Timeshare Exchange Platform

Software Requirement Specification

TE: TE

Document Code: SP24SE17

<Ho Chi Minh, March 2024>

RECORD OF CHANGES

*A - Added M - Modified D - Deleted

Effective Date	Changed Items	A* M, D	Change Description	New Version
11/2/2024	Initial	a	Addition of the project overview and complete requirements.	

SIGNATURE PAGE

ORIGINATOR:

Hà Văn Phước (leader)

Lê Viết Quốc Thịnh

Trần Đại Nghĩa

Nguyễn Song Trọng Thắng

REVIEWERS: <Name>_____ <Date>_

<Position>

APPROVAL: <Name>_____ <Date>_

<Position>

Table Of Contents

1 INTRODUCTION

The Timeshare Exchange Platform aims to facilitate the efficient management and exchange of timeshares between owners and renters. It provides an integrated solution to enhance the booking process, manage listings, and ensure a seamless user experience for both parties involved.

1.1 Purpose

This project aims to develop a "Timeshare exchange platform" website, offering a comprehensive solution for managing and executing timeshare exchanges in the vacation industry between owners (LANDLORD) and RENTER. The system will enhance workflow efficiency, improve the monitoring of client reservation requests, and facilitate the evaluation and comparison of timeshare options. This will provide a visual understanding of vacation needs, helping clients make accurate and effective decisions

1.2 Scope

The "Civil housing construction quotation system" website is designed to support communication between clients, contractors (admins), and authorized staff. It offers features for tracking, creating, managing, and comparing quotations for residential construction projects. Additionally, it integrates with project and financial management systems to optimize workflow processes.

1.3 Definitions, Acronyms, and Abbreviations

- Timeshare Exchange Platform: A platform facilitating the exchange of timeshare ownership.

- Client: An individual interested in exploring timeshare pricing, submitting consultation requests, executing timeshare exchanges, or seeking other information about the company on the website.

1.4 References

IEEE 830-1998: Recommended Practice for Software Requirements Specifications

1.5 Overview

1. Introduction

This section lays the foundation for the SRS document, outlining its purpose, scope, and the definitions, acronyms, and abbreviations used throughout. It also includes references for further reading and an overview to give readers a snapshot of the document's contents.

2. Overall Description

Provides a high-level overview of the system, including its general capabilities, user interactions, constraints, and assumptions. This section sets the context for the detailed requirements that follow.

3. Functional Requirements

Detailed descriptions of the system's functionalities, represented through various use cases.

3.1 Use Cases Diagram: *Visual representation of the system's functionalities and their interactions with the users.*

3.2 - 3.38 Specific Use Cases: *Detailed scenarios that describe the system's interactions with users and other systems, covering everything from project management and user authentication to feedback management and notifications.*

4. Non-Functional Requirements

Specifies the criteria not directly related to the functionality of the system but critical for its effectiveness and user satisfaction. This includes usability, reliability, performance, supportability, design constraints, and more. Each sub-section outlines specific requirements that ensure the system's quality and operational integrity.

5. Supporting Information

Additional information that supports the understanding, development, and implementation of the system. This could include data models, additional requirements, guidelines for developers, and any supplementary documents or references needed for a comprehensive understanding of the system.

Non-functional Requirements:

1. **Performance**: The system must respond quickly, with a response time of no more than 3 seconds for each request.
2. **Scalability**: The system should be designed to easily accommodate user and data growth.
3. **Security**: The system must comply with high security standards to protect personal information and transaction data.
4. **Availability**: The system needs to be highly available, with a minimum uptime of 99.9%.
5. **Compatibility**: The system must be compatible with popular web browsers and mobile devices.

6. **Usability**: The user interface should be intuitive and easy to use for both clients and contractors.

Functional Requirements:

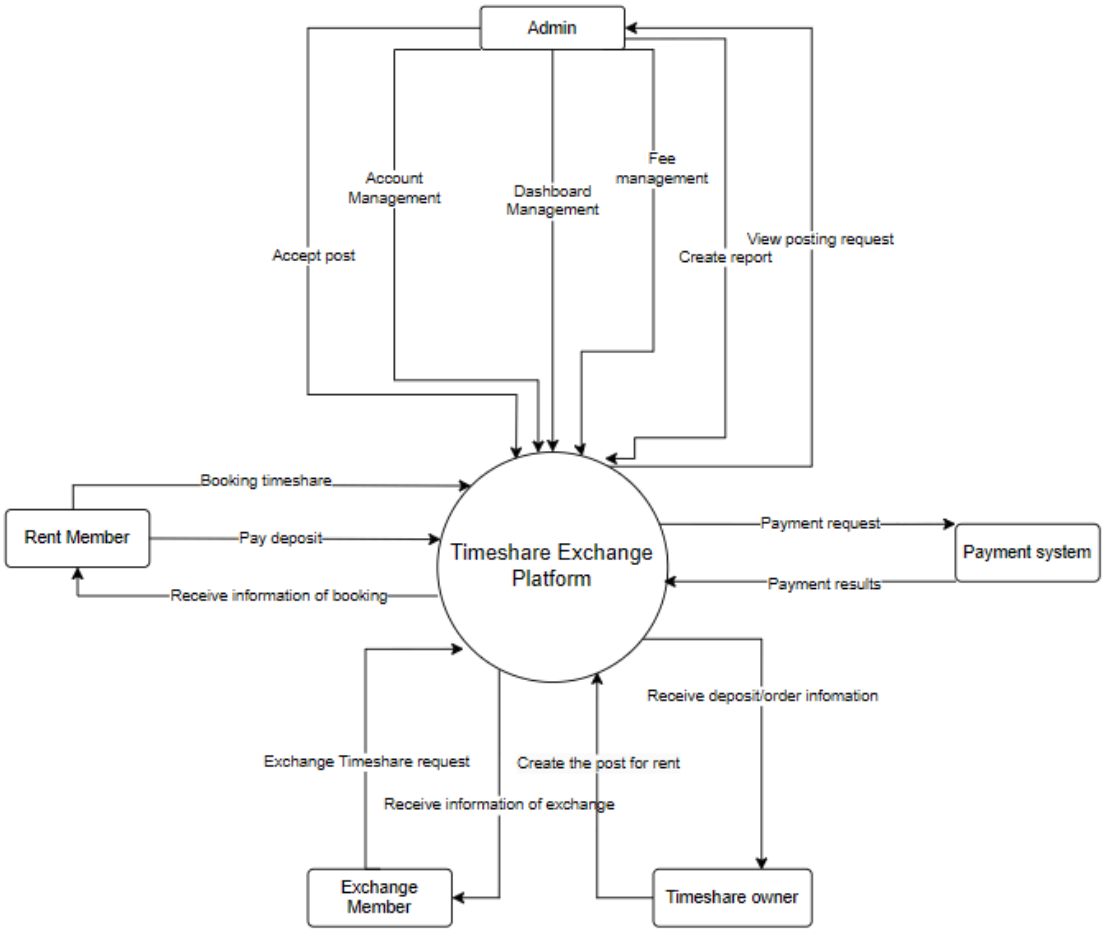
1. **Login/Register:** The system allows guests to create customer accounts.
2. **Profile Management:** Users can view and update their personal profile information.
3. **Create and Manage Quotes:** Contractors can create, review, and update quotes for projects.
4. **Search and Filter Quotes:** Customers can search and filter quotes based on criteria such as location, budget, and project type.
5. **Submit a Consulting Request:** Customers can send a consulting request directly to the contractor through the system.
6. **Review Feedback:** Both customers and contractors can provide reviews and feedback on projects or services.
7. **Notification:** The system will send notifications to users via email when there are new quotes, updates, or feedback.

2 OVERALL DESCRIPTION

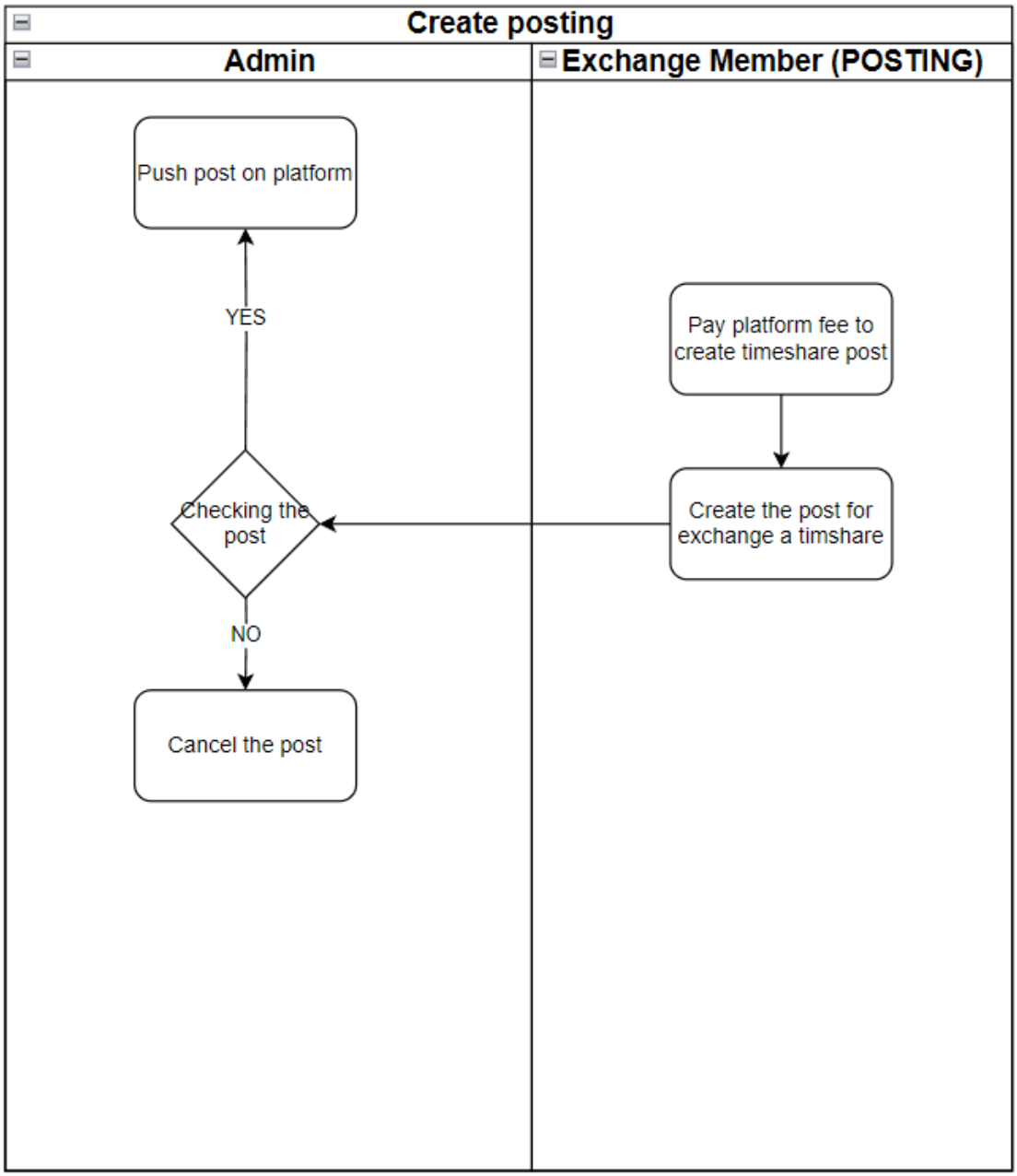
1. Product Overview

Our Timeshare Rental/Exchange system will also undergo a similar evolution, with the expectation that it will be linked to online booking services for many local areas and information confirmation services. credit and debit cards. This system not only makes it more convenient for users to make reservations and exchange timeshare usage rights, but also facilitates financial transactions through card validation services.

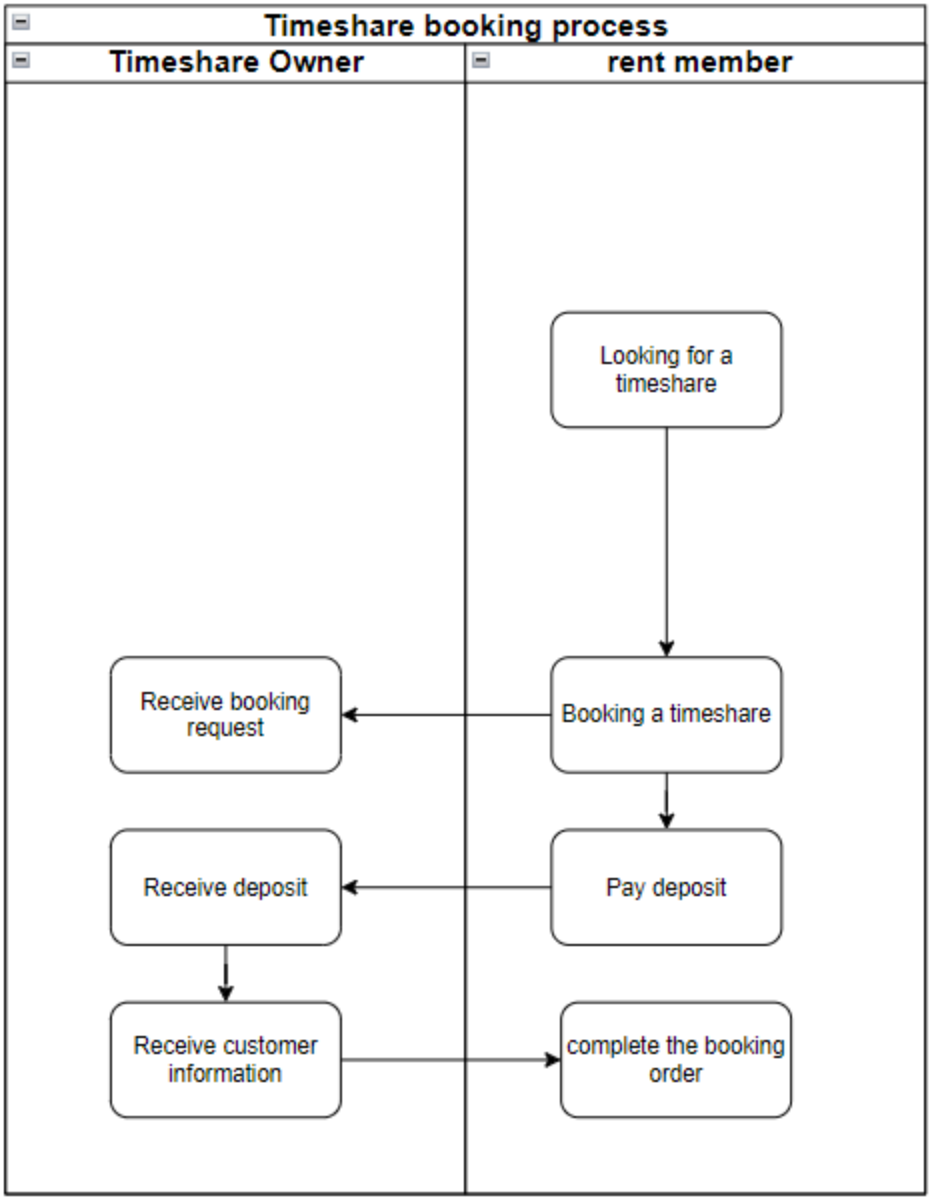
As with all new software, we anticipate that the system will undergo continuous updates to meet evolving user needs, including expanding connectivity to multiple booking processes. and add improved features. This will provide a flexible and comprehensive experience for those who want to enjoy the benefits of using and exchanging timeshares in the most convenient way.


Admin
Account Management
Dashboard Management
Fee management
Accept post
Create report
View posting request
Booking timeshare
Rent Member
Pay deposit
Receive information of booking
Timeshare Exchange Platform
Payment request
Payment results
Payment system
Receive deposit/order infomation
Exchange Timeshare request
Create the post for rent
Receive information of exchange
Exchange Member
Timeshare owner

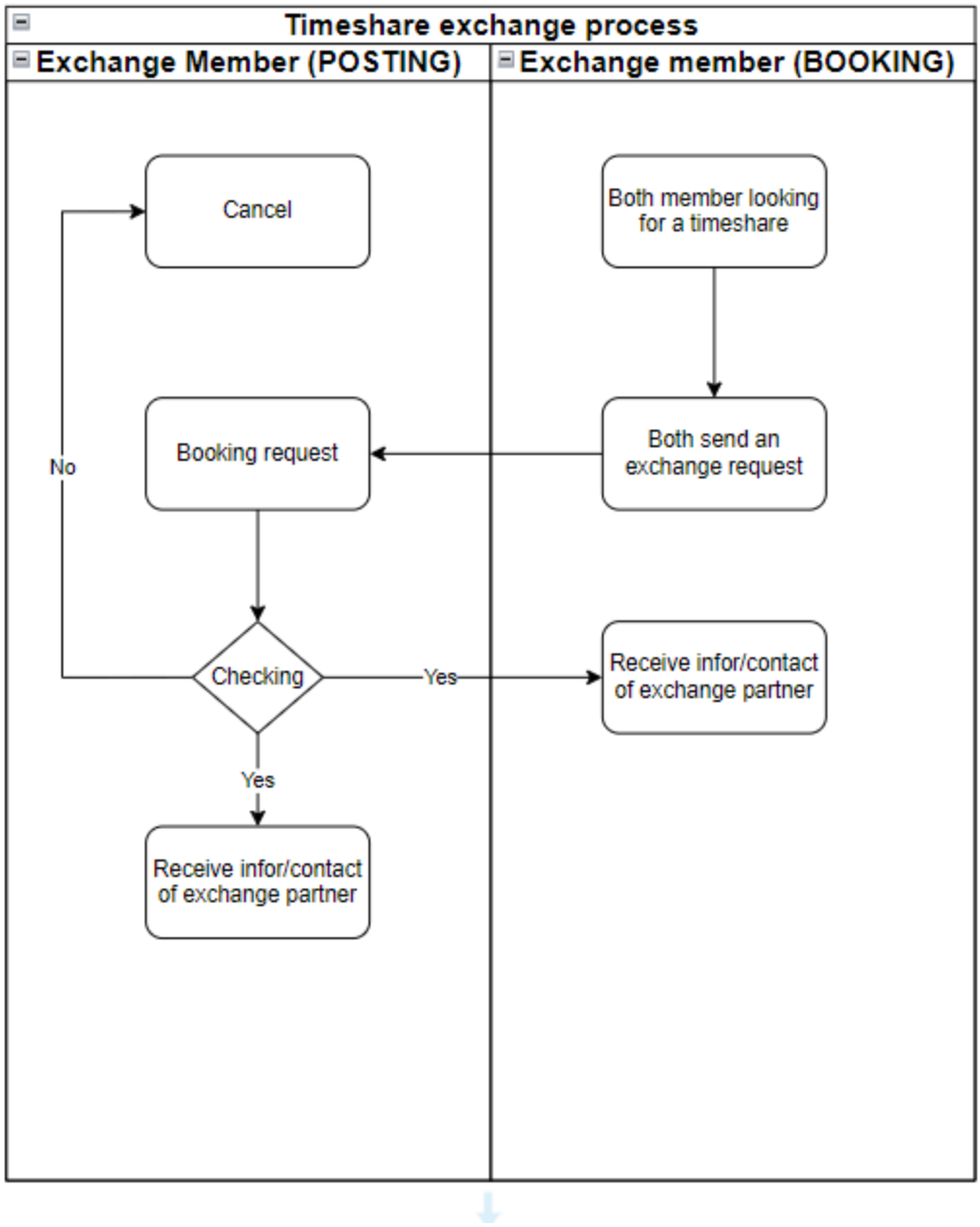
2. Business Process



#	Process Step	Description
1	Pay platform fee to create timeshare post	Members must pay a fee for the platform to create TimeShare posts
2	Create the post for exchange a timeshare	After paying for the Member platform, articles can be created
3	Admin:Checking the post	If the admin accepts the post will be approved and posted to the platform, if the admin refuses the post will be cancelled


Timeshare booking process
Timeshare Owner
rent member
Looking for a timeshare
Booking a timeshare
Receive booking request
Pay deposit
Receive deposit
Receive customer information
complete the booking order

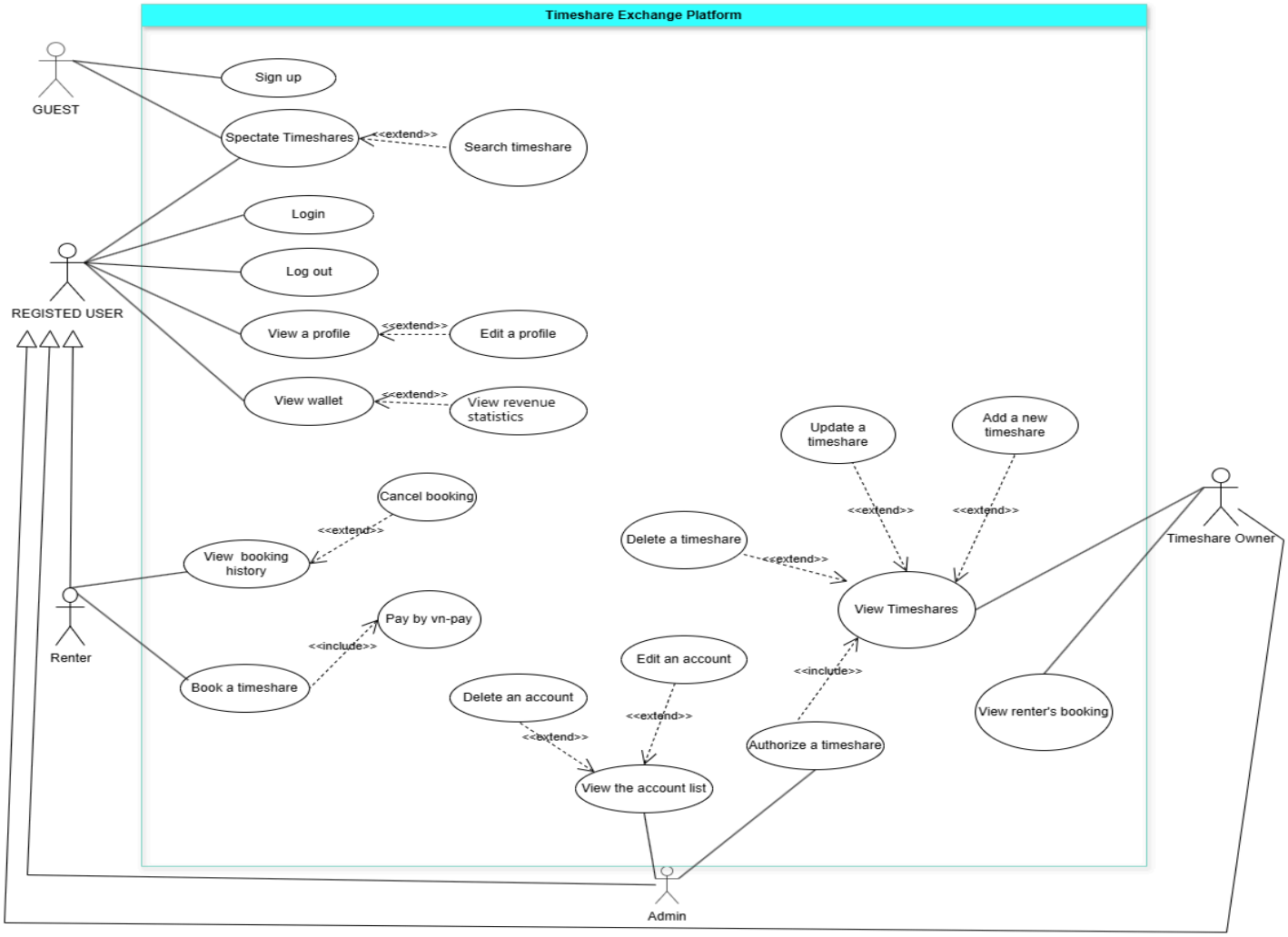
#	Process Step	Description
1	Looking for a timeshare	Rent members will search for each other on the platform from posts
2	Booking a timeshare	After finding a suitable post rent member booking a timeshare
3	Receive booking request	Timshare Owner receives a booking request from a rent member
4	Pay deposit	If the teacher is suitable, the rent member can make a deposit
5	Receive deposit	After the rent member makes a deposit, the timeshare owner receives the deposit
6	Receive customer information	TimeShare Owner receives rent member information
7	Complete the booking order	Transaction completed


Timeshare exchange process
Exchange Member (POSTING)
Exchange member (BOOKING)
Cancel
Both member looking for a timeshare
Booking request
Both send an exchange request
No
Checking
Yes
Receive infor/contact of exchange partner
Yes
Receive infor/contact of exchange partner

#	Process Step	Description
1	Both member looking for a timeshare	Two exchange members find each other on posts from the platform
2	Both send an exchange request	If deemed suitable, exchange booking can send a request to exchange pos
3	Booking request	Exchange posting received a request from exchange booking
4	Exchange Posting:Checking	If Exchange Posting terminates Exchange Booking's request, both parties will receive each other's information. If the request is not accepted, the transaction will be canceled.

3 FUNCTIONAL REQUIREMENTS

3.1 Use Cases Diagram



3.2 Use Case Sign Up

USE CASE-1 SPECIFICATION			
Use-case No.	UC001	**Use-case Version**	1.0
Use-case Name	Sign Up		
Author	Le viet quoc thinh		
Date	19/01/2024	**Priority**	High

Actor:

Member

Summary:

- This Use Case describes the process of members with button sign in or start now, sign up with form field information user, notification confirm sign up or error, link to home page

Triggers

- Actors want to sign up for a new account on the web application

Preconditions:

- 1.Member has click button sign up
- 2.Member has filed information the form
- 3.The member has notification successfully or error

Post Conditions:

- Success: The system successfully creates a new account and logs in.
- Fail: The system displays an error message, and the new account is not created.

Main Success Scenario:

No	Actor Action	System Response
1	Actor selects “Đăng ký” button	System shows a form with these fields: - Username: free text input, required, length 1 - 20. - Email: free text input, required, regex format /^\S+@\S+\.\S+$/ - Password: free text input, required, must have 1 special character and 1 capital letter. - Phone number: number text input, required, length 9 – 11. - OTP number: number text input, required, length 6.
2	Actor clicks “Gửi mã xác thực” after input the email.	System validates email and requests service sends OTP numbers to actor’s email.

3	Actor selects “Tạo tài khoản” button.	System checks OTP numbers. If correct, create a new account and redirect to Homepage.

Exceptions:

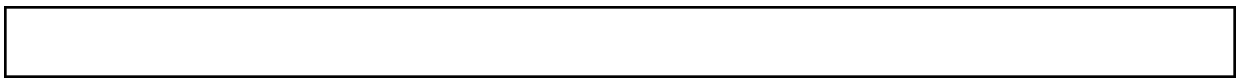
No	Cause	System Response
1	Actor enters incomplete information.	System shows error message: “Vui lòng điền vào mục này”.
2	Email or phone number existed in the system	System shows error message: “Tài khoản đã tồn tại trong hệ thống”.
3	OTP number is expired.	System shows error message: “Mã xác thực đã hết hiệu lực”.
4	OTP number is not correct.	System shows error message: “Mã xác thực không đúng”.

Business Rules:

- Only guests can sign up for a new account.
- Each new account must have a unique username and a valid email address.
- The email must be confirmed via a secret 6-digit number sent to the registered email address.
- Required fields must be filled in.

3.3 Use Case Book a Timeshare

USE CASE-2 SPECIFICATION			
Use-case No.	UC002	**Use-case Version**	1.0
Use-case Name	Book a Timeshare		
Author	Tran Dai Nghia		
Date	19/01/2024	**Priority**	High



Actor:

User, Customer

Summary:

This feature allows both guests and customers to book a timeshare on the Timeshare Exchange Platform.

Triggers

The actor wants to edit a project on the web application.

Preconditions:

The actor must be logged in.

Post Conditions:

A detailed booking confirmation is sent to the actor, including property details, dates, and payment information.

Main Success Scenario:

No	Actor Action	System Response
1	The user selects a timeshare.	Redirects to timeshare page and shows detail information such as: - Name of timeshare - Owner - Description - Location - Costs - Images - Booking form
2	The user choose booking form.	The system allows choose date selection and any specific preferences (view, additional amenities).
3	The user confirms the booking.	The system sends a detailed confirmation to the user's account and send booking request to Owner

Alternative Scenario:

N/A

Exceptions:

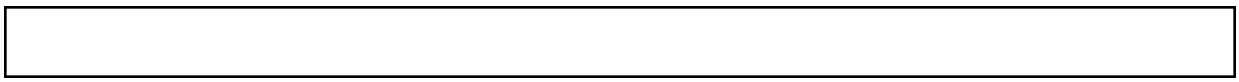
No	Cause	System Response
1	Project is not found in building projects database.	The system notifies the user and suggests alternative options.

Business Rules:

- *Both Guests and Customers can proceed with bookings.*
- *Contact information is required for booking confirmation.*

3.4 Use Case Login

USE CASE-6 SPECIFICATION			
Use-case No.	UC006	**Use-case Version**	1.0
Use-case Name	Login		
Author	Le Viet Quoc Thinh		
Date	19/01/2024	**Priority**	High



Actor:

Admin, Staff, Member

Summary:

This feature allows actors (Admin, Staff, and Members) to log in to their respective accounts on the web application.

Triggers

The actor wants to access their accounts on the web application.

Preconditions:

The actor is not logged in.

Post Conditions:

The system successfully logs in the actor, and they gain access to their respective account functionalities.

Main Success Scenario:

No	Actor Action	System Response
1	Actor selects “Đăng nhập” button.	System shows a login form with these fields: - Phone - Password
2	Actor enters email and password, then submits.	System validates email and password. If correct, redirect the actor to the homepage with the corresponding role (Admin, Staff, or Member).

Alternative Scenario:

Exceptions:

No	Cause	System Response
1	The email or password is not the correct format.	System informs error message to remind user inputs correct format.
2	Email or password is not correct.	System inform “Email hoặc mật khẩu không chính xác”.

Business Rules:

- Different actors have different roles (Admin, Staff, Member) with varying levels of access.
- Password is not empty and at least 8 characters
- Each actor must enter correct credentials to log in successfully.
- If a user enters incorrect login information more than 5 times, the account will be temporarily locked for 15 minutes.

3.5 Use Case Log out

USE CASE-1 SPECIFICATION			
Use-case No.	MA01	**Use-case Version**	<1.0>
Use-case Name	User Logging Out		
Author	Nguyen Song Trong Thang		

Date	16/01/2024	**Priority**	High

Actor:

- *User*

Summary:

- *This use case describes the process by which a user logs out of their account on a timeshare exchange platform, ensuring their account remains secure.*

Goal:

- *To securely log the user out of the platform, safeguarding their account.*

Triggers

- *The user selects the "Log Out" option on the platform.*

Preconditions:

- *The user must be logged into their account on the platform.*
- *There should be no ongoing transactions or critical processes associated with the user's account.*

Post Conditions:

- *The user is logged out of their account.*
- *Any session-related information is cleared from the user's device.*

Main Success Scenario:

Step	Actor Action	System Response
1	The user navigates to the account settings.	The platform displays the account settings page.
2	The user selects the “Log Out” option.	The platform confirms there are no ongoing processes.
3		The platform logs out the user, redirecting to the login page.
4		Session-related data is cleared from the device.

Alternative Scenario:

Step	Actor Action	System Response

1	The user selects the “Log Out” option.	The platform detects ongoing processes and alerts the user.
2	The user confirms the decision to log out.	The platform proceeds to log out the user, navigating back to the login page.
3		Session-related information is cleared from the device.

Exceptions:

Step	**Actor Action**	**System Response**
1	The user selects the "Log Out" option.	The platform experiences technical difficulties.
2	The platform experiences technical difficulties.	The platform displays an error message and advises retrying.

Relationships:

- Each user session on the platform is unique and directly associated with the individual user's account.

Business Rules:

- The platform reserves the right to implement logouts after specific periods or due to inactivity for enhanced security.
- The platform has the authority to terminate Renter and Member accounts under certain conditions.

3.6 Use Case View timeshares (Thắng)

USE CASE-9 SPECIFICATION			
Use-case No.	UC009	**Use-case Version**	1.0
Use-case Name	View timeshares		
Author	Nguyễn Song Trọng Thắng		
Date	19/01/2024	**Priority**	High

Actor:

Member, Landlord, Admin

Summary:
This use case involves the action of viewing timeshares, providing detailed information about specific building projects on the web application.

Triggers:
The user intends to access detailed information about a particular timeshare project on the web application.

Post Conditions:
Upon completion, the system displays detailed information about the selected timeshare project.

Main Success Scenario

No	Actor Action	System Response
1	The actor selects a specific timeshare project.	The system redirects to the project information page, displaying details such as project name, author, description, content, location, estimated time, and estimated costs.

Alternative Scenario::

None

No	Cause	System Response
1	The selected project is not found in the database.	The system redirects to the "not found" page (404).

Business Rules:

- Both Members and Guests have access to detailed project information. Guests may have restricted access to certain project details compared to Members.

3.7 Use Case View Renter's Booking (Thắng)

USE CASE-10 SPECIFICATION			
Use-case No.	UC010	**Use-case Version**	1.0
Use-case Name	View renter's booking		
Author	Nguyễn Song Trọng Thắng		
Date	19/01/2024	**Priority**	High

Actor: Member, Admin

Summary:

This use case enables Members and Admins to access and review detailed information about a renter's booking.

Triggers

The actor initiates the action of viewing a specific renter's booking information.

Preconditions:

The actor must be logged in as a Member or Admin.

Post Conditions:

The system displays the requested renter's booking information.

Main Success Scenario:

No	Actor Action	System Response
1	The actor navigates to the renter's booking.	The system retrieves and displays detailed booking information.
2	The actor reviews the booking details.	The system presents comprehensive renter's booking information.

Alternative Scenario:

None

Exceptions:

No	Cause	System Response
1	The requested booking information is not found.	The system notifies the actor that the booking information is not found.

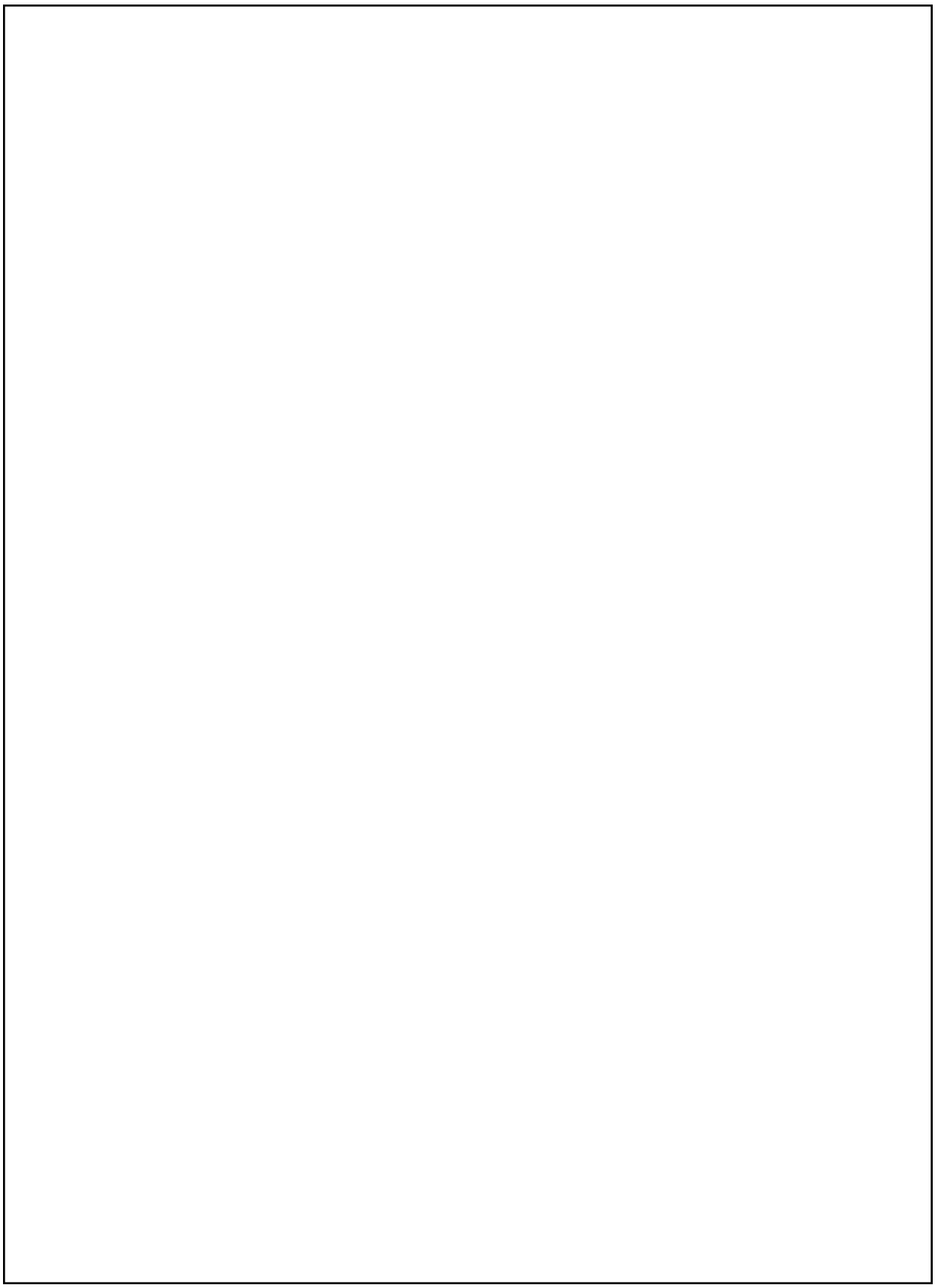
Business Rules:

- Only logged-in Members and Admins have access to renter's booking information.
- The system provides detailed booking information including renter's details, booking dates, property details, payment status, and any additional remarks.
- Admins may have additional privileges such as editing or canceling bookings.

3.8 Use Case Get User Account list

USE CASE-13 SPECIFICATION

Use-case No.	UC013	**Use-case Version**	1.0
Use-case Name	Get User Account list		
Author	Le Viet Quoc Thinh		
Date	19/01/2024	**Priority**	High



Actor:

Admin, Staff

Summary:

This feature allows Admin and Staff to retrieve a list of user accounts in the system.

Triggers:

The actor *wants to view the list of user accounts on the web application.*

Preconditions:

The actor has logged in using an Admin or Staff account.

Post Conditions:

The list of user accounts is displayed successfully.

Main Success Scenario:

No	Actor Action	System Response
1	Actor selects "Quản lý người dùng" button.	System shows a list of user accounts with the following details for each user: - User ID - Username - Email - Role (Admin, Staff, Customer, Guest) - Status (Active, Inactive, etc.)

Exceptions:

No	Cause	System Response
1	List user accounts is empty	System informs “Không có người dùng nào”.

Alternative Scenario:

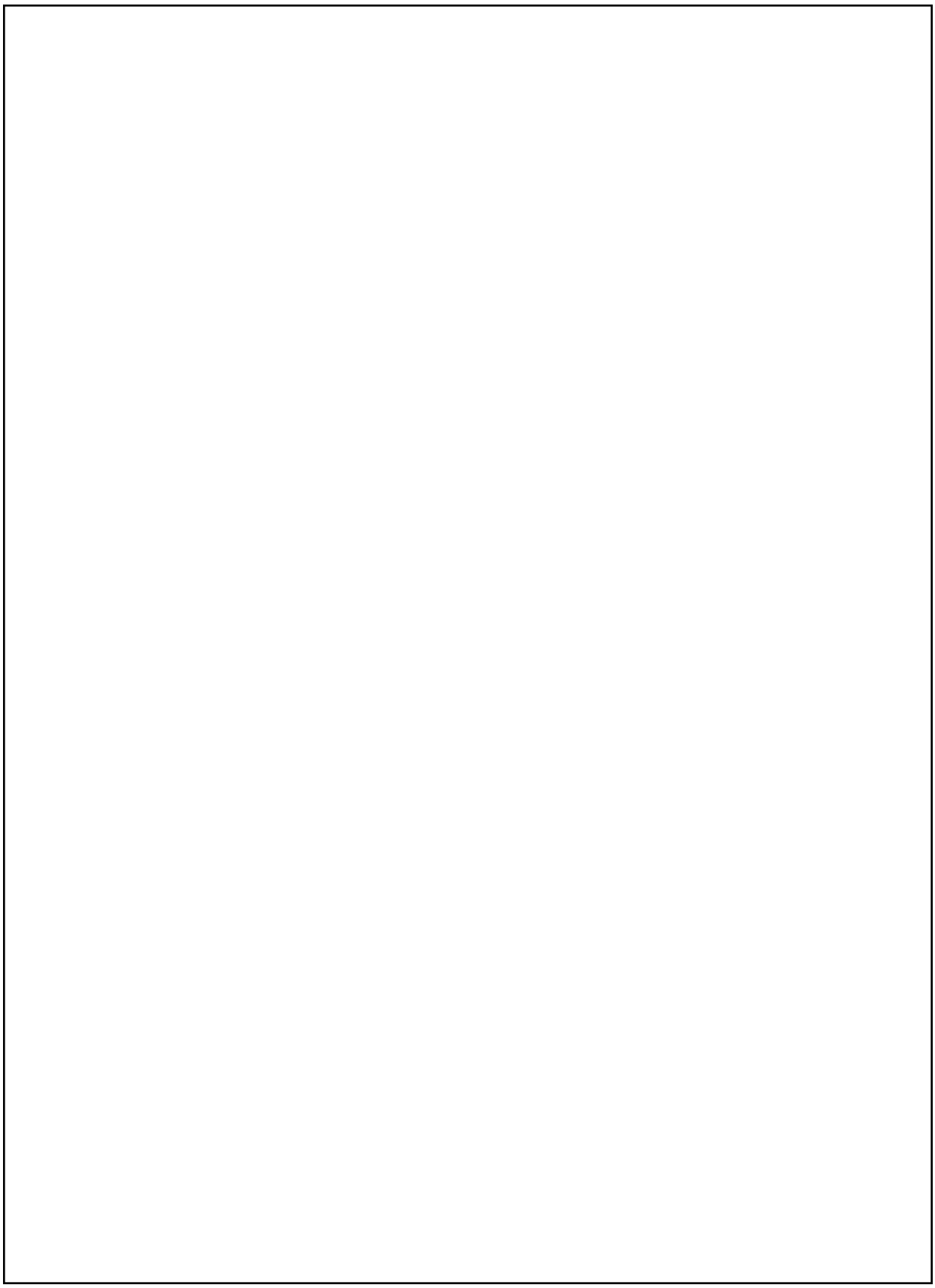
N/A

Business Rules:

- Only Admin and Staff can get the user account list.
- The user account list should be regularly updated to reflect any changes in user accounts.

3.9 Use Case Admin Manage Account

USE CASE-18 SPECIFICATION			
Use-case No.	UC019	**Use-case Version**	1.0
Use-case Name	Manage account		
Author	LE VIET QUOC THINH		
Date	16/01/2024	**Priority**	High



Actor:

Admin

Summary:

Manage all users that are logged into the database

Triggers:

Admin can manage user accounts by getting all user list and delete users.

Preconditions:

Admin is logged into the system

Post Conditions:

Admin successfully view and delete users accounts

Main Success Scenario:

No	Actor Action	System Response
1	In Admin screen after log in and tab Manage Account	The system shows the list of user detail : -id -username -email -role -encoded password

Alternative Scenario:

N/A

Exceptions:

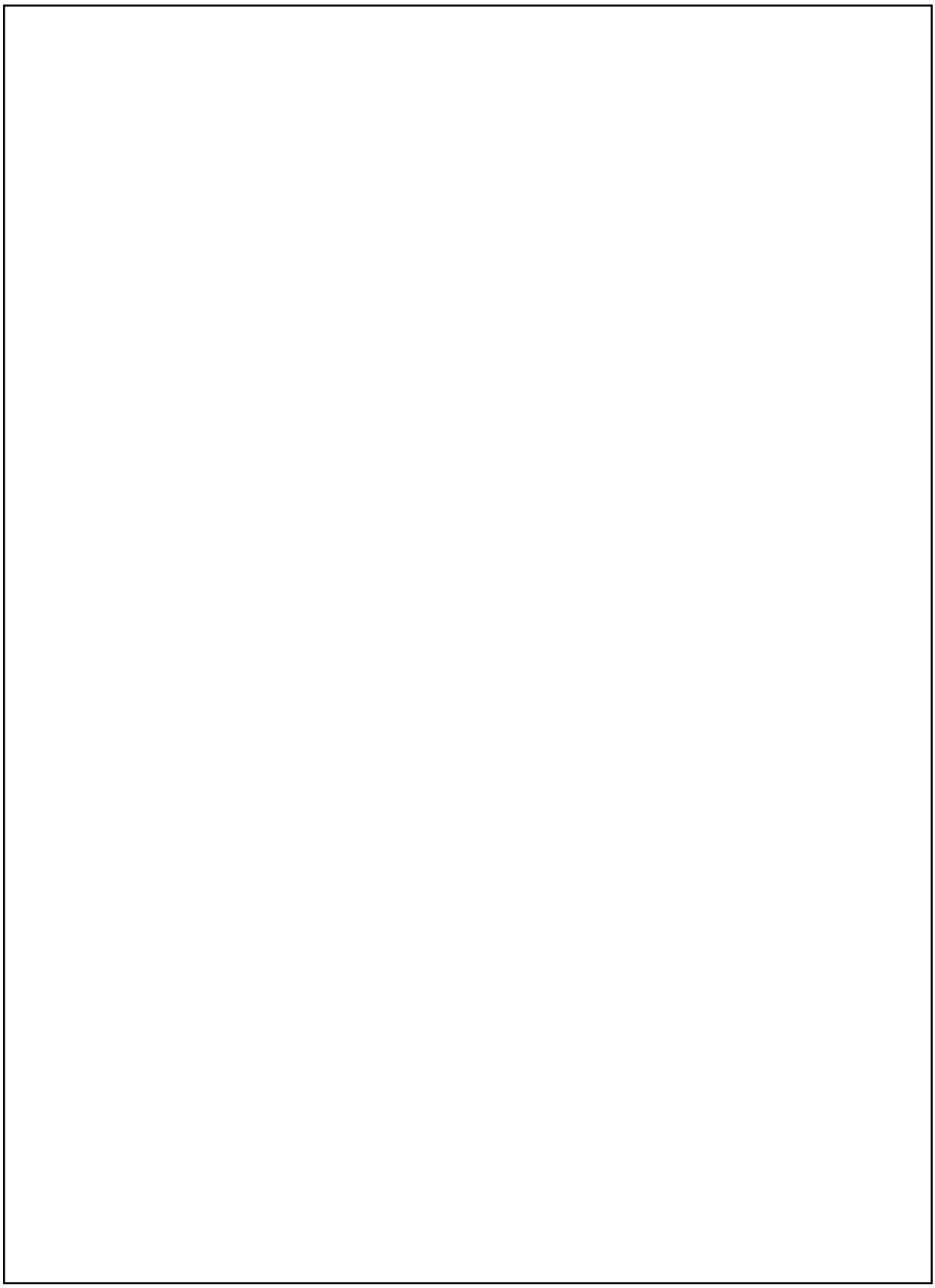
N/A

Business Rules:

-Only Admin has the authority to delete Renter and Member accounts.

3.10 Use Case Admin Manage Timeshares

USE CASE-18 SPECIFICATION			
Use-case No.	UC020	**Use-case Version**	1.0
Use-case Name	Manage Timeshares		
Author	LE VIET QUOC THINH		
Date	19/01/2024	**Priority**	High



Actor:

Admin

Summary:

Manage all LANDLORD real estates.

Triggers:

Admin navigates to the "Manage real estates" section

Preconditions:

Admin is logged into the system

Post Conditions:

Admin successfully authorizes from false to yes section of a real estate.

Main Success Scenario:

No	Actor Action	System Response
1	Admin navigate to "Manage timeshares"	The system handles the status of a post from "Inactive" to "Active" . Then shows message: "Đã duyệt bài viết thành công."

Alternative Scenario:

N/A

Exceptions:

N/A

Business Rules:

- Only the admin has the authority to authorize timeshares.

3.11 Use Case User Edit Profile

USE CASE-2 SPECIFICATION			
Use-case No.	MT01	**Use-case Version**	<1.0>
Use-case Name	User Editing Profile		
Author	Tran Dai Nghia		
Date	16/01/2024	**Priority**	High

Actor:

User

Summary:

- *This use case outlines the process for a user to update and modify their profile information on the timeshare exchange platform.*

Goal:

- *To enable the user to successfully edit and save their updated profile details.*

Triggers

- *The user selects the "Edit Profile" option within the account settings.*

Preconditions:

- *The user must be currently logged into their account on the timeshare exchange platform.*
- *The user has previously provided profile information that they want to update.*

Post Conditions:

- *The user's profile information is updated and saved.*
- *The platform reflects the changes in the user's profile.*

Main Success Scenario:

Step	Actor Action	System Response
1	User navigates to account settings.	Platform displays account settings page.
2	User clicks the "Edit Profile" option.	Platform displays the profile editing form.
3	User updates profile information.	Platform validates the information in real-time.
4	User clicks the "Save Changes" button.	Platform saves the changes and updates the profile.

Alternative Scenario:

Step	Actor Action	System Response
1	User updates profile information.	Platform detects errors, highlights them, and prompts correction.
2	User clicks the "Save Changes" button.	Platform displays an error message and prevents saving.
3	User corrects errors and clicks "Save Changes" again.	Platform saves the corrected information and updates the profile.

Exceptions:

Step	Actor Action	System Response
1	User clicks the "Save Changes" button.	Platform displays an error message and suggests retry.

Relationships:

- Each user's profile on the platform is unique and has a direct one-to-one relationship with the user.

Business Rules:

- *Some profile information fields may be mandatory, and the platform may impose limitations on the editability of certain details for security or verification purposes.*

3.12 Use Case User View Personal Profile

USE CASE-2 SPECIFICATION			
Use-case No.	MT01	**Use-case Version**	<1.0>
Use-case Name	User View Profile		
Author	Tran Dai Nghia		
Date	16/01/2024	**Priority**	High

Actor:

User

Summary:

- *This use case outlines the process for a user can check their profile on the timeshare exchange platform.*

Goal:

- *To enable the user to successfully view their profile details.*

Triggers

- *User selects the "View Profile" option within the account settings.*

Preconditions:

- *Users must be currently logged into their account on the timeshare exchange platform.*
- *The user has previously provided profile information that they want to view.*

Post Conditions:

- *The user's profile information is displayed.*

Main Success Scenario:

Step	Actor Action	System Response
1	User navigates to account settings.	Platform displays account settings page.
2	User clicks the "Edit View" option.	Platform displays the user’s profile details.

Alternative Scenario:

Step	Actor Action	System Response
1	User navigates to account settings.	Platform displays the account setting page.

2	User clicks the "View Profile" button.	Platform identifies an outdated session, asks for re-login, and then shows the user's profile.

Exceptions:

Step	Actor Action	System Response
1	User clicks the "View Profile" button.	Platform displays an error message if the user isn't authorized and suggests logging in.

Relationships:

- Each user's profile is unique and linked directly to that user.
- Edit Profile is considered an extension of View Profile.

Business Rules:

- *Only logged in users can view and edit their profiles.*

4 . NON-FUNCTIONAL REQUIREMENTS

1. External Interface Requirements

To ensure effective communication, a Timeshare Exchange Platform must:

- Provides an intuitive user interface to browse and book vacations.

- Integrate with secure payment gateways for online transactions.

- Send automatic email notifications for booking updates.

1.1 User Interfaces

UI-1: A help link shall be provided on every page of the system, offering users instant access to relevant guidance and instructions on how to effectively utilize the features and functionalities available on that specific page.

UI-2: The user interface shall feature intuitive and user-friendly design elements, including clear navigation menus, descriptive labels, and interactive elements, facilitating seamless interaction and enhancing the overall user experience.

UI-3: The system shall provide consistent feedback to users upon performing actions such as booking confirmation, error alerts, or status updates, ensuring transparency and keeping users informed about the outcomes of their interactions with the system.

UI-4: Accessibility features, such as keyboard shortcuts, screen reader compatibility, and adjustable text sizes, shall be incorporated into the user interface to cater to users with diverse needs and disabilities, ensuring inclusivity and compliance with accessibility standards.

1.2 Software Interfaces

SI-1: PayPal Payment Gateway Integration

- The TEP integrates with the PayPal Payment Gateway API (version X.X) to facilitate secure online transactions for timeshare bookings. This integration allows users to make payments using their PayPal accounts or credit/debit cards.
- Data exchanged: The TEP sends payment details such as transaction amount, currency, and user information to the PayPal API. Upon successful payment authorization, PayPal sends back a payment confirmation along with a transaction ID.
- Format of messages: Data is exchanged in JSON format over HTTPS protocol to ensure encrypted communication.
- Nonfunctional requirements: The interface must maintain PCI compliance standards for handling payment data and ensure transaction security by implementing tokenization and encryption mechanisms.

SI-2: User Account Management System

- The TEP communicates with the User Account Management System API (version X.X) to handle user authentication and authorization securely before initiating PayPal transactions.

- Data exchanged: The TEP sends user authentication tokens or credentials to the User Account Management System for verification. Upon successful authentication, the system provides authorization tokens indicating the user's access level.
- Format of messages: Data is exchanged in JSON format over HTTPS protocol to maintain data integrity and confidentiality.
- Nonfunctional requirements: The interface must enforce rate limiting and IP whitelisting to prevent unauthorized access attempts and ensure user account security.

SI-3: Email Notification Service

- The TEP interacts with the Email Notification Service API (version X.X) to send automated email notifications to users regarding PayPal payment confirmations.
- Data exchanged: The TEP sends email content and recipient information to the Email Notification Service API. Upon successful delivery, the service sends back a delivery confirmation.
- Format of messages: Data is exchanged in MIME format over SMTP protocol to ensure compatibility with email systems and reliable delivery.
- Nonfunctional requirements: The interface must guarantee email delivery within a specified time frame and handle bounced or undelivered emails gracefully to maintain communication reliability.

SI-4: Timeshare Listing Management System

- The TEP synchronizes with the Timeshare Listing Management System database to update the availability status of timeshare properties based on successful PayPal transactions.
- Data exchanged: The TEP sends booking details including property ID, booking dates, and transaction status to the Timeshare Listing Management System. Upon successful synchronization, the system updates the availability status of the respective property.
- Format of messages: Data is exchanged in SQL or JSON format over a secure connection to ensure data integrity and consistency.
- Nonfunctional requirements: The interface must handle concurrent updates gracefully to prevent data inconsistencies and ensure real-time availability updates for users. Additionally, it should implement data caching mechanisms to optimize performance during high traffic periods.

1.3 Hardware Interfaces

HI-1: ***Server Configuration***

Application Server

CPU	: Intel Xeon 8 Core 2.40 GHz
Memory Space	: 128 GB RAM 1333 MHz
Storage Space	: 2 TB
Operating System	: MS Windows Server 2012 R2
Software	: Microsoft IIS 8.x, .NET Framework 4.5

Database Server

CPU	: Intel Xeon 12 Core 2.40 GHz
Memory Space	: 256 GB RAM 1333 MHz
Storage Space	: 3 TB
Operating System	: MS Windows Server 2012 R2
Software	: Microsoft SQL Server 2000

HI-2: ***Client Configuration***

PC Device

CPU	: Intel Core 2 Dual 2.00 GHz
Memory Space	: 4 GB RAM 1333MHz
Storage Space	: HDD: 500GB, 2.5" SATA x 2, RAID 1
Operating System	: Windows Win7/Win8/Win10
Operator Display	: 18.5-inch widescreen, 16:9 format 1280 x 720 pixel, 1024 x 768 pixel

POS Device

CPU	: Intel® Celeron® Processor G1820TE (2.2GHz, Dual Core)
Memory Space	: 2GB (Max. 4GB), DDR3 SO-DIMM slot x 2 (1 open)
Storage Space	: HDD: 500GB, 2.5" SATA x 2, RAID 1
Operating System	: Windows Embedded POS Ready 7
Operator Display	: 15" XGA TFT color LCD with resistive touch screen (TeamTouch), Integrated 1024 x 768 pixel Proximity Sensor Option: Integrated Camera/Mic

Mobility Device

CPU	: Quad-Core 1.2 GHz
Memory Space	: 1.5 GB
Storage Space	: 8 GB
Operating System	: Android 8.0
Wifi Standard	: 5GHz Wi-fi
Operator Display	: 7.0 inches 1280 x 800 Pixels

HI-3: ***Network***

LAN Network : Speed ≥ 1 Gbps

WAN Network : Speed ≥ 2Mbps/10 Users operate together

1.4 Communications Interfaces

CI-1: The platform shall allow users to set and modify their preferred communication methods (e.g., email, SMS/text message, in-app notifications) for receiving confirmations, updates, and alerts related to timeshare exchanges.

CI-2: Upon successful submission of a timeshare exchange request, the platform shall send an email or text message (based on user account settings) to the user to confirm the request details, including exchange dates, locations, and any applicable fees.

CI-3: The platform shall send real-time updates via the user's preferred communication method regarding the status of their exchange request, including acceptance, rejection, or need for additional information.

CI-4: The platform shall send alerts for security-related events (e.g., login from a new device, password changes) using the user's preferred communication method to ensure account safety.

2. Quality Attributes

2.1 Usability

USE-1: The platform shall offer an intuitive navigation system, allowing users to find the information or feature they need within no more than three interactions from the home page.

USE-2: Users shall be able to initiate a timeshare exchange process with a single click or tap from the platform's home screen or dashboard.

USE-3: To facilitate ease of learning, the platform shall offer an optional interactive tutorial for first-time users, covering key features and actions required to perform a timeshare exchange.

USE-4: The platform shall implement inline validation for all input fields to provide immediate feedback on errors or omissions, minimizing the likelihood of failed submissions due to input errors.

USE-5: Users shall be able to quickly access a list of recently viewed timeshare listings with no more than two interactions, facilitating easier comparison and decision-making.

2.2 Performance

PER-1: The platform shall ensure that 95% of all pages are fully loaded and interactive within 2 seconds for users with a 50Mbps or faster Internet connection, under the assumption of average network conditions.

PER-2: The system shall update availability and status information for timeshare listings in real-time, with any changes reflected on the user's screen within 1 second under normal operating conditions.

PER-3: For timeshare exchange transactions, the platform shall process and return a confirmation to the user within an average of 4 seconds and a maximum of 8 seconds, assuming standard network conditions and server load.

PER-4: The search functionality shall return results within an average of 1 seconds and a maximum of 3 seconds, even during peak load times, with the system optimized for efficient query processing.

2.3 Security

SEC-1: All personal information, transactions and sensitive data must be encrypted to ensure security during transmission and storage.

SEC-2: User activities must be recorded and monitored to detect any unusual behavior early.

SEC-3: Systems should be periodically security tested to ensure that no security vulnerabilities exist and that current security measures remain effective..

SEC-4: The system must have protections against denial of service (DDoS) attacks and other attacks such as SQL injection or cross-site scripting (XSS).

SEC-5: The system needs to manage user sessions to prevent unauthorized or illegal access to accounts.

2.4 Safety

SAF-1: Requires assurance that users' personal information such as payment information, address and contact information is closely protected against risk of loss or misuse..

SAF-2: Require to ensure that the system has strong authentication mechanisms and correct authorization to prevent unauthorized access to sensitive information or critical functions.

SAF-3: Require encryption of financial information and transactions to ensure security and avoid financial fraud.

SAF-4: Implement security measures to protect against cyber attacks such as distributed denial of service (DDoS), SQL injection and cross-site scripting (XSS)

SAF-5: Require the use of security protocols and encryption to protect information during exchanges between parties.

SAF-6: Requirements ensure that the system is always ready to operate and capable of recovering quickly from problems

2.5 Availability

AVL-1: The system must be available at least 99% of the time for approximately 24 hours per day, including the period from 6:00 AM to 12:00 PM local time which is peak platform usage time. This ensures that users can access and use the service continuously and reliably

AVL-2: Downtime is defined as the time the system is inactive due to routine maintenance. This maintenance is often scheduled during low-load hours such as at night or on off-peak days of the week. This downtime must be minimized and not counted against system availability.

AVL-3: The system needs to be able to automatically switch to standby or backup mode when a failure or failure is detected. This helps minimize downtime and maintain system availability.

AVL-4: There should be a backup recovery plan designed to recover the system in the event of a catastrophic failure or failure.

AVL-5: The system needs to be able to warn early of any problems or incidents that could lead to service interruptions. This includes installing automated monitoring mechanisms to monitor system performance and detect signs of problems before they become serious. Alerts provide information to the admin team so they can intervene promptly and prevent service disruptions.

2.6 Reliability

REL-1: No more than 5 experimental runs out of 1,000 can be lost because of software failures.

REL-2: The mean time between failures of the booking system component shall be at least 90 days.

REL-3: The system should maintain a uptime of at least 99.9% over a period of one year, excluding scheduled maintenance windows.

REL-4: The platform's database backups must be performed daily, with a retention period of at least 30 days, ensuring data integrity and recovery in case of system failures.

REL-5: In the event of a server outage, the system should automatically failover to a backup server within 5 minutes to minimize downtime and maintain uninterrupted service.

REL-6: The platform should be capable of handling concurrent user sessions without degradation in performance, supporting a minimum of 10,000 simultaneous users during peak hours.

REL-7: Critical system components, such as the authentication and payment processing modules, must have redundancy measures in place to ensure continuous operation in case of component failure.

REL-8: Regular performance monitoring and load testing should be conducted to identify and address any potential bottlenecks or performance issues before they impact users.

REL-9: The platform's disaster recovery plan should be regularly reviewed and tested to ensure swift recovery and minimal data loss in the event of a catastrophic failure or disaster scenario.

REL-10: Software updates and patches should be applied in a timely manner to address security vulnerabilities and improve system reliability, with minimal disruption to users.

2.7 Design Contrainsts

DES-1: Programming Languages: The software will be developed using Python for the backend and React.js for the frontend.

DES-2: Web Server: The application will be hosted on Apache HTTP Server version 2.4.

DES-3: Web Browser: The application will be compatible with the latest versions of popular web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge.

DES-4: Database: The system will utilize PostgreSQL version 12 as the primary relational database management system (RDBMS).

DES-5: Reporting: Reporting functionalities will be implemented using a combination of built-in reporting tools in PostgreSQL and external libraries such as matplotlib for data visualization.

DES-6: Design Tools: The software architecture and database design will be created using tools such as Lucidchart for diagrams and PostgreSQL Workbench for database modeling.

DES-7: Development Environment: The development environment will consist of Git for version control, Docker for containerization, and Visual Studio Code as the integrated development environment (IDE) for code editing.

DES-8: Frameworks and Libraries: The backend will be built using the Django framework, while the frontend will utilize React.js for building user interfaces and Redux for state management.

DES-9: Operating System: The application will be deployed on Linux-based servers, specifically Ubuntu Server 20.04 LTS.

DES-10: Deployment Platform: The application will be deployed on cloud infrastructure, utilizing services such as Amazon Web Services (AWS) or Microsoft Azure for scalability and reliability.

2.8 Others as relevant

Efficiency:

Efficiency is crucial for ensuring that the software operates smoothly and responds promptly to user actions.

The software should load timeshare listings and perform database queries within 2 seconds on average.

The priority for efficiency is high, as users expect a responsive and fast system.

Installability:

Installability refers to the ease of installing and deploying the software on different environments.

The software installation process should be straightforward and well-documented.

The software should be deployable on both Windows and Linux operating systems without significant configuration changes.

The priority for installability is moderate, as it affects the ease of adoption by users and administrators.

Interoperability:

Interoperability ensures that the software can seamlessly integrate and communicate with external systems or services.

The software should support standard data exchange formats such as JSON and XML for interoperability with third-party applications.

It should also provide APIs or web services for integration with external systems.

The priority for interoperability is high, especially for enabling data exchange with other platforms or services.

Modifiability:

Modifiability reflects the ease with which the software can be modified or extended to accommodate changes or new requirements.

The software architecture should be modular and well-documented to facilitate future modifications.

Changes to business rules or user interfaces should be implementable without affecting the entire system.

The priority for modifiability is high, as the software needs to adapt to evolving business needs and user feedback.

Portability:

Portability refers to the ability of the software to run on different hardware and software platforms without modification.

The software should be developed using platform-independent technologies and libraries.

It should be deployable on various cloud platforms and on-premises infrastructure.

The priority for portability is moderate, as it ensures flexibility in deployment options but may not be immediately critical for all users.

Reusability:

Reusability involves designing software components that can be reused in different parts of the system or in future projects.

Common functionalities such as authentication and authorization should be implemented as reusable modules.

Code documentation and clear interfaces should facilitate the reuse of components.

The priority for reusability is moderate, as it contributes to development efficiency and maintenance.

Robustness:

Robustness ensures that the software can handle unexpected inputs or errors gracefully without crashing or compromising security.

The software should include error handling mechanisms to prevent data loss or corruption.

It should validate user inputs and sanitize data to prevent security vulnerabilities.

The priority for robustness is high, as it is essential for maintaining data integrity and user trust.

Scalability:

Scalability refers to the ability of the software to handle increasing loads or users without significant performance degradation.

The software architecture should be designed to scale horizontally by adding more servers or instances.

Database queries and data processing should be optimized for efficiency to support a growing user base.

The priority for scalability is high, especially for platforms expecting rapid growth or fluctuations in user traffic.

Verifiability:

Verifiability ensures that the software behavior can be tested and verified against specified requirements.

The software should include automated test suites covering functional and non-functional requirements.

Test cases should be designed to validate system behavior under different scenarios and edge cases.

The priority for verifiability is high, as it ensures the reliability and correctness of the software.

3.13

Table Of Contents

- **1.1 Use Cases Diagram**