment method, and is exported as an Excel (.xlsx) for accounting and auditing purposes under the Owner's authority | | |
| Preconditions | * The actor is authenticated with the **Manager** role. * The system contains transaction records for the time frame, or by the selected court. * The reporting module is active and has access to transaction logs. | | |
| Postconditions | * A transaction report is generated and ready for download. * The file includes accurate transaction data based on the selected filters. | | |
| Normal Sequence/Flow | Manager login to the admin dashboard.  Navigates to the **Transaction** section.  Selects:   * Date range * Payment status * By day * Order status * Order type * Court Name   Click on **“Xuất báo cáo”**.  System queries the transaction database.  System generates a file including:   * Period * Court ID * Court Name * Total Revenue * Deposit Amount * Paid Amount * Refund Amount   System prepares the file for download.  Actor downloads the file successfully. | | |
| Alternative Sequences/Flows | * **No Transactions Found:** * System shows: “No data.” * **Download Failed:** * System shows error: “Xuất file không thành công”. | | |

#### 2.9.11 Forgot Password

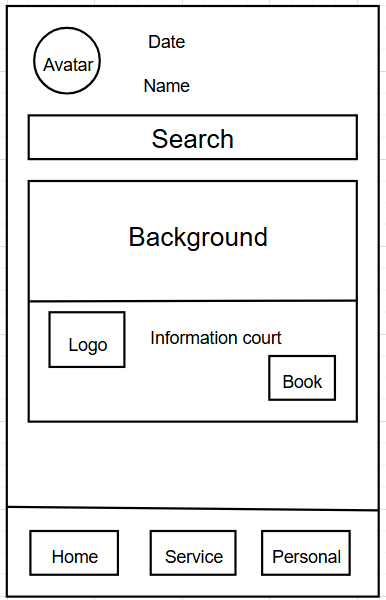
| Primary Actors | User, Staff, Manager, Owner | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | When users forget the password and want to get the password back to log in | | |
| Preconditions | * Users must have an associated account with valid email | | |
| Postconditions | * Users get the password back via email | | |
| Normal Sequence/Flow | 1. Users on Home page 2. Users click on "Forget Password" 3. User username/email/phonenumber to get a password via email 4. Users press the "Reset Password" button | | |
| Alternative Sequences/Flows | **N/A** | | |

## 

## 3. Functional Requirements

### 3.3 System Owneristration

#### 3.3.1 Home Screen



This screen allows the User to:

* View Home Screen: view the main interface of the application.
* Search Content: use the search bar to find court names.
* Navigate Sections: access different sections of the application (Home, Service, Personal) via the navigation bar.

On the screen, s/he can also:

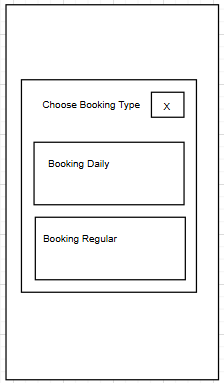
* View User Information: see the user’s avatar, name, and date.
* Access Information Court: interact with the "Information Court" section, including a "Book" option for additional details.
* Display Branding: view the application logo.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Name | Data type: non-digit string, max length of 50 characters. Displays user’s name. |
| Date | Displays the current date in a readable format (e.g., MM/DD/YYYY). |
| Search | Data type: any string, max length of 100 characters. Allows keyword search. |
| Home | Navigation button to access the Home section (current screen). |
| Service | Navigation button to access the Service section |
| Personal | Navigation button to access the Personal section |

#### 

#### 3.3.2 Booking Method Dialog



This screen allows the User to:

• View Booking Options: view the available booking types on the main interface.

• Select Booking Type: choose between different booking options (Booking Daily or Booking Regular).

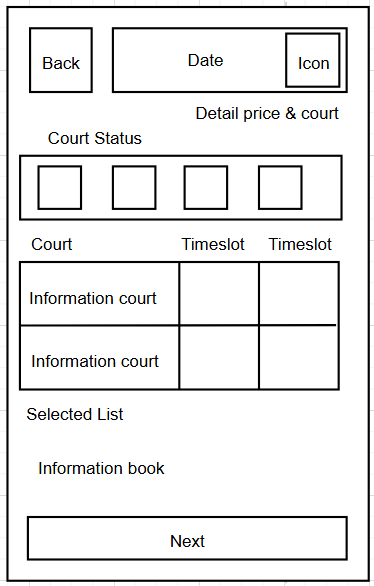
On the screen, s/he can also:

• Close Selection: close the booking selection screen by clicking the "X" button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| X | Navigation button to close the booking selection screen. |
| Booking Daily | Button to select the daily booking option. |
| Booking Regular | Button to select the regular booking option. |
| Choose Booking Type | Data type: non-digit string, max length of 50 characters. Displays the title of the booking selection screen |

#### 3.3.3 Booking Daily Screen



This screen allows the User to:

• View Court Booking Details: view the court status, timeslots, and pricing on the main interface.

• Select Court and Timeslot: choose a court and specific time slot for booking.

On the screen, s/he can also:

• Navigate Back: return to the previous screen by clicking the "Back" button.

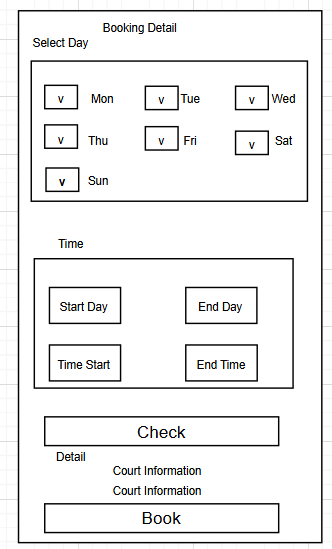
• Proceed to Next Step: move to the next step of the booking process by clicking the "Next" button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Date | Displays the current date in a readable format (e.g., MM/DD/YYYY). |
| Court Status | Data type: non-digit string, max length of 50 characters. Displays the availability status of courts. |
| Detail Price & Court | Data type: non-digit string, max length of 100 characters. Displays pricing and court details. |
| Court | Navigation button to select a specific court for booking. |
| Timeslot | Navigation button to select a specific time slot for booking. |
| Information Court | Data type: any string, max length of 200 characters. Displays detailed information about the court. |
| Information Book | Data type: any string, max length of 200 characters. Displays selected booking details. |
| Back | Navigation button to return to the previous screen. |
| Next | Navigation button to proceed to the next step of booking. |

#### 

#### 3.3.4 Regular booking scheduling Screen

******

This screen allows the User to:

• View Booking Details: view the interface for selecting booking days and times.

• Select Days: choose specific days (Mon, Tue, Wed, Thu, Fri, Sat, Sun) for booking.

• Set Booking Time: specify the start day, end day, start time, and end time for the booking.

On the screen, s/he can also:

• Check Availability: click the "Check" button to verify court availability for the selected days and times.

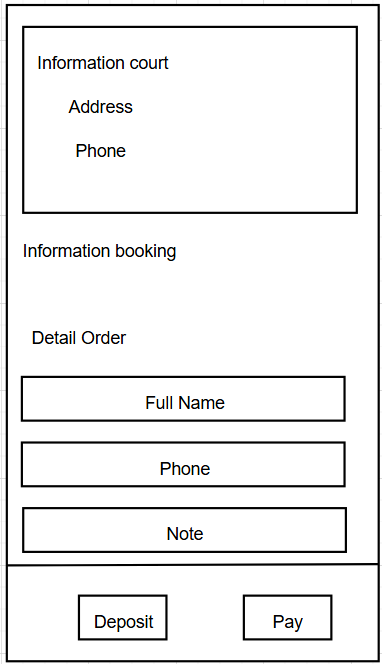
• Confirm Booking: finalize the booking by clicking the "Book" button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Select Day | Allows the user to select days of the week for booking using checkboxes labeled "Mon," "Tue," "Wed," "Thu," "Fri," "Sat," and "Sun." |
| Start Day | Data type: date, format (e.g., MM/DD/YYYY). Specifies the booking start date. |
| End Day | Data type: date, format (e.g., MM/DD/YYYY). Specifies the booking end date. |
| Time Start | Data type: time, format (e.g., HH:MM). Specifies the booking start time. |
| End Time | Data type: time, format (e.g., HH:MM). Specifies the booking end time. |
| Check | Navigation button to check court availability for the selected schedule. |
| Court Information | Data type: any string, max length of 200 characters. Displays detailed information about the court for user viewing. |
| Book | Navigation button to confirm and finalize the booking. |

#### 

#### 3.3.5 Confirm Order Screen



This screen allows the User to:

• View Order Summary: view the court and booking information on the main interface.

• Enter Order Details: input personal details such as full name, phone number, and notes for the booking.

On the screen, s/he can also:

• Proceed with Payment: choose to either pay a deposit by clicking the "Deposit" button or pay the full amount by clicking the "Pay" button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Information Court | Data type: any string, max length of 200 characters. Displays detailed information about the court for user viewing. |
| Address | Data type: any string, max length of 100 characters. Displays the court’s address for user viewing. |
| Phone (Court) | Data type: any string, max length of 20 characters. Displays the court’s phone number for user viewing. |
| Information Booking | Data type: any string, max length of 200 characters. Displays booking details for user viewing. |
| Full Name | Data type: non-digit string, max length of 50 characters. Specifies the user’s full name (required). |
| Phone (User) | Data type: non-digit string, max length of 50 characters. Specifies the user’s full name (required). |
| Note | Data type: any string, max length of 200 characters. Allows the user to add additional notes for the booking. |
| Deposit | Navigation button to proceed with paying a deposit for the booking. |
| Pay | Navigation button to proceed with paying the full amount for the booking |

#### 3.3.6 QR Code screen

This screen allows the User to:

• View QR Code: view the QR code for scanning on the main interface.

• Scan QR Code: use the displayed QR code to complete a booking or verification process.

On the screen, s/he can also:

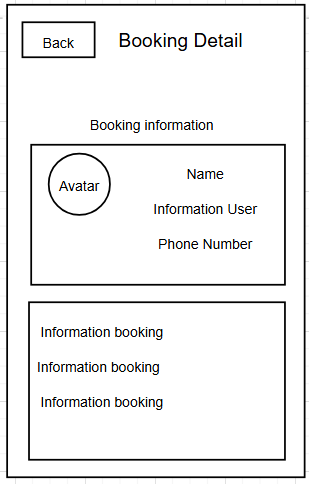
• Navigate Back: return to the previous screen by clicking the "Back" button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Scan QR Code | Data type: non-digit string, max length of 50 characters. Displays the title of the QR code scanning section. |
| Warning | Data type: any string, max length of 200 characters. Displays warning or instructional message for user viewing. |
| QR | Displays the QR code for scanning (non-interactive). |
| Time Remaining | Data type: any string, max length of 50 characters. Displays the title of the time remaining section for user viewing. |
| Time | Data type: time, format (e.g., MM:SS). Displays the remaining time for the QR code validity for user viewing. |
| Back | Navigation button to return to the previous screen. |

#### 

#### 3.3.7 Booking details screen



This screen allows the User to:

• View Booking Details: view the details of a specific booking.

• Navigate Back: return to the previous screen using the Back button.

On the screen, s/he can also:

• View User Information: see the user’s name and phone number in the Booking Information section.

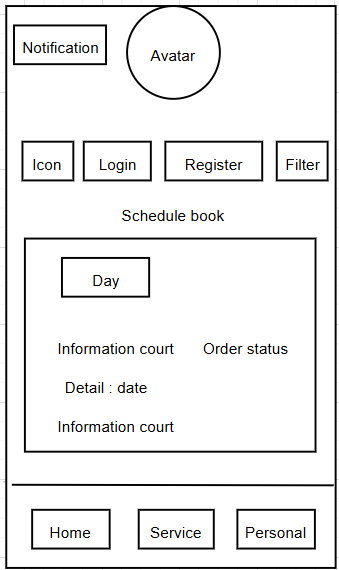
• View Booking Information: see details related to the booking in the Information Booking section.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Back | Navigation button to return to the previous screen. |
| Name | Data type: non-digit string, max length of 50 characters. Displays user’s name (non-interactive). |
| Phone Number | Data type: string, max length of 20 characters. Displays user’s phone number (non-interactive). |
| Information Booking | Displays booking-related details (non-interactive). |

#### 

#### 3.3.8 List order of Guest Screen



This screen allows the Guest to:

• View Order List: view a list of orders or bookings in the Schedule Book section.

• Filter Orders: use the Filter button to refine the list of orders.

• Navigate Sections: access different sections of the application (Home, Service, Personal) via the navigation bar.

On the screen, s/he can also:

• View Notifications: see notifications via the Notification button.

• Access Authentication: use the Login or Register buttons to sign in or create an account.

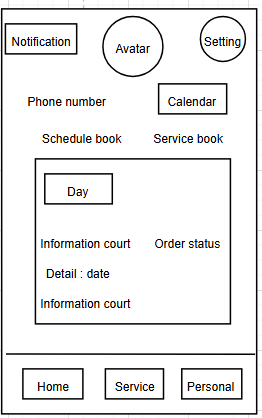
• View Court Information: see details about the court, including order status and date, in the Schedule Book section.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Notification | Displays notifications for the user (interactive button). |
| Login | Button to access the login functionality for existing users. |
| Register | Button to access the registration functionality for new users. |
| Filter | Button to filter the list of orders based on user-defined criteria. |
| Day | Displays the selected day for the schedule (non-interactive). |
| Information Court | Displays court-related details (non-interactive). |
| Order Status | Displays the status of the order (non-interactive). |
| Detail | Displays the date of the booking or order (non-interactive). |
| Home | Navigation button to access the Home section. |
| Service | Navigation button to access the Service section. |
| Personal | Navigation button to access the Personal section. |

#### 

#### 3.3.9 User Information Screen



This screen allows the User to:

• View User Information: view details such as phone number and booking information.

• Access Calendar: interact with the Calendar to select dates for bookings.

• Navigate Sections: access different sections of the application (Home, Service, Personal) via the navigation bar.

On the screen, s/he can also:

• View Notifications: see notifications via the Notification button.

• Access Settings: use the Setting button to adjust application settings.

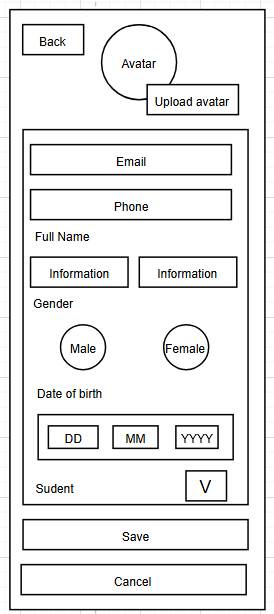
• View Court Information: see details about the court, including order status and date, in the Schedule Book section.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Notification | Interactive button to display notifications for the user. |
| Setting | Interactive button to access application settings. |
| Phone Number | Data type: string, max length of 20 characters (e.g., "123-456-7890"). Displays user’s phone number (non-interactive). |
| Calendar | Interactive element to select dates for bookings (format: MM/DD/YYYY). |
| Day | Data type: string, max length of 10 characters (e.g., "Monday"). Displays the selected day for the schedule (non-interactive). |
| Information Court | Data type: string, max length of 200 characters. Displays court-related details (non-interactive). |
| Order Status | Data type: string, max length of 20 characters (e.g., "Confirmed"). Displays the status of the order (non-interactive). |
| Detail | Data type: string, max length of 10 characters (format: MM/DD/YYYY). Displays the date of the booking or order (non-interactive). |
| Home | Interactive navigation button to access the Home section. |
| Service | Interactive navigation button to access the Service section. |
| Personal | Interactive navigation button to access the Personal section. |

#### 

#### 3.3.10 Edit Profile Screen



This screen allows the User to:

• Edit User Information: update personal details such as email, phone, full name, gender, date of birth, and student status.

• Navigate Back: return to the previous screen using the Back button.

On the screen, s/he can also:

• Upload Avatar: upload a new profile image via the Upload Avatar button.

• Save Changes: save the updated information using the Save button.

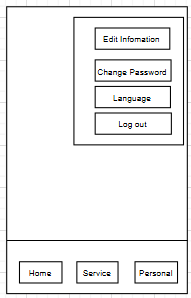
• Cancel Changes: discard changes and return to the previous screen using the Cancel button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Back | Interactive navigation button to return to the previous screen. |
| Upload Avatar | Interactive button to upload a new profile image. |
| Email | Data type: string, max length of 100 characters (e.g., "user@gmail.com"). Editable field for user’s email. |
| Phone | Data type: string, max length of 20 characters (e.g., "123-456-7890"). Editable field for user’s phone number. |
| Full Name | Data type: string, max length of 50 characters. Editable field for user’s full name. |
| Gender (Male/Female) | Options: Male, Female (radio buttons for selection). Editable field for user’s gender. |
| Date of Birth | Data type: string, format: DD/MM/YYYY (e.g., "25/04/2000"). Editable field with separate inputs for day (DD), month (MM), and year (YYYY). |
| Student | Data type: boolean (checkbox, checked for Yes, unchecked for No). Editable field to indicate if the user is a student. |
| Save | Interactive button to save the updated user information. |
| Cancel | Interactive button to discard changes and return to the previous screen. |

#### 

#### 3.3.11 Dialog Edit Profile Screen

****

This screen allows the User to:

• Access Profile Options: interact with options to edit profile information, change password, select language, or log out.

• Navigate Sections: access different sections of the application (Home, Service, Personal) via the navigation bar.

On the screen, s/he can also:

• Edit Information: navigate to the Edit User Information screen by selecting the Edit Information option.

• Change Password: navigate to the Change Password screen by selecting the Change Password option.

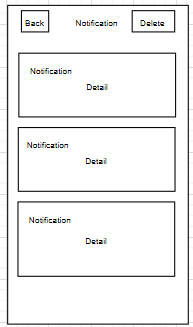
• Select Language: navigate to the Language selection screen by selecting the Language option.

• Log Out: log out of the application by selecting the Logout option.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Edit Information | Interactive button to navigate to the Edit User Information screen. |
| Change Password | Interactive button to navigate to the Change Password screen. |
| Language | Interactive button to navigate to the Language selection screen. |
| Log Out | Interactive button to log out of the application. |
| Home | Interactive navigation button to access the Home section. |
| Service | Interactive navigation button to access the Service section. |
| Personal | Interactive navigation button to access the Personal section. |

#### 3.3.12 Notification Screen



This screen allows the User to:

• View Notifications: view a list of notifications with their details.

• Navigate Back: return to the previous screen using the Back button.

On the screen, s/he can also:

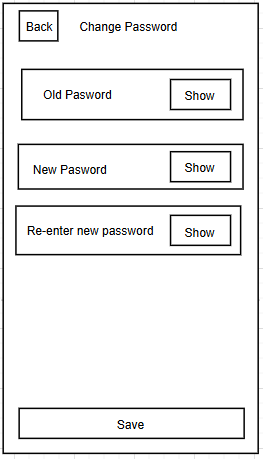
• Delete Notifications: remove notifications using the Delete button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Back | Interactive navigation button to return to the previous screen. |
| Notification | Interactive button to delete notifications. |
| Delete | Data type: string, max length of 100 characters. Displays the notification title (non-interactive). |
| Detail | Data type: string, max length of 200 characters. Displays the details of the notification (non-interactive). |

#### 

#### 3.3.13 Change Password Screen



This screen allows the User to:

• Change Password: update the user’s password by entering the old and new passwords.

• Navigate Back: return to the previous screen using the Back button.

On the screen, s/he can also:

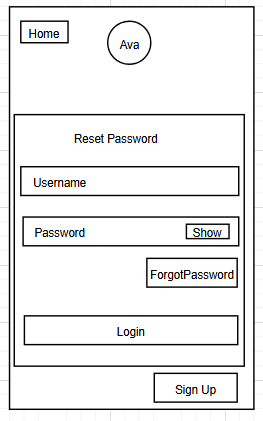
• Save Changes: save the updated password using the Save button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Back | Interactive navigation button to return to the previous screen. |
| Old Password | Data type: string, max length of 50 characters. Editable field for the current password. |
| New Password | Data type: string, max length of 50 characters. Editable field for the new password. |
| Re-enter New Password | Data type: string, max length of 50 characters. Editable field to confirm the new password. |
| Show | Interactive button to toggle visibility of the password fields (Show/Hide). |
| Save | Interactive button to save the updated password. |

#### 

#### 3.3.14 Login Screen



This screen allows the User to:

• Log In: access the application by entering a username and password.

• Navigate Back: return to the Home screen using the Home button.

On the screen, s/he can also:

• Reset Password: navigate to the Reset Password screen using the Reset Password link.

• Access Forgot Password: navigate to the Forgot Password screen using the Forgot Password link.

• Sign Up: navigate to the Sign Up screen using the Sign Up button.

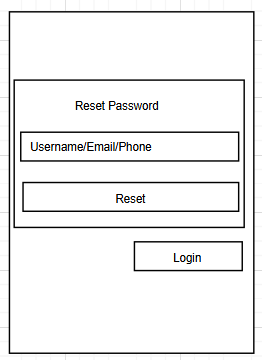
• Toggle Password Visibility: show or hide the password input field using the Show button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Home | Interactive navigation button to return to the Home screen. |
| Reset Password | Interactive link to navigate to the Reset Password screen. |
| Username | Data type: string, max length of 50 characters. Editable field for the user’s username. |
| Password | Data type: string, max length of 50 characters. Editable field for the user’s password. |
| Show | Interactive button to toggle visibility of the password field (Show/Hide). |
| Forgot Password | Interactive link to navigate to the Forgot Password screen. |
| Login | Interactive button to log in to the application. |
| Sign Up | Interactive button to navigate to the Sign Up screen. |

#### 

#### 3.3.15 Reset Password

****

This screen allows the User to:

• Reset Password: request a password reset by entering a username, email, or phone number.

On the screen, s/he can also:

• Navigate to Login: return to the Login screen using the Login button.

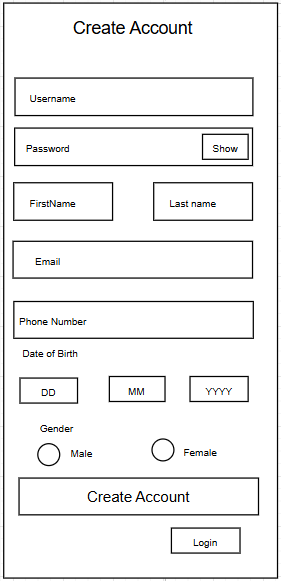
• Submit Reset Request: submit the reset request using the Reset button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Username/Email/Phone | Data type: string, max length of 100 characters (e.g., "user@gmail.com" or "0332526273"). Editable field for the user’s username, email, or phone number. |
| Reset | Interactive button to submit the password reset request. |
| Login | Interactive button to navigate to the Login screen. |

#### 

#### 3.3.16 Create Account Screen

****

This screen allows the User to:

• Create Account: register a new account by entering personal details such as username, password, first name, last name, email, phone number, date of birth, and gender.

On the screen, s/he can also:

• Navigate to Login: return to the Login screen using the Login button.

• Submit Account Creation: create the account using the Create Account button.

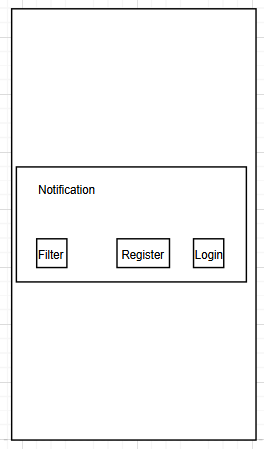
• Toggle Password Visibility: show or hide the password input field using the Show button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Username | Data type: string, max length of 50 characters. Editable field for the user’s username. |
| Password | Data type: string, max length of 50 characters. Editable field for the user’s password. |
| Show | Interactive button to toggle visibility of the password field (Show/Hide). |
| First name | Data type: string, max length of 50 characters. Editable field for the user’s surname. |
| Last Name | Data type: string, max length of 50 characters. Editable field for the user’s last name. |
| Email | Data type: string, max length of 100 characters (e.g., "user@gmail.com"). Editable field for the user’s email. |
| Phone Number | Data type: string, max length of 20 characters (e.g., "123-456-7890"). Editable field for the user’s phone number. |
| Date of Birth | Data type: string, format: DD/MM/YYYY (e.g., "25/04/2000"). Editable field with separate inputs for day (DD), month (MM), and year (YYYY). |
| Gender (Male/Female) | Options: Male, Female (radio buttons for selection). Editable field for the user’s gender. |
| Create Account | Interactive button to submit the account creation request. |
| Login | Interactive button to navigate to the Login screen. |

#### 

#### 3.3.17 Sign Up Dialog Screen



This screen allows the User to:

• Access Sign Up Options: interact with options to register, log in, or filter content.

On the screen, s/he can also:

• View Notifications: see notifications via the Notification button.

• Filter Content: use the Filter button to refine content (if applicable).

• Register: navigate to the Create Account screen using the Register button.

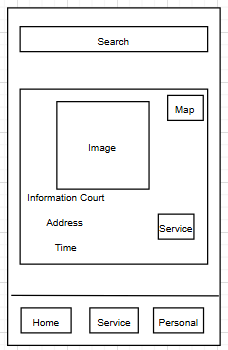
• Log In: navigate to the Login screen using the Login button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Notification | Interactive button to display notifications for the user. |
| Filter | Interactive button to filter content (if applicable). |
| Register | Interactive button to navigate to the Create Account screen. |
| Login | Interactive button to navigate to the Login screen. |

#### 

#### 3.3.18 Service screen

****

This screen allows the User to:

• Search Services: use the search bar to find services.

• Navigate Sections: access different sections of the application (Home, Service, Personal) via the navigation bar.

On the screen, s/he can also:

• View Service Information: see details about a service, including court information, address, and time.

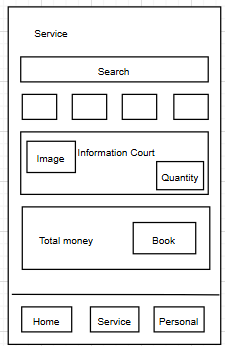
• View Map: interact with the Map to see the service location.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Search | Data type: string, max length of 100 characters. Editable field for searching services. |
| Search | Interactive element to view the service location on a map. |
| Image | Displays an image of the service (non-interactive). |
| Information Court | Data type: string, max length of 200 characters. Displays court-related details (non-interactive). |
| Address | Data type: string, max length of 200 characters. Displays the service address (non-interactive). |
| Time | Data type: string, max length of 50 characters (e.g., "09:00 AM - 05:00 PM"). Displays the service operating time (non-interactive). |
| Service | Interactive button to access the Service section (current screen). |
| Home | Interactive navigation button to access the Home section. |
| Personal | Interactive navigation button to access the Personal section. |

#### 

#### 3.3.19 Detail Service Screen

****

This screen allows the User to:

• Search Services: use the search bar to find services.

• Navigate Sections: access different sections of the application (Home, Service, Personal) via the navigation bar.

On the screen, s/he can also:

• View Service Details: see detailed information about a service, including court information and quantity.

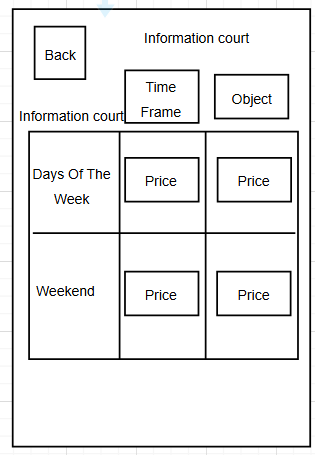
• Book Service: initiate a booking using the Book button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Search | Data type: string, max length of 100 characters. Editable field for searching services. |
| Image | Displays an image of the service (non-interactive). |
| Information Court | Data type: string, max length of 200 characters. Displays court-related details (non-interactive). |
| Quantity | Data type: string, max length of 50 characters. Displays the quantity or availability (non-interactive). |
| Total Money | Data type: string, max length of 50 characters (e.g., "đ 100.000"). Displays the total cost (non-interactive). |
| Book | Interactive button to initiate a booking for the service. |
| Service | Interactive navigation button to access the Home section. |
| Home | Interactive button to access the Service section (current screen). |
| Personal | Interactive navigation button to access the Personal section. |

#### 

#### 3.3.20 Detail Court Screen



This screen allows the User to:

• View Court Details: view detailed information about a court, including time, frame, object, and pricing.

• Navigate Back: return to the previous screen using the Back button.

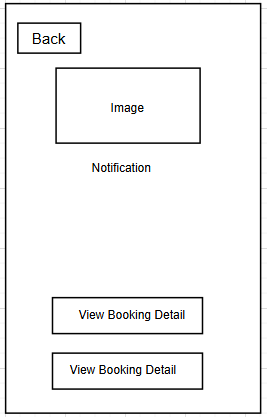
On the screen, s/he can also:

• View Pricing: see pricing details for days of the week and weekends across different frames and objects.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Back | Interactive navigation button to return to the previous screen. |
| Information Court | Data type: string, max length of 200 characters. Displays court-related details (non-interactive). |
| TimeSlot | Data type: string, max length of 50 characters (e.g., "09:00 AM - 05:00 PM"). Displays the operating time (non-interactive). |
| Frame | Data type: string, max length of 50 characters. Displays the time frame (non-interactive). |
| Object | Data type: string, max length of 50 characters. Displays the object or category (non-interactive). |
| Days of the Week | Data type: string, max length of 50 characters (e.g., "Monday - Friday"). Displays the days of the week (non-interactive). |
| Weekend | Data type: string, max length of 50 characters (e.g., "Saturday - Sunday"). Displays the weekend days (non-interactive). |
| Price | Data type: string, max length of 50 characters (e.g., "$50"). Displays the price for each frame and object (non-interactive). |

#### 3.3.21 Payment Successfully Screen



This screen allows the User to:

• View Payment Failure: see a confirmation of a failed payment attempt.

On the screen, s/he can also:

• Cancel History: cancel the payment history using the Cancel History button.

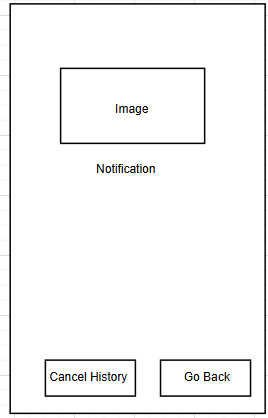
• Go Back: return to the previous screen using the Go Back button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Back | Displays an image indicating payment failure (non-interactive). |
| Notification | Data type: string, max length of 200 characters (e.g., "Payment Failed"). Displays the payment failure message (non-interactive). |
| Cancel History | Interactive button to cancel the payment history. |
| Go Back | Interactive button to return to the previous screen. |

#### 

#### 3.3.22 Payment Failed Screen



This screen allows the User to:

• View Payment Failure: see a confirmation of a failed payment attempt.

On the screen, s/he can also:

• Cancel History: cancel the payment history using the Cancel History button.

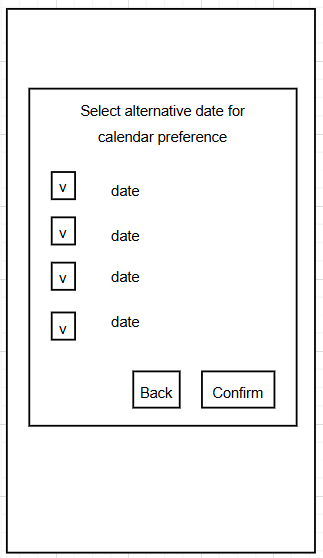
• Go Back: return to the previous screen using the Go Back button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Image | Visual element related to payment countdown (e.g., clock or warning symbol). Data type: image, input: system-generated, initial state: visible |
| Notification | Message informing the user about payment time running out or pending action. Data type: text display, input: system-generated, initial state: visible |
| Cancel History | Button to cancel the booking history or remove pending booking. Data type: action trigger, input: none, initial state: enabled |
| Go Back | Button to return to the previous screen. Data type: action trigger, input: none, initial state: enabled |

#### 

#### 3.3.23 Select alternative date for calendar preference Screen



This screen allows the User to:

• Select Alternative Dates: choose alternative dates for calendar preferences using checkboxes.

On the screen, s/he can also:

• Navigate Back: return to the previous screen using the Back button.

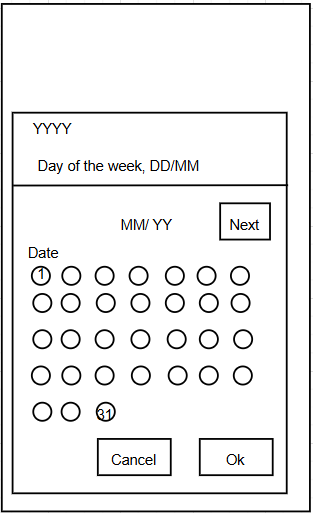
• Confirm Selection: confirm the selected dates using the Confirm button.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Image | Data type: string, max length of 10 characters (format: DD/MM/YYYY, e.g., "25/04/2025"). Checkbox options for selecting alternative dates (interactive). |
| Notification | Interactive navigation button to return to the previous screen. |
| Cancel History | Interactive button to confirm the selected alternative dates. |
| Go Back | Button to return to the previous screen. Data type: action trigger, input: none, initial state: enabled |

#### 

#### 3.3.24 Select date for booking daily Screen



This screen allows the User to:

• Select Booking Date: choose a date for booking using a calendar interface.

On the screen, s/he can also:

• Navigate Months: switch to the next month using the Next button.

• Confirm Selection: confirm the selected date using the OK button.

• Cancel Selection: discard the selection and return to the previous screen using the Cancel

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| YYYY | Data type: string, max length of 4 characters (e.g., "2025"). Displays the selected year (non-interactive). |
| Day of the Week, DD/MM | Data type: string, max length of 20 characters (e.g., "Friday, 25/04"). Displays the day of the week and date (non-interactive). |
| MM/YY | Data type: string, max length of 5 characters (e.g., "04/25"). Displays the selected month and year (non-interactive). |
| Next | Interactive button to navigate to the next month. |
| Date | Data type: string, max length of 10 characters (format: DD/MM/YYYY, e.g., "25/04/2025"). Interactive calendar dates for selection (1 to 31). |
| Cancel | Interactive button to discard the selection and return to the previous screen. |
| OK | Interactive button to confirm the selected date. |

#### 

## 4. Non-Functional Requirements

### *4.1 External Interfaces*

**User Interface:**

* The system must provide a **modern and user-friendly web interface** accessible via major browsers (Chrome, Firefox, Edge).
* The UI must follow **WCAG 2.1 accessibility standards**, ensuring usability for users with disabilities.
* Interface elements (buttons, menus, notifications) must be **responsive and mobile-friendly**, working seamlessly on desktop, tablet, and smartphone.

**Hardware Interface:**

* Compatible with common hardware platforms: **PCs, laptops, tablets, smartphones**.
* Must support integration with **time attendance devices** (e.g. fingerprint scanners) for staff check-in/out features (if implemented).

**Software / System Interface:**

* The system must support secure API integration with **external systems** (e.g., HRM or club membership systems).
* Must provide **RESTful API endpoints** for booking, user, payment, and inventory data exchange with partner services or mobile apps.

**Communication Interface:**

* All data exchange must be done over **HTTPS (TLS 1.2 or higher)** for security.
* The system must support **in-app notifications and reminders**, such as booking confirmations, payment alerts, and schedule changes.

### *4.2 Quality Attributes*

#### 4.2.1 Usability

* **Standardized Interface**:  
   Must follow modern UI patterns (e.g., Google Material Design or IBM Carbon Design) for consistency.
* **Consistency & Branding**:  
   All screens must use the same **colors, font families, icons**, and layout system to ensure brand recognition and intuitive use.
* **Responsive Design**:  
   Interface must automatically adapt to different screen sizes, orientations, and resolutions.
* **Clear Navigation**:  
   Use **bottom navigation (mobile)** and/or **left sidebar menu (web)** to simplify access to features like “Book Court”, “My Bookings”, “Profile”, etc.
* **Interactive UX**:  
  + Search bars must show **instant (real-time)** suggestions/results.
  + Calendar pickers must highlight **available/unavailable time slots** dynamically.
* **Error Handling**:  
  + Display short, clear error messages in **red**.
  + Show **green success messages** when actions are completed.

#### 4.2.2 Reliability

* **Availability**:  
   System uptime must reach **99.95%**, except during scheduled maintenance.
* **Accuracy**:  
  + Booking and payment data must be **99.99% accurate**, especially when calculating total amounts and applying discounts.
  + User access rights (Owner, staff, guest) must always reflect current status.

#### 4.2.3 Performance

* **Response Time**:  
   95% of user interactions (e.g., search, booking, payment) must respond in under **2 seconds**.
* **Throughput**:  
   System must handle up to **500 concurrent transactions per second** during peak usage (e.g., holidays, weekends).
* **Capacity**:  
   Must support at least **10,000 concurrent users** and **1 million transactions/day** without crashing.
* **Resource Usage**:  
   CPU usage must stay below **75%** and RAM below **70%** under normal usage conditions.
* **Scalability**:  
   The system must allow **horizontal and vertical scaling**, ensuring stable performance when expanding to new courts, locations, or features (e.g., tournament mode, equipment rental).

#### 4.2.4 Maintainability

* **Modular Codebase**:  
   The backend/frontend code should be **modular and component-based**, allowing isolated updates without system-wide impact.
* **Seamless Upgrades**:  
   The system must support **hot deployment/automatic upgrade mechanisms** that do not disrupt ongoing bookings or sessions.
* **Technical Documentation**:  
  + Must include full documentation for:  
    - REST API
    - User manual (for club staff and end-users)
    - System architecture (for dev/IT team)

### 

## 5. Requirement Appendix

### 5.1 Common Requirements

| **ID** | **Business Rule** | **Description** |
| --- | --- | --- |
| CR-01 | Cross-Platform Compatibility | The system must support web browsers (desktop/tablet/mobile) and mobile applications (Android). |
| BR-02 | Responsive Design | UI must adjust automatically across different screen sizes and devices. |
| BR-03 | Localization | The system must support Vietnamese and English languages, with easy switching. |
| BR-04 | QR Code Payment Integration | Payment system must support MB Bank QR payment generation and status verification. |
| BR-05 | Security Standards | Passwords must be encrypted. Sensitive user data must follow GDPR or equivalent local privacy protection standards. |
| BR-06 | Notification Management | Users should be able to mark notifications as read and manage notification preferences. |
| BR-07 | Error Handling | System must show clear error messages and guidance in case of failures (e.g., failed payment, failed booking). |
| BR-08 | Data Backup | Database must have scheduled backup at least once per day to prevent data loss. |
| BR-09 | System Availability | The system should maintain 99.5% uptime outside of scheduled maintenance windows. |
| BR-10 | Scalability | The system must be scalable to support future expansion to other cities or additional facilities. |

### 

### 5.2 Business Rules

| **ID** | **Business Rule** | **Description** |
| --- | --- | --- |
| BR-01 | Court Booking Limitations | Users can only hold one active booking for the same time slot. |
| BR-02 | Booking Payment Rule | A deposit or full payment must be completed before a booking is confirmed. |
| BR-03 | Refund Policy | Refunds are processed automatically, deducting the deposit amount. No refund if cancellation happens too close to the play time (within defined threshold). |
| BR-04 | User Role Access | Only Owners can create, update, or delete user roles. Managers and Staff have restricted access according to assigned permissions. |
| BR-05 | Maintenance Scheduling | Courts under maintenance cannot be booked. Courts must be deactivated during the maintenance period. |
| BR-06 | Service Availability | Only active services can be selected and ordered by users. Deactivated services are hidden from customer-facing screens. |
| BR-07 | Notification Delivery | Critical events (e.g., booking confirmation, payment success) must trigger real-time notifications via in-app and optionally email. |
| BR-08 | Session Timeout | User sessions expire after 30 minutes of inactivity for security reasons. |
| BR-09 | Student Discount Eligibility | Student discounts are granted only after email verification with a .edu.vn domain or submission of a valid student ID. |
| BR-10 | Financial Reporting | All revenue reports and transaction histories must be exportable into Excel format (.xlsx) for auditing and financial review. |

### 

### 5.3 System Messages

| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| --- | --- | --- | --- | --- |
| 1 | MSG01 | In line | No court/service/user found on search | *No search results.* |
| 2 | MSG02 | In red, under the text box | Required fields are left empty (e.g., booking info, service name) | *The \* field is required.* |
| 3 | MSG03 | Toast message | Update user/court/service information successfully | *Update asset(s) successfully.* |
| 4 | MSG04 | Toast message | Create new user/court/service successfully | *Created successfully.* |
| 5 | MSG05 | Toast message | Reset password successfully | *Password reset successfully.* |
| 6 | MSG06 | Toast message | Delete user/court/service/booking successfully | *Deleted successfully.* |
| 7 | MSG07 | In red, under the text box | Input value length > allowed maximum (e.g., name too long) | *Exceed max length of {max\_length}.* |
| 8 | MSG08 | In line | Incorrect username or password on login attempt | *Incorrect username or password. Please check again.* |
| 9 | MSG09 | Toast message | Booking cancelled successfully | *Cancel booking successfully.* |
| 10 | MSG10 | Toast message | Refund processed successfully | *Refund processed successfully.* |
| 11 | MSG11 | Toast message | Payment completed successfully | *Payment completed successfully.* |
| 12 | MSG12 | Toast message | Court maintenance schedule created successfully | *Maintenance scheduled successfully.* |

### 5.4 Other Requirements…