FPT UNIVERSITY

Vinhomes Grand Park On-demand Service Software Requirement Specification

Project Code: VinGiG

Document Code: SRS – v1.1

Group Members:

Nguyen Hoang Nam
Nguyen Ba Huy
Le Trung Duc
Le Nguyen Tien Nhat

Ho Chi Minh City, June 26th, 2023

TABLE OF CONTENT

1. Introduction

1.1 Purpose

This Software Requirements Specification provides readers with a closer look at the behaviors and the functionalities of the application VinGiG, Vinhomes Grand Park's on-demand task service platform. The SRS encompasses a high-level use-case diagram, actors, use case details and other non-functional requirements of the system. The business rules are distributed across the use-case details where relevant. Additional information includes the names of authors, publish date and the scope that the project is in.

1.2 Scope

- This SRS applies to the VinGiG application, a third-party website for Vinhomes Grand Park residents to book on-demand local services or register to become a service provider.
- The application offers services such as AC services (repair/maintenance/cleansing), chores, Home maintenance (repainting/door battery replacement, sink and toilet unclogging), home-based care (baby sitting/elderly catering), medical home treatment (wound care/drip medication), culinary services (market shopping/cooking), and laundry.
- Features and functionality:
    - Admin's management:
        - FE-ADM-01: Functions to manage buildings
        - FE-ADM-02: Functions to manage badges
        - FE-ADM-03: Functions to manage service categories
        - FE-ADM-04: Functions to manage services in a category
        - FE-ADM-05: Functions to manage subscription plans
        - FE-ADM-06: Functions to manage Customers

- FE-ADM-07: Functions to manage Providers and Wallets
- FE-PRO-08: Functions to view statistics of booking fees, subscription fees, transaction

- Customer's features:
  - FE-CUS-01: Functions to view provider services in a service option
  - FE-CUS-02: Functions to view all services offered by a provider on the provider's page
  - FE-CUS-03: Functions to place a booking
  - FE-CUS-04: Functions to chat with booked provider and view chat history
  - FE-CUS-05: Functions to view current booking activities
  - FE-CUS-06: Functions to view booking history
  - FE-CUS-07: Functions to feedback and rate bookings completed

- Provider's features:
  - FE-PRO-01: Functions to view all the services on his/her page
  - FE-PRO-02: Functions to mark the provider's availability status
  - FE-PRO-03: Function to manage (add, edit, delete, hide) the services on his home page
  - FE-PRO-04: Functions to get notified for the booking placement
  - FE-PRO-05: Functions to accept/deny a booking
  - FE-PRO-06: Functions to mark complete/cancel the booking upon acceptance.

- FE-PRO-07: Functions to chat with booked customer and view chat history
- FE-PRO-08: Functions to view current booking activities
- FE-PRO-09: Functions to view booking history
- FE-PRO-10: Functions to view monthly statistics of booking fees, subscription fees, deposits
- FE-PRO-11: Functions to manage wallet and view transactions
- FE-PRO-12: Functions to register a subscription plan and get notified when the due date is close.
- FE-CUS-13: Functions to feedback and rate customers upon booking completion.

1.3 Definitions, Acronyms, and Abbreviations

- GiG: on-demand tasks that are needed occasionally and implies little loyalty.
- Customer: registered users, who place a booking for a particular service.
- Provider: a registered service provider who creates posts of services, responds to the booking placement of the customers and must adhere to the policies and the services available to the system.
- Provider Service: each service has many providers, each of which is unique and of the discretion of the provider.

1.4 References

1.5   Overview

This SRS document consists of two main sections: Overall Description and Specific Requirements. The Overall Description section provides a general understanding of the product, its functions, user characteristics, constraints, assumptions, dependencies, and requirement subsets. The Specific Requirements section contains detailed requirements for each user category (Customer, Provider, and Admin) along with their respective features and operations. The section also includes non-functional requirements related to the technology stack used for the development of the application.

2. Overall Description

The VinGiG application is a third-party website that enables users to book various on-demand local services provided by Vinhomes Grand Park. The application serves as a platform for connecting users and service providers. The following factors affect the product and its requirements:

- Product Perspective:

The VinGiG application is an independent system that interacts with users, service providers, and administrators.

It integrates with the Vinhomes Grand Park service infrastructure to facilitate seamless service bookings and management.

- Product Functions:

User Account and Authentication: Users can create accounts, log in, manage their account information, and change their passwords.

Service Booking: Customers can browse available services, select a provider, chat with providers, and rate/review providers after receiving services.

Booking History: Customers can view their booking history.

Provider Profile: Providers can view and edit their information.

Service Operation: Providers can select services to provide, chat with customers, confirm order placements, and confirm service completions.

Service History: Providers can view their service history, including ratings, reviews, badges, and order quantity.

Admin Management: Admins can manage services, customers, providers, and financial aspects of the application, including the ability to modify service details, register providers to services, view order histories, and adjust commission fees.

- User Characteristics:

Customers: Individuals who require on-demand local services.

Providers: Service professionals who offer their expertise and services through the platform.

Administrators: Responsible for managing the application, overseeing customer/provider interactions, and maintaining the financial aspects of the system.

- Constraints:

Front-end technology: React

Back-end technology: Java frameworks (Spring Boot)

- Assumptions and Dependencies:

The VinGiG application assumes an internet-connected environment for customers, providers, and administrators to access and interact with the platform.

The application depends on the availability and functionality of the Vinhomes Grand Park service infrastructure for seamless service bookings and management.

- Requirements Subsets:

The SRS defines requirements for customer management, service booking, provider management, and admin management, addressing the needs of each user category within the VinGiG application.

3. Functional Requirements

3.1. Use Case Diagram

Figure: Use Case Diagram

3.2. List of Use Cases

3.2.1.   <Guest> Overview Use Case

Figure: \<Guest\> Overview use case

3.2.1.1. \<Guest\> Login
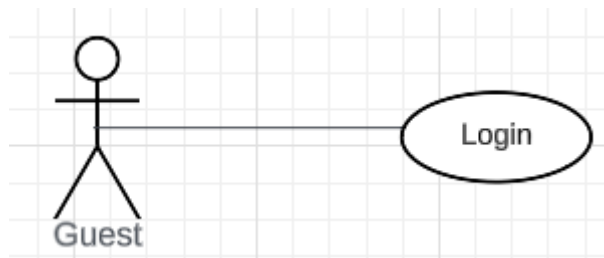
Figure: \<Guest\> Login

Use Case Specification

USE CASE – UC01			
Use Case No.	**UC01**	Use Case Version	1.1
Use Case Name	Login		
Author	Nguyen Ba Huy		
Date	07/06/2023	**Priority**	Normal

Actor:

-         Guest

Summary:

-         This use case allows the user to login to the system.

Goal:

-         Guest can login to the system

Trigger:

-         Guest inputs username and password and clicks login

Precondition:

-         Guest is on login screen

Post Conditions:

-         Success: User logs into the system successfully

Main Success Scenario:

Step	Actor Action	System Response
1	Guest goes to the login screen	The system requires the following inputs from User include: · Username: text, required · Password: text, required . Role: checkbox, required
2	User inputs information.	

| 3 | User clicks button "Login" | The system shows a message that the user has logged in successfully [Exception 1] |

Exception 1:

Step	Cause	System Response
1	Invalid username or password	System shows the message "Username or password is not correct."

Relationship: N/A.

Business rule:

- User password character is secured and hidden with special character (*)

3.2.1.2. <Guest> Register
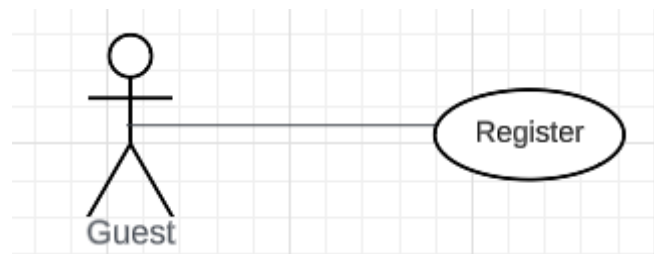
Figure: <Guest> Register

Use Case Specification

USE CASE - UC02			
Use Case No.	UC02	Use Case Version	1.1
Use Case Name	Register a new account		
Author	Nguyen Ba Huy		
Date	07/06/2023	**Priority**	Normal

Actor:

-        Guest

Summary:

-        This use case allows user to register an account

Goal:

-        Guest can register an account

Trigger:

-        The guest inputs information and clicks the "Register" button.

Precondition:

-        Guest is viewing the register screen.

Post Conditions:

-        Success: User registers a new account in the system successfully.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest goes to the register screen	The system requires the following input from Guest includes: · Username: text, required · Password: text, required · Confirm password: text, required · Name: text, required · Email: text, required · Phone: text as number, required . Role: checkbox, required

2	Guest inputs information.	
3	Guest clicks "Register"	The system check validation and show message that the user has registered successfully [Exception 1]

Exception 1:

Step	Cause	System Response
1	Username is used by another account	System shows message "Username is unavailable"

Relationship: N/A.

Business rule:

·   Password and confirm password field must be hidden with special character and must be the same.

·   The system validates user input with the following constraints:

    o   Username: unique and have more than 6 and fewer than 20 characters

    o   Phone number: number a format

    o   Email: email a format

    o   Password: Have fewer than 20 characters

    o   The confirm password matches the password

    o   Only one of three role checkboxes must be checked

3.2.1.3. <Guest> View service providers
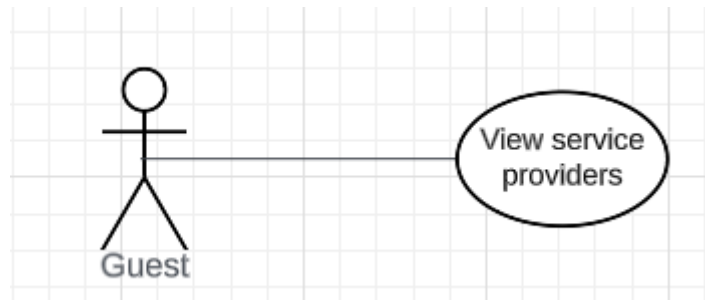
Figure: <Guest> View service providers

Use Case Specification

USE CASE - UC03			
Use Case No.	**UC03**	**Use Case Version**	1.1
Use Case Name	View service providers		
Author	Nguyen Ba Huy		
Date	07/06/2023	**Priority**	Normal

Actor:

- Guest

Summary:

- This use case allows the user to view providers of a service in the system.

Goal:

- Guest can see the list of service providers.

Trigger:

- Guest clicks on one service from the service list in the homepage.

Precondition:

- Guest is on the home screen.

Post Conditions:

- Success: List of all providers of the service chosen will be displayed on screen.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest clicks on one service from the service list in the homepage.	Show all providers of the service chosen including details: . provider name . rating . number of past bookings . unit price

| | | | . service description |
| | | | . provider availability |

Relationship: N/A.

Business rule:

    o Display provider avatar when showing service provider.

    o Display provider service detail when showing service provider including:

        . provider name

        . rating

        . number of past bookings

        . unit price

        . service description

        . provider availability

3.2.2.   <Customer> Overview Use Case
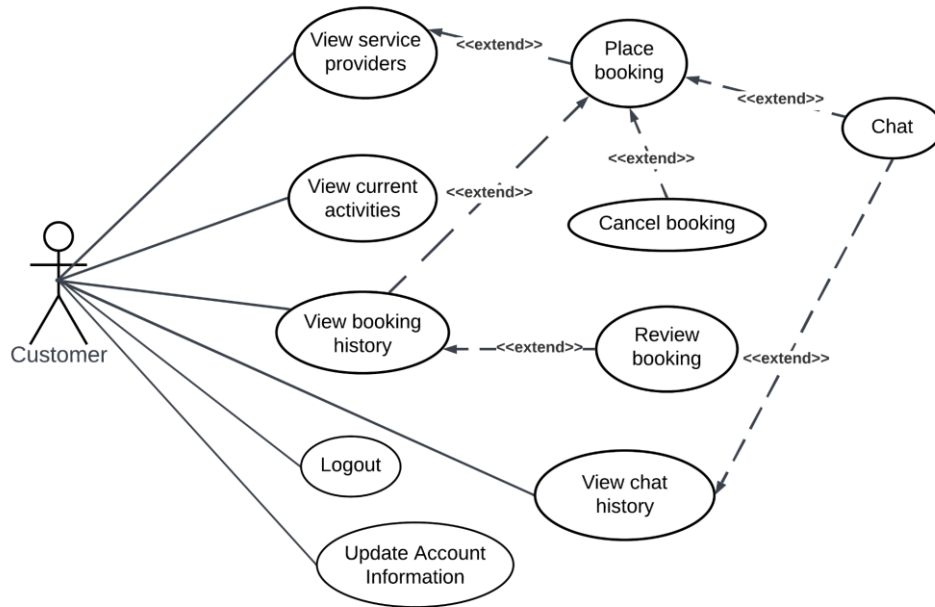
Figure: <Customer> Overview use case

3.2.2.1. <Customer> View service providers

Figure: <Customer> View service providers

Use Case Specification

USE CASE - UC04			
Use Case No.	**UC04**	**Use Case Version**	1.1
Use Case Name	View service providers		
Author	Nguyen Ba Huy		
Date	17/06/2023	**Priority**	Normal

Actor:

-        Customer

Summary:

-        This use case allows the user to view providers of a service in the system.

Goal:

-        Customer can see the list of service providers.

Trigger:

-        Customer clicks on one service from the service list in the homepage.

Precondition:

-        Customer is on the home screen.

Post Conditions:

-        Success: List of all providers of the service chosen will be displayed on screen.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer clicks on one service from the service list in the homepage.	Show all providers of the service chosen including details: . provider name . provider service's rating . number of past bookings . unit price . service description

Relationship: N/A.

Business rule:

- o Display provider avatar when showing service provider.

- o Display provider service details including:

    . provider name

    . provider service's rating

    . number of past bookings

    . unit price

    . service description

3.2.2.2. <Customer> Place Booking
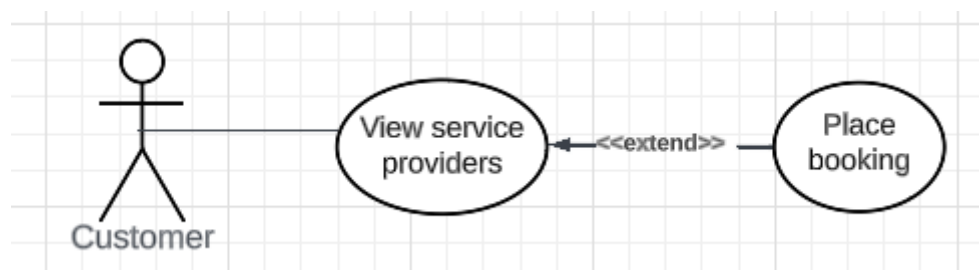
Figure: <Customer> Place Booking

USE CASE-05 SPECIFICATION			
Use-case No.	UC05	**Use-case Version**	1.1
Use-case Name	Place Booking		
Author	Nguyen Hoang Nam, Nguyen Ba Huy		
Date	17/06/2023	**Priority**	High

Actor:

Customer (primary), provider (secondary)

Summary:

A customer places a booking order after selecting a service provided by a provider.

Goal:

A customer can successfully send a booking placement request to the chosen provider

Triggers

A customer clicks the "Book" button on a service post.

Preconditions:

The customer must log in the system. The provider's service must be available.

Post Conditions:

- Success: The customer successfully booked the service.
- Failure: The booking request is declined (timeout for 3 minutes or declined by the provider) and the customer is prompted to pick the service of another provider.

Main Success Scenario:

Step	Actor Action	System Response
1	Customers click on one service from the service list in the homepage.	Show all providers of the service chosen.
2	Logged-in customer hits the "Book" button of one service provider to place the booking.	The system shows a count-down clock for 3 minutes waiting for the provider's response.

3.2.2.3. <Customer> View booking history
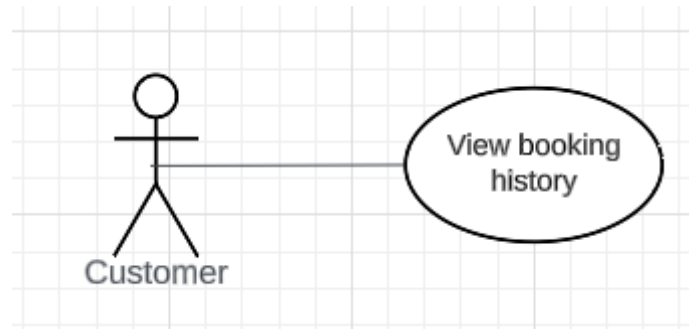
Figure: <Customer> View booking history

USE CASE-06 SPECIFICATION			
Use-case No.	UC06	**Use-case Version**	1.0
Use-case Name	View booking history		
Author	Nguyen Ba Huy		
Date	07/06/2023	**Priority**	Normal

Actor:

-        Customer

Summary:

-        This use case allows customer to view booking history

Goal:

-        Customer can view booking history.

Triggers:

-        Customer sends a view-booking-history command to the system.

Preconditions:

-        Customer must login into the system with the role Customer.

Post conditions:

-        Success: View the booking history successfully

Main Success Scenario:

Step	Actor	System Response
1	Customer clicks "Booking History" on navigation menu	System redirect to the booking history screen with a list of completed bookings of the customer with details about the booking: 1. Name of provider 2. Service and price 3. Booking Date

Alternative Scenario:

Step	Actor	System Response
1	Customer is not logged in or their account is invalid	Customer is redirected to the login/registration page.

Exceptions:

No	Actor	System Response
1	Customer does not have	The system displays a me

	booking history	ssage indicating that no bookings are available.
2	The customer's login	They will be prompte

session expires during the process	d to login again before accessing their booking history

Relationships: N/A

Business Rules:

- The system displays list of bookings based on customer's id

- A booking must have:

  1. Name of provider

  2. Service and price

  3. Booking Date

3.2.2.4. <Customer> View current activities
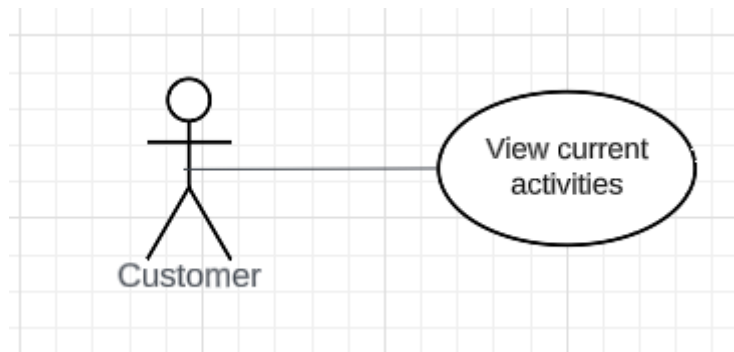
Figure: <Customer> View current activities

USE CASE-07 SPECIFICATION			
Use-case No.	UC07	**Use-case Version**	1.0
Use-case Name	View current activities		
Author	Nguyen Ba Huy		
Date	16/06/2023	**Priority**	Normal

Actor:

-          Customer

Summary:

-          This use case allows customer to view current activities

Goal:

-          Customer can view current activities.

Triggers:

-          Customer sends a view-current-activities command to the system.

Preconditions:

-          Customer must login into the system with the role Customer.

Post conditions:

-          Success: View the current activities successfully

Main Success Scenario:

Step	Actor	System Response
1	Customer clicks "Current Activities" on navigation menu	System redirect to the current activities screen with a list of current booking activities of the customer.
2	Customer clicks on an activity	System redirect to current activity detail screen. Users can view all information details about the current booking activity: 1. Name of provider

		2. Service and price
		3. Booking Date
		4. Status (confirmed by the provider or not)

Alternative Scenario:

Step	Actor	System Response
1	Customer is not logged in or their account is invalid	Customer is redirected to the login/registration page.

Exceptions:

No	Actor	System Response
1	Customer does es	The system dis

	not have current activities	plays a message indicating that no current activities are available.
2	The	They

	customer's login session expires during the process	will be prompted to login again before accessing their booking history

Relationships: N/A

Business Rules:

- Display list of current booking activities based on customer's id

- A current booking activity must have:

  1. Name of provider

  2. Service and price

  3. Booking Date

  4. Status (confirmed by the provider or not)

3.2.2.5. <Customer> Cancel booking
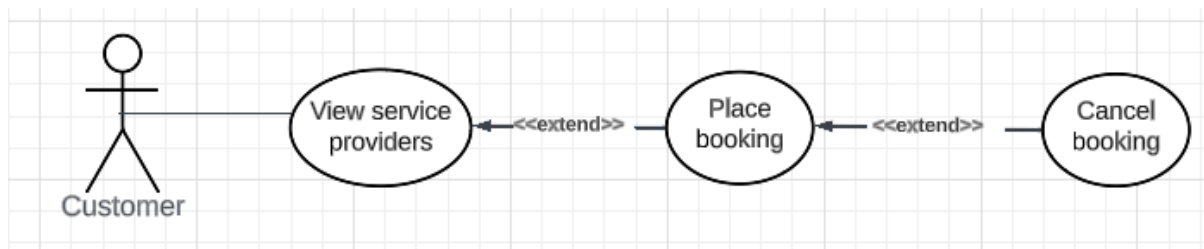
Figure: <Customer> Cancel booking

USE CASE - UC08			
Use Case No.	UC08	Use Case Version	1.0
Use Case Name	Cancel booking		
Author	Nguyen Ba Huy		
Date	16/06/2023	Priority	Normal

Actor:

\-   Customer

Summary:

\-   This use case allows customer to cancel service booking

Goal:

\-   Customer cancels booking successfully.

Triggers:

\-   Customer sends the cancel booking command to the system.

Preconditions:

\-   The user must login into the system with the role Customer.

Post conditions:

\-   Success: Booking was canceled successfully

Main Success Scenario:

Step	Actor Action	System Response
1	Customer clicks " Current Activity" on navigation menu	System redirect to the current activity screen with a list of current bookings of the customer.
2	Customer clicks "Cancel" on one current booking.	System will show message "Booking Canceled" and return to current activity screen

Relationships: View current activities

Business Rules:

·   Status of booking will be changed to "Canceled"

3.2.2.6. <Customer> Review booking
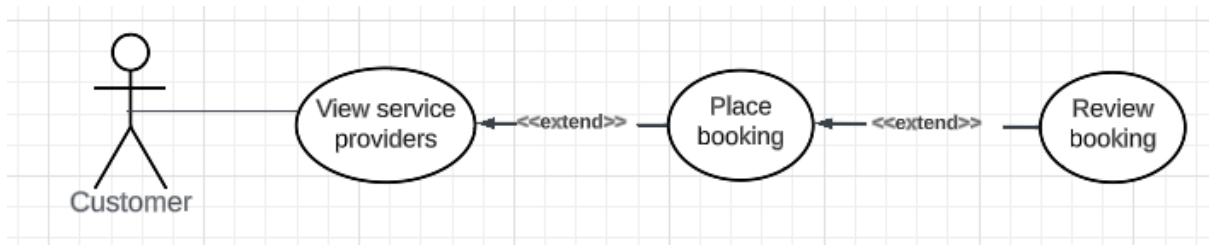
Figure: <Customer> Review booking

USE CASE - UC09			
Use Case No.	UC09	Use Case Version	1.0
Use Case Name	Review booking		
Author	Nguyen Ba Huy		
Date	16/06/2023	Priority	Normal

Actor:

-          Customer

Summary:

-          This use case allows customer to review and rate completed booking

Goal:

-          Customer can review and rate booking successfully

Triggers:

-          Customer clicks button "Review" of a booking article in the booking history screen.

Preconditions:

-          Customer must login into the system with the role Customer.

-          Customer must use services and status of booking is "Completed"

Post conditions:

-          Success: Review and rate booking successfully

Main Success Scenario:

Step	Actor Action	System Response
1	Customer clicks " Booking History" on the navigation menu.	System redirect to the booking history screen with a list of completed bookings of customers. [Exception 1]

No	Actor Action	System Response
2	Customer clicks the button "Review".	System redirect to review detail screen.
3	Customer gives a comment and rates the booking then clicks the button "Submit".	System will show a message "Review and Rate successfully" and return to the booking history screen.

Exception 1:

No	Actor Action	System Response
1	Customer does not have bookings	The system displays a message indicating that no bookings are available.

Relationships: N/A.

Business Rules:

- Customer must use services and status of booking is "Completed"

3.2.2.7. <Customer> Chat

Figure: <Customer> Chat

USE CASE - UC10			
Use Case No.	UC10	Use Case Version	1.0
Use Case Name	Chat		
Author	Nguyen Ba Huy		
Date	16/06/2023	Priority	Normal

Actor:

\-   Customer

Summary:

\-   This use case allows customer to chat with the service provider

Goal:

\-   Customer can chat with the service provider successfully

Triggers:

\-   Customer clicks the "Chat" button on one booking in the current activity screen.

Preconditions:

\-   Customer must login into the system with the role Customer.

\-   Customer must place booking and status of booking is "Order Confirmed By Provider" or "Completed"

Post conditions:

\-   Success: Chat with the service provider successfully

Main Success Scenario:

Step	Actor Action	System Response
1	Customer clicks " Current Activity" on navigation menu	System redirect to the current activity screen with a list of current bookings of the customer. (Exception 1)
2	Customer clicks "Chat" on one current booking.	System will open a chat box for the customer to chat with the provider of the current booking.

No	Actor Action	System Response
3	Customer types chat messages and hits "Send" to send messages to the provider.	System will send messages of the customer to the provider. At the same time, the system will show the messages sent by the provider in the chat box.

Exception 1:

No	Actor Action	System Response
1	Customer does not have current activities	The system displays a message indicating that no activities are available.

Relationships: View current activities

Business Rules:

- Customer must place booking and status of booking is "Processing" or "Completed"

3.2.2.8. <Customer> View chat history
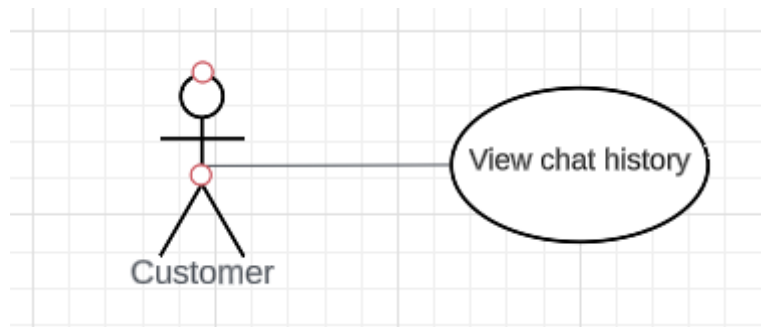
Figure: <Customer> View chat history

USE CASE-11 SPECIFICATION			
Use-case No.	UC11	**Use-case Version**	1.0
Use-case Name	View chat history		
Author	Nguyen Ba Huy		
Date	07/06/2023	**Priority**	Normal

Actor:

- Customer

Summary:

- This use case allows customer to view chat history

Goal:

- Customer can view chat history.

Triggers:

- Customer sends a view-chat-history command to the system.

Preconditions:

- Customer must login into the system with the role Customer.

Post conditions:

- Success: View chat history successfully

Main Success Scenario:

Step	Actor	System Response
1	Customer clicks	System redirect

		"Chat History" on navigation menu	to the chat history screen with a list of chats of the customer.
2		Customer	System will ope

	clicks on a chat	n a chat box for the customer to view messages in the past and continue to chat with the provider.

3	Customer types chat messages and hits "Send" to send messages to the provider.	System will send messages of the customer to the provider. At the same time, the system will sho

		w the mes sag es sent by the pro vide r in the chat box.

Alternative Scenario:

Ste p	Act or	Sy ste m Re sp on se
1	Cu sto me	Cu sto me

	r is not logged in or their account is invalid	r is redirected to the login/registration page.

Exceptions:

No	Actor	System Response

1	Customer does not have chats	The system displays a message indicating that no chats are available.

2	The customer's login session expires during the process	They will be prompted to login again before e accessing chat.

Relationships: N/A

Business Rules:

- Display list of chats based on customer's id

3.2.2.9. <Customer> Logout
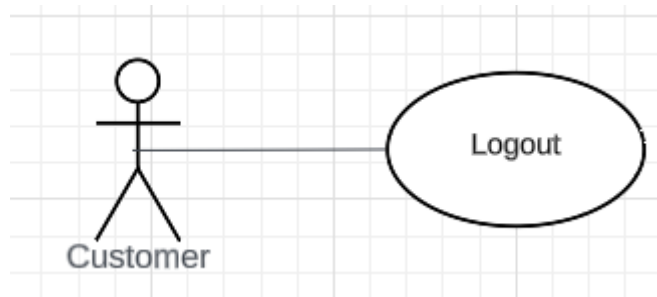
Figure: <Customer> Logout

USE CASE - UC12			
Use Case No.	**UC12**	**Use Case Version**	1.1
Use Case Name	Logout		
Author	Nguyen Ba Huy		
Date	16/06/2023	**Priority**	Normal

Actor:

- Customer

Summary:

- This use case allows the customer to logout to the system.

Goal:

- Customer can logout from the system

Trigger:

- Customer sends a logout command.

Precondition:

- Customer must log in before logging out.

Post Conditions:

- Success: Customer logout to the system successfully

Main Success Scenario:

Step	Actor Action	System Response
1	Customer clicks "Logout" on the navigation menu.	Return the user to the home screen.

Relationship: N/A.

Business rule:

- After logged out, user access the system as role "Customer"
- Return the user to the home screen after logged out.

3.2.2.10. <Customer> Update account information

Figure: <Customer> Update account information

USE CASE-13 SPECIFICATION			
Use-case No.	UC13	**Use-case Version**	1.0
Use-case Name	Update account information		
Author	Nguyen Ba Huy		
Date	17/06/2023	**Priority**	Normal

Actor:

- Customer

Summary:

- This use case allows customer to update their account information

Goal:

- Customer can update their account information.

Triggers:

- Customer sends an update account information command to the system.

Preconditions:

- Customer must login into the system with the role Customer.

Post conditions:

- Success: update the account information successfully

Main Success Scenario:

Step	Actor	System Response
1	Customer clicks "Account" on navigation menu	System redirect to the account screen with account information of the customer.
2	Customer clicks on "Update" button	System show a form pop-up for the customer to fill in details: 1. Name of customer

		2. Email
		3. Phone
3	Customer fills in details (name, email, phone) and hits the "Submit" button.	System will show the message "Update account information successfully" and return to the account screen.

Alternative Scenario:

Step	Actor	System Response
1	Customer is not logged in	Customer is redirected to the

	or the ir acc ou nt is inv ali d	log in/ reg ist rat ion pa ge.

Exceptions:

No	Act or	Sy ste m Re sp on se
1	Th e cu sto me r's	Th ey wil l be pr

	login session expires during the process	ompted to login again before accessing their booking history

Relationships: N/A

Business Rules:

- Display customer's information based on customer's id

- Customer's account information must have:

    1. Name of customer

2.   Email
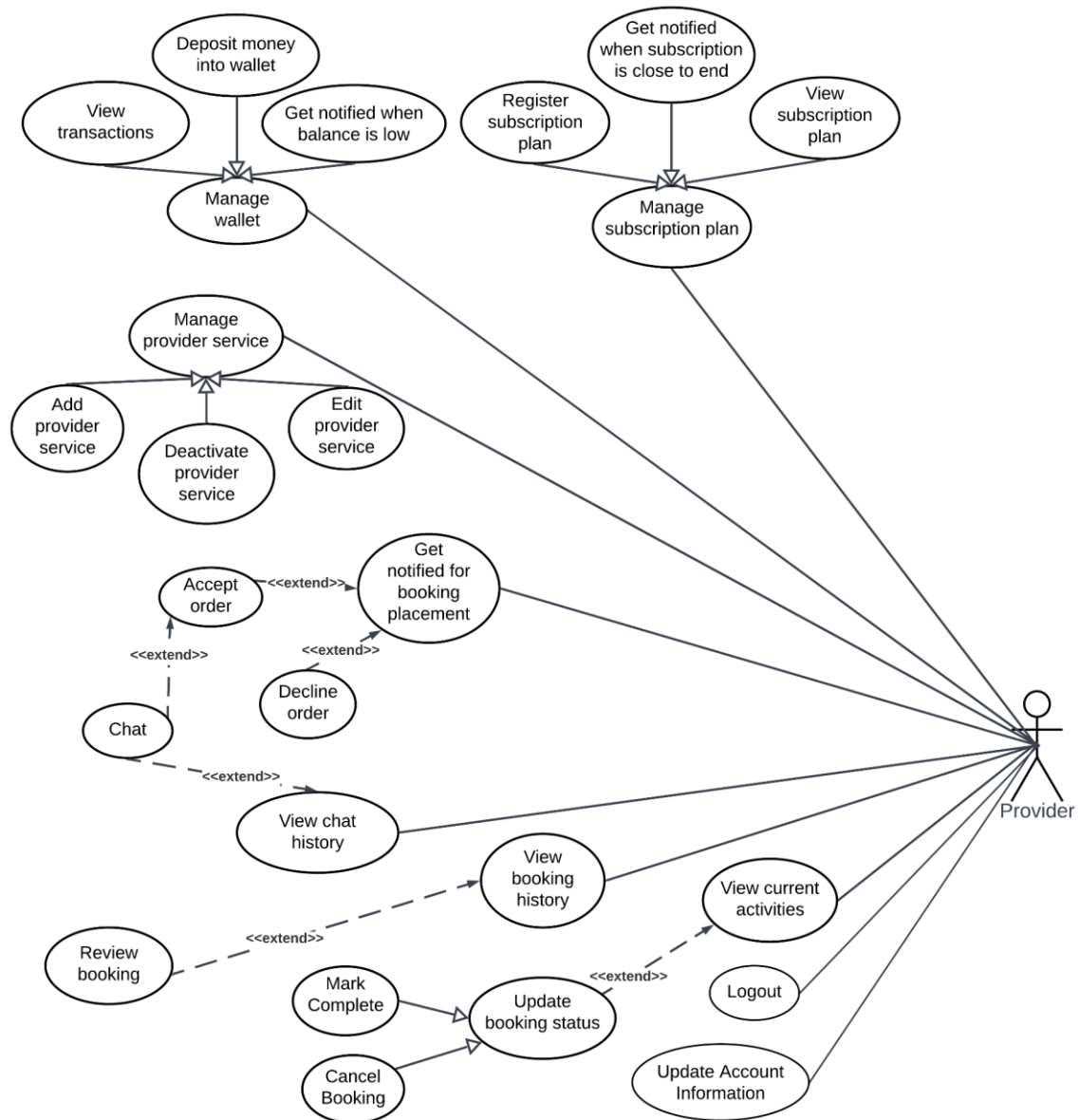
3.   Phone

3.2.3. <Provider> Overview Use Case

Figure: <Provider> Overview use case

3.2.3.1. <Provider> View current activities

Figure: <Provider> View current activities

USE CASE – UC_14			
Use Case No.	**UC_14**	**Use Case Version**	2.0
Use Case Name	View current activities		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to view his on-going activities like pending bookings or working bookings.

Goal:

- Provider successfully views his on-going activities

Triggers:

- Provider clicks " Current Activity" tab on nav bar

Preconditions:

- User must login to the system as provider
- Provider account is active in the system

Post conditions:

- Success:

    Provider successfully views his list of current bookings categorized into different groups (Pending/Working)

- Fail: Show error message of booking retrieval failure

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Current Activities" tab option	Display a full list of current activities categorized into groups. The booking details include: - Customer name

		- Provider service name and ID
		- Building name
		- Apartment
		- Unit price
		- Status
3	Provider can click buttons available in on-going bookings to update the status of those bookings and carry other tasks	Update the status of current bookings and walk customer to other tasks

Exceptions:

No	Cause	System Response
1	Page of current activities is attempted to be retrieved even when the provider is not logged in (no authentication)	System shows a message with authentication and walks the user to the login page.

Relationships: N/A

Business Rules:

● Provider can view full details of current bookings.

● Categories include:

   o Pending bookings: waiting for provider's response (decline or accept).

   o Working bookings: the task that the provider is completing.

3.2.3.2. <Provider> Get notified for booking placement

Figure: <Provider> Get notified for booking placement

USE CASE – UC_15			
Use Case No.	**UC_15**	Use Case Version	2.0
Use Case Name	Get notified for booking placement		
Author	Nguyen Hoang Nam		
Date	17/06/2023	Priority	High

Actor:

- Provider (primary), customer(secondary), system (secondary)

Summary:

- This use case allows a provider to get notified in case some customer places a booking of his service.

Goal:

- Provider successfully receives a notification for his new booking placement.

Triggers:

- Customer clicks on the button "Book" on the post of the provider service.

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

- The customer books a booking of the provider's available services.

Post conditions:

- Success:

    Provider successfully views his list of current categories categorized into different groups (Pending/Working)

- Fail: Show error message of booking retrieval failure

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider

2	Clicks on the "Current Activities" tab option	Display full list of current activities categorized into groups
3	Provider can click buttons available in on-going bookings to update the status of those bookings and carry other tasks	Update the status of current bookings and walk customer to other tasks

Exceptions:

No	Cause	System Response
1	Page of current activities is attempted to be retrieved even when the provider is not logged in (no authentication)	System shows a message with authentication and walks the user to the login page.

Relationships: Place booking by a customer

Business Rules:

● Provider can view full details of current bookings.

● Categories include:

    o Pending bookings: waiting for provider's response (decline or accept).

    o Working bookings: the task that the provider is completing.
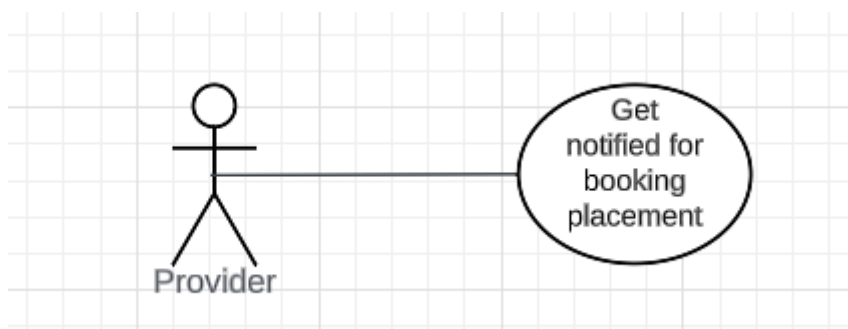
3.2.3.3. <Provider> Accept booking request

Figure: <Provider> Accept booking request

USE CASE – UC_16			
Use Case No.	**UC_16**	**Use Case Version**	2.0
Use Case Name	Accept booking request		

Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), customer(secondary), system (secondary)

Summary:

- This use case allows a provider to accept a booking request after being notified of its placement by a customer.

Goal:

- Provider successfully accepts the booking from a customer.

Triggers:

- Provider clicks on the "Accept" button on the notification pop-up message upon the booking placement of a customer.

Preconditions:

- User must login to the system as provider
- Provider account is active in the system
- A booking of the provider's available services has been placed.

Post conditions:

- Success:

    Provider successfully accepts his booking

- Fail: Show error message of failure in accepting the booking.

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks notification icon on his nav bar	Display list of latest notifications including the booking placement of his service.
3	Provider clicks the specific notification of the service being booked	The system displays the details of the booking, including: - Customer Name

			- Provider Service name - Building - Apartment - Unit price - Date and time - with 2 options "Accept" and "Decline".
4	Provider clicks on the button "Accept" within the time remaining		- The system updates the status of the booking to ACCEPTED and moves this booking to the "Current Activities" page of both the provider and the customer. - The system switches the availability status of the provider off until the completion of such a task. This hides all the services of this provider from the board of services visible to customers. - The system automatically generates a booking fee account on the provider's wallet, which results in a subtraction to his balance. The transaction is recorded in the database, and can be retrieved in the provider's transaction history. - From the customer's view, the clock counting down will stop and a successful

| | | acceptance message pops up to notify the customer of the booking acceptance. |
| | | - The current position of the provider is displayed in the corresponding booking in the "Current Activity page". The customer is allowed to book other services upon this. |

Alternative Scenario

Step	Actor	System Response
1	The provider accesses his email account and checks the notification for a new booking placement.	
2	The provider clicks on the link to pending booking for further action	Ask the provider to log in upon the access of the application Upon successful authentication, bring the provider to the waiting booking on the "Current Activities" page.
3	Similar to step 3 of the main scenario	
4	Similar to step 3 of the main scenario	

Exceptions:

No	Cause	System Response

1	Page of current activities is attempted to be retrieved even when the provider is not logged in (no authentication)	System shows a message with authentication and walks the user to the login page.
2	System fails to update the working bookings	System shows a message with an internal exception and the booking is marked DECLINED automatically. The system will then ask the customer to rebook another service.
3	The provider does not respond to the booking request or hit the "Accept" button in the last seconds and the network connection fails to catch up with the countdown clock on the customer side	The system will notice the customer of the decline of his/her booking. The status of such a booking will be set DECLINED and the provider cannot accept or decline it anymore. A recommendation of similar service by other providers is displayed on the customer's screen.

Relationships: Get notified of a new booking

Business Rules:

● The countdown time for a booking request is 3 minutes, which equates to the allowance time for the customer to accept a booking, disregarding latencies or network glitches.

● If the provider does not respond to a booking within countdown time, the system automatically marks such booking DECLINED and moves the booking to the booking history. No booking fee is charged for this circumstance.

● The current position of the provider is updated constantly on the customer's screen upon booking acceptance to give some hints of where the provider is.

● The provider should strive to arrive at the destination within 15 minutes and complete his task satisfactorily.

3.2.3.4. <Provider> Decline booking request
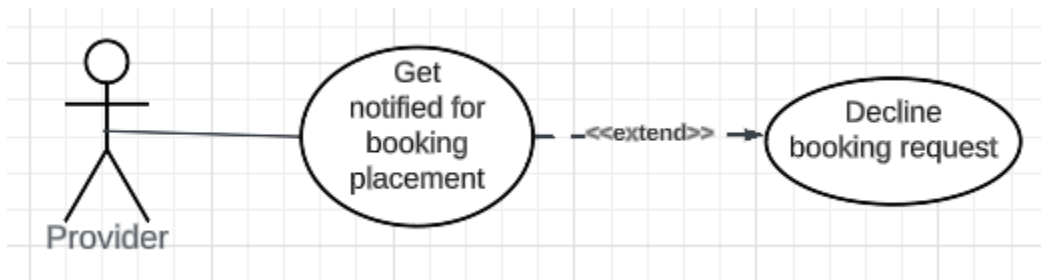
Figure: <Provider> Decline booking request

USE CASE – UC_17			
Use Case No.	**UC_17**	Use Case Version	2.0
Use Case Name	Decline booking request		
Author	Nguyen Hoang Nam		
Date	17/06/2023	Priority	High

Actor:

- Provider (primary), customer(secondary), system (secondary)

Summary:

- This use case allows a provider to decline a booking request after being notified of its placement by a customer.

Goal:

- Provider successfully declines the booking from a customer.

Triggers:

- Provider clicks on the "Decline" button on the notification pop-up message upon the booking placement of a customer.

Preconditions:

- User must login to the system as provider
- Provider account is active in the system
- A booking of the provider's available services has been placed.

Post conditions:

- Success:

        Provider successfully declines a booking

- Fail: Show error message of failure in declining the booking.

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks notification icon on his nav bar	Display list of latest notifications including the booking placement of his service.
3	Provider clicks the specific notification of the service being booked	The system displays the details of the booking, including: - Customer Name - Provider Service name - Building, apartment - Unit price - Date and time - With 2 options "Accept" and "Decline".
4	Provider clicks on the button "Decline" within the time remaining	A confirmatory popup message appears, asking "Are you sure to decline this booking?" with 2 options: "Yes" and "No".
5	Provider clicks on the button "Yes"	- The system updates the status of the booking to DECLINED and moves this booking to the "Current Activities" page of both the provider and the customer. - A message of decline is displayed on the provider's screen and he is brought back to the home page.

| | | - The system will switch the availability status of the system off for 15 minutes; therefore, he will not be visible on customers' screens for such a time interval. |
| | | - The countdown clock is collapsed and an apology message is displayed on the customer's screen, signifying the decline of the booking. A recommended alternative list is displayed for the same service by other providers. |

Alternative Scenario

Step	Actor	System Response
1	The provider accesses his email account and checks the notification for a new booking placement.	
2	The provider clicks on the link to pending booking for further action	Ask the provider to log in upon the access of the application Upon successful authentication, bring the provider to the waiting booking on the "Current Activities" page, with details including the Customer Name, Provider Service name, building, apartment, unit price, date and time with 2 options "Accept" and "Decline".

3	Similar to step 4 of the main scenario
4	Similar to step 5 of main scenario

Exceptions:

No	Cause	System Response
1	Page of current activities is attempted to be retrieved even when the provider is not logged in (no authentication)	System shows a message with authentication and walks the user to the login page.
2	System fails to update the working bookings	System shows a message with an internal exception and the booking is marked DECLINED automatically. The system will then ask the customer to rebook another service.

Relationships: Get notified of a new booking

Business Rules:

- The countdown time for a booking request is 3 minutes, which equates to the allowance time for the customer to accept a booking, disregarding latencies or network glitches.

- If the provider does not respond to a booking within countdown time, the system automatically marks such booking DECLINED and moves the booking to the booking history. No booking fee is charged for this circumstance.

- The decline of the provider is a negative event, which results in a 15-minute unavailability status on the provider's profile and the prompt for other options of the same service for the declined customer.

- The DECLINED status of the booking does not allow any reviews of ratings from either side. This item is still visible in the booking history of both.

3.2.3.5. <Provider> Mark Complete a Booking (Update booking status)
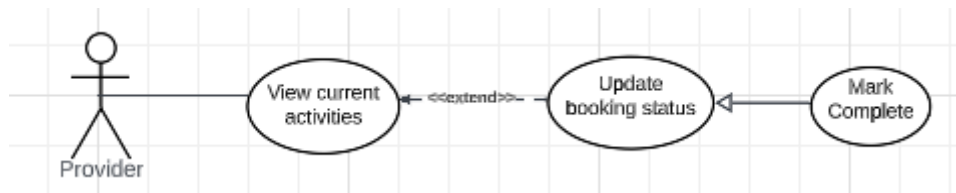
Figure: <Provider> Mark Complete a Booking (Update booking status)

USE CASE – UC_18			
Use Case No.	**UC_18**	Use Case Version	2.0
Use Case Name	Mark Complete a Booking (Update booking status)		
Author	Nguyen Hoang Nam		
Date	17/06/2023	Priority	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to mark the completion of bookings in their status on the 'Current Activity' page of a provider.

Goal:

- Provider successfully marks complete the status of a booking upon the finish of his work.

Triggers:

- Provider completes his practical tasks.
- Provider hits the "Complete" button on the working bookings

Preconditions:

- User must login to the system as provider
- Providers must have on-going working bookings.
- Provider should complete his task in real life.

Post conditions:

- Success:

    Provider successfully updates the completion status of his working bookings

- **Fail:** Show error message of failure of updating booking status

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Current Activities" tab option	Display full list of current activities
2	Navigate to working bookings, provide optional information like the total amount (for statistics).	Save additional information for further sum-up statistics at the end of each month
3	Hit the button "Complete" in the lower right corner	Update the completion status of the booking and switch the availability status of the provider back on so that he will appear on the board of customers

Exceptions:

No	Cause	System Response
1	Provider gives the wrong format of total amount, which should be long.	System shows a message with incorrect format and prompts the provider to retry
2	User tries to retrieve the working bookings without logging in	System shows a message with authentication and walks the user to the login page.
3	System fails to update the working bookings	System shows a message with an internal exception and asks the provider to come back later.

Relationships: View current activities

Business Rules:

● Logged in provider can update the status of his working bookings upon completion in real life. During the working status of the provider, his availability

will be automatically switched off, hiding him away from the search view of customers.

● Updating the completion status of the working bookings activates the provider's availability. The provider can give additional information, like the total amount earned for further statistics at the end of the month for revenue sum-up.

3.2.3.6. <Provider> Cancel a Booking (Update booking status)
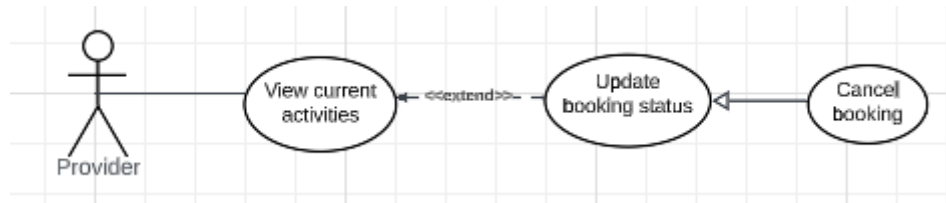
Figure: <Provider> Cancel a Booking (Update booking status)

USE CASE – UC_19			
Use Case No.	**UC_19**	**Use Case Version**	2.0
Use Case Name	Cancel a Booking (Update booking status)		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to cancel a booking on the 'Current Activity' page of a provider.

Goal:

- Provider successfully cancels a booking after accepting it.

Triggers:

- Provider hits the "Cancel" button on the working bookings in his current activities

Preconditions:

- User must login to the system as provider

- Providers must have on-going working bookings.

- Provider may fail to assume his job.

Post conditions:

- Success:

Provider successfully cancels a booking.

- Fail: Show error message of failure of updating booking status.

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Current Activities" tab option	Display full list of current activities
2	Navigate to working bookings, hit the "Cancel" button on the current activities that he fails to complete.	Prompt a pop-up warning that the cancellation will mark a red flag in the provider's credibility. A popup window for confirmation appears with the "Confirm" and "Discard" option.
3	Hit the button "Confirm" button	- Update the status of such booking, bringing it to the booking history of both and freeze the unavailability status of the provider for 1 hour. - The system increments the cancellation red flag on the provider's profile as the basis for account suspension. - The canceled customer is notified with the cancellation of his/her booking and is encouraged to pick another provider - The system allows customers and providers to

		leave review and rating on the canceled bookings.

Exceptions:

No	Cause	System Response
1	User tries to retrieve the working bookings without logging in	System shows a message with authentication and walks the user to the login page.
2	System fails to update the working bookings	System shows a message with an internal exception and asks the provider to come back later.

Relationships: View current activities

Business Rules:

● Logged in provider can cancel a booking if he fails to fulfill it upon his acceptance. This action increments his cancellation red flag count, the threshold of which is 5 times per month. Passing this cutoff point results in account suspension.

● The cancellation of a booking is a negative event, which disables the provider's availability for 1 hour and prompts the customers for alternative option repicks.

● The provider and customer of such booking is allowed to leave reviews and ratings for their reasons and complaints. This counts to the overall rating of the provider himself and his provider service.

3.2.3.7. <Provider> Chat
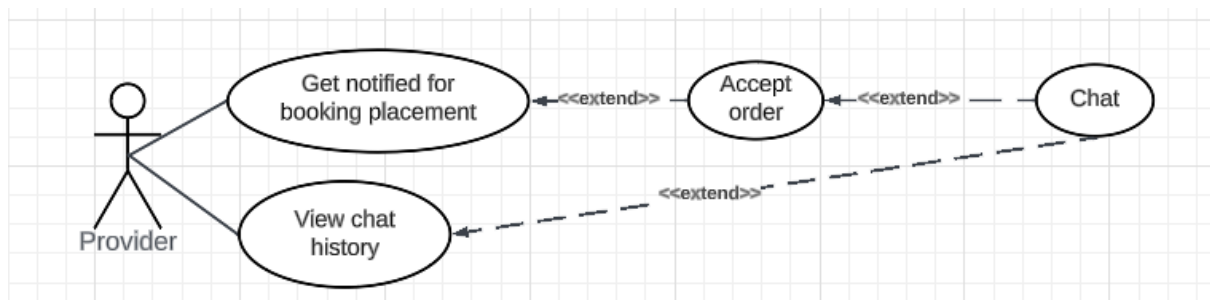
Figure: <Provider> Chat

USE CASE - UC20			
Use Case No.	**UC20**	**Use Case Version**	**1.0**
Use Case Name	**Chat**		
Author	**Nguyen Ba Huy**		
Date	**16/06/2023**	**Priority**	**Normal**

Actor:

\-   Provider

Summary:

\-   This use case allows the service provider to chat with the customers upon the booking acceptance.

Goal:

\-   The provider can proceed to chat with the customers upon accepting the booking placed by them.

Triggers:

\-   Provider clicks the "Chat" button on one booking in the current activity screen or clicks on the booking to chat on the chat history page.

Preconditions:

\-   Provider must login into the system with the role Provider.

\-   Provider must accept the booking request and the status of booking is ACCEPTED.

Post conditions:

\-   Success: The provider successfully chats with the customer.

Main Success Scenario:

Step	Actor Action	System Response
1	Provider clicks " Current Activity" on navigation menu	System redirect to the current activity screen with a list of current bookings. (Exception 1)

No	Actor Action	System Response
2	Provider clicks "Chat" on one current booking.	System will open a chat box for the provider to chat with the customer of the current booking.
3	Provider types chat messages and hits "Send" to send messages to the customer.	System will send messages of the provider to the customer.. At the same time, the system will show the messages sent by the customer in the chat box.

Exception 1:

No	Actor Action	System Response
1	Provider does not have current activities.	The system displays a message indicating that no activities are available.

Relationships: View current activities

Business Rules:

- The provider can chat with customers, the booking of whom has been accepted by the provider. To start the chat, the status of booking must be ACCEPTED.

- The messages sent are not subject to modification neither by the provider, the customer nor the admin. The background worker of the system will periodically remove messages older than 3 months.

- The customer and provider can retrieve their recent chats in the chat history. The content of the conversation should revolve around the booking being completed and no private contact is expected to be disclosed.

3.2.3.8. <Provider> View chat history
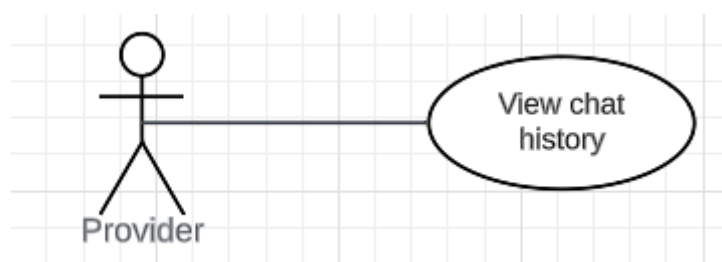
Figure: <Provider> View chat history

USE CASE-21 SPECIFICATION

Use-case No.	UC21	**Use-case Version**		1.0
Use-case Name	View chat history			
Author	Nguyen Ba Huy			
Date	17/06/2023	**Priority**	Normal	

Actor:

-         Provider

Summary:

-         This use case allows provider to view chat history

Goal:

-         Provider can view chat history.

Triggers:

-         Provider sends a view-chat-history command to the system.

Preconditions:

-         Provider must login into the system with the role Provider.

Post conditions:

-         Success: View chat history successfully

Main Success Scenario:

Step	Actor	System Response
1	Provider clicks	System redirect

	"Chat History" on navigation menu	to the chat history screen with a list of chats of the provider.
2	Provider	System will ope

	clicks on a chat	n a chat box for the provider to view messages in the past and continue to chat with the customer.

3	Provider types chat messages and hits "Send" to send messages to the customer.	System will send messages of the provider to the customer. At the same time, the system will sho

		w the mes sag es sent by the cust om er in the chat box.

Alternative Scenario:

Ste p	Act or	Sy ste m Re sp on se
1	Pr ovi	Pr ovi

	der is not logged in or their account is invalid	der is redirected to the log in/ registration page.

Exceptions:

No	Actor	System Resp

		onse
1	Provider does not have chats	The system displays a message indicating that no chats are avail

		abl e.
2	The provide r's log in ses sio n ex pir es du rin g the pr oc ess	Th ey wil l be pr om pte d to log in ag ain bef or e acc ess ing ch at.

Relationships: N/A

Business Rules:

- Display list of chats based on provider's id

3.2.3.9. <Provider> View Booking History

Figure: <Provider> View Booking History

USE CASE – UC_22			
Use Case No.	**UC_22**	**Use Case Version**	2.0
Use Case Name	View booking history		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to view his booking history, including completed, declined, and canceled bookings.

Goal:

- Provider successfully views his booking history categorized by status types and sorted by time.

Triggers:

- Provider clicks "Booking History" tab on nav bar

Preconditions:

- User must login to the system as provider
- Provider account is active in the system

Post conditions:

- Success:

  Provider successfully views his list of history bookings categorized by status: declined, completed, and canceled and sorted by time.

- Fail: Show error message of booking history retrieval failure

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Booking History" tab option	Display full list of history bookings categorized by status and sorted by time
3	Provider can filter bookings by date and by status: completed/canceled/declined button	Returns list of bookings fitting the filters, including information: - Customer name - Provider service name and ID - Building name, apartment - Unit price and total - Rating by provider and customer - Review by customer and provider - Status

Exceptions:

No	Cause	System Response
1	Page of current activities is attempted to be retrieved even when the provider is not logged in (no authentication)	System shows a message with authentication and walks the user to the login page.

Relationships: N/A

Business Rules:

- Provider can view full details of history bookings.

- Booking status:

  - Canceled: bookings canceled (not able to complete) by provider upon his acceptance.

  - Declined: booking placements refused by provider

  - Completed: bookings completed by provider.

- Provider cannot delete or create new entries in booking history nor the current activity. System's background worker will periodically remove booking history older than 3 months.

- Providers can only update the total amount and leave rating, review (for statistics purpose only) in the booking history.

3.2.3.10. <Provider> Review Booking

Figure: <Provider> Review Booking

USE CASE – UC_23			
Use Case No.	**UC_23**	Use Case Version	2.0
Use Case Name	Review Booking		
Author	Nguyen Hoang Nam		
Date	17/06/2023	Priority	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to review and rate the customer of a completed booking in the booking history.

Goal:

- Provider successfully reviews and rates the customer of a completed booking in booking history.

Triggers:

- Provider clicks "Review" button on a completed booking in booking history

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

- The booking to be reviewed must be completed

Post conditions:

- Success:

    Provider successfully adds review and rating to a completed booking, which subsequently updates the overall rating of the corresponding customer.

- Fail: Show error message of failure to add review/rating

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Booking History" tab option	Display full list of history bookings categorized by status and sorted by time
3	Provider selects a booking to review and hits "Review" button	The system displays the full details of selected booking, including: - Customer name - Provider service name - Building name, apartment - Unit price, total - Date - Input field for review along with number of stars for rating.

4	Provider selects a number of stars (from 1 to 5) for rating and enters review in the text box or attaches an image if he wishes to. Provider then hits "Save" button on the booking being reviewed.	- System automatically saves the rating and review of the booking into the system. These items will be displayed in the details of a provider service. - The image would be forwarded and saved in Firebase storage, the result of which, a link to the image, is then saved in the system's database. - The system will recalculate the average rating of the corresponding customer by averaging the ratings of bookings by such customer. - A success message is sent to the provider and he will be back on the page for booking details being reviewed.

Exceptions:

No	Cause	System Response
1	Provider inputs a review longer than max length of the box	System shows a message warning the excessive length of the review

Relationships: View Booking History

Business Rules:

- The provider can only rate and review a booking that has been completed (status: COMPLETED). Otherwise, the option for review is disabled.

- The rating scale includes integers from 1 to 5

- The provider can review without rating and vice versa or can do both.

- The max length for review is 1500 characters

- The image of relevant content can be uploaded from the local machine of the provider

- The reviews will be displayed in the response portion of the review section of the corresponding provider service. The rating serves the purpose of calculating the overall rating of a customer.

3.2.3.11. <Provider> Add Provider Service

Figure: <Provider> Add Provider Service

USE CASE – UC_24			
Use Case No.	**UC_24**	**Use Case Version**	2.0
Use Case Name	Add Provider Service		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:
- Provider (primary), system (secondary)

Summary:
- This use case allows a provider to add a provider service of one of available services in the system.

Goal:

- Provider successfully creates a new provider service and could be accessed in his service main page from both provider and customer view.

Triggers:

- Provider clicks "Create Service" in the provider service form after filling required fields.

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

- The service and the service category must be defined by the system and remain available.

Post conditions:

- Success:

    Provider successfully creates a provider service.

- Fail: Show error message of failure to create a new provider service

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Provider Service" tab option	Display full list of provider services of the provider, categorized by service type.
3	Provider clicks "Add Provider Service" button	A provider service form is displayed with fields to fill in: - serviceName (required and must be an option in a dropdown list) - unitPrice - description. Other read-only fields include: - lowerUnitPrice

		- upperUnitPrice,
		- Unit
		- BookingFee (commission).
4	Provider fills in all the required fields and hit "Create" button at the bottom	- The system automatically creates a new record in the database, with bookingNo being set to 0, rating: 0, visible: true and available: true.
		- A success message is prompted to the provider screen, which then brings him back to the service main page.

Exceptions:

No	Cause	System Response
1	Provider inputs an out-of-range unit price	System shows a message with a warning that the unit price is out-of-bound. Attempting to create a new provider service results in failure of such action.
2	Provider inputs a description longer than max length of text box	System shows a message with a warning that the description is surpassing the length limit. Attempting to create a new provider service results in failure of such action.
3	Provider leaves the service type blank	System shows a message with a warning that the service is unselected. Attempting to create a

		new provider service results in failure of such action.

Relationships: N/A

Business Rules:

● Providers can create a provider service of one of the defined services by the system admin.

● The unit and booking commission fee are fixed for all provider services of a service type. The unit price must be within the range constrained by admin.

● Description max length: 1500 characters.

● The bookingNo is updated by the total number of completed bookings in the past. The rating is the average rating of all ratings by customers of bookings in the past. There must be a background worker to update such fields.

● The default value for availability and visibility is true, which means that newly created provider service is visible to all customers and guests searching for services.

3.2.3.12. <Provider> Edit Provider Service

Figure: <Provider> Edit Provider Service

USE CASE – UC_25			
Use Case No.	**UC_25**	Use Case Version	2.0
Use Case Name	Edit Provider Service		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High
Actor:			

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to edit an existing provider in his service main page.

Goal:

- Provider successfully updates an existing provider service and the modified provider service could be accessed in his service main page from both provider and customer view.

Triggers:

- Provider clicks on the "Edit Service" button on a specific provider service on the service main page.

Preconditions:

- User must login to the system as provider
- Provider account is active in the system
- The provider service must be active and existent.

Post conditions:

- Success:

  Provider successfully updates a provider service.

- Fail: Show error message of failure to update the provider service

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Provider Service" tab option	Display full list of provider services of the provider, categorized by service type.
3	Provider clicks "Edit" button on a provider service	A provider service form is displayed with fill-in fields, which have been injected with the values of current provider service:

		- Service name (required and must be an option in a dropdown list) - Unit price - Description - Checkbox for visible. Other read-only fields include: - Lower and upper bound for unit price - Unit - Booking commission fee.
4	Provider fills in the fields that need updating and hit "Update" button at the bottom	- The system automatically updates the existing provider service in the database. - A success message is prompted to the provider screen, which then brings him back to the service main page.

Exceptions:

No	Cause	System Response
1	Provider inputs an out-of-range unit price	System shows a message with a warning that the unit price is out-of-bound. Attempting to create a new provider service results in failure of such action.
2	Provider inputs a description longer than max length of text box	System shows a message with a warning that the description is surpassing the length limit. Attempting to create a new

		provider service results in failure of such action.
3	Provider leaves the service type blank	System shows a message with a warning that the service is unselected. Attempting to create a new provider service results in failure of such action.

Relationships: N/A

Business Rules:

● Providers can update the service type of an existing provider service to one of the defined services by the system admin.

● The unit and booking commission fee are fixed for all provider services of a service type. The unit price must be within the range constrained by admin.

● Description max length: 1500 characters.

● The provider can switch the checkbox visible on and off. This flags the visibility of such provider service on the customer view. In some cases, a specific provider service is not available sometimes, this field can be helpful.

● The default value for availability is true, which means that newly created provider service is visible to all customers and guests searching for services.

3.2.3.13. <Provider> Deactivate Service (Soft Delete)

Figure: <Provider> Deactivate Service (Soft Delete)

USE CASE – UC_26			
Use Case No.	**UC_26**	**Use Case Version**	2.0
Use Case Name	Deactivate Provider Service		

Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to deactivate (soft delete) a provider service of his..

Goal:

- Provider successfully deactivates a provider service, which removes such the item from all actors' views, but the item still remains inactive in the system.

Triggers:

- Provider clicks the "Delete" button on a specific provider service on the service main page.

Preconditions:

- User must login to the system as provider
- Provider account is active in the system
- The provider service to be deleted must be existent.

Post conditions:

- Success:

    Provider successfully deletes a provider service.

- Fail: Show error message of failure to delete the new provider service

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Provider Service" tab option	Display full list of provider services of the provider, categorized by service type.

3	Provider selected a provider service to delete and clicks the "Delete" button on a provider service	The system will prompt a popup dialog asking if the provider is sure to delete this provider service with 2 options: "Yes" and "No". There is no way to reverse this action.
4	Provider clicks the "Yes" button on the popup dialog.	- The system switches the active flag of the provider service to false, hiding it from all system queries. - A finish message is displayed on the provider's screen and he will be brought back to the service main page.

Exceptions:

No	Cause	System Response
1		

Relationships: N/A

Business Rules:

● Providers can delete a provider service from his view and that of the customers. However, the deleted item still remains inactive in the system. This avoids the deletion of foreign keys in subsequent tables.

● The action of deletion can only be reversed by a request for re-activation to the database admin, but this circumstance is unlikely.

● The deleted provider services would not be seen by the provider, the customers and the admin.

● Any child objects of the deleted provider service will remain in the system, including the bookings and the images.

3.2.3.14. <Provider> View Transactions

Figure: <Provider> View Transactions

USE CASE – UC_27			
Use Case No.	**UC_27**	**Use Case Version**	2.0
Use Case Name	View Transactions		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to view the transactions conducted on his wallet over the past 3 months.

Goal:

- Provider successfully views the transactions categorized into types over the past 3 months.

Triggers:

- The provider clicks on the "View Transaction History" button on the wallet main page.

Preconditions:

- User must login to the system as provider
- Provider account is active in the system

Post conditions:

- Success:

    Provider successfully views a list of transactions that he has executed over the past 3 months sorted temporally.

- Fail: Show error message of failure to retrieve transaction history.

Main Success Scenario:

Step	Actor Action	System Response
1	Provider navigates to the nav bar and clicks on the "Wallet" tab	-The system displays the wallet main page, containing specific pages for "Deposit" and "View Transaction History" and the current balance that the provider possesses.
2	Provider clicks on the "View Transaction History" button.	The system displays a list of transactions that the provider has conducted, the order of which is chronologically managed. Each transaction contains the following fields: - category (deposit/booking fee/subscription fee) - Amount of charge - Date.
3	The provider can additionally perform some filtration or sorting (by date/by category) according to his desire.	The system sorts/filters/ the results based on the requirements of the provider.

Exceptions:

No	Cause	System Response
1	Internal error in sorting or filtering results.	The system will prompt a message notifying the trouble and ask the provider to try again later. The provider can email system admin about this contingency.

Relationships: N/A

Business Rules:

- Providers can view the transaction history that his wallet has been associated with over the past 3 months.

- The history is not subject to modification neither by the provider nor the admin. The background worker of the system would periodically remove the past history older than 3 months old.

- The transactions serve the purpose of managing the cash flow over a time period, and aids in generating the statistical figures for income summary.

3.2.3.15. <Provider> Deposit money into wallet

Figure: <Provider> Deposit money into wallet

USE CASE – UC_28			
Use Case No.	**UC_28**	**Use Case Version**	2.0
Use Case Name	Deposit Money into Wallet		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to deposit money into his wallet

Goal:

- Provider successfully deposits money into his wallet account.

Triggers:

- Provider clicks the "Deposit" button on the wallet management page.

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

Post conditions:

- Success:

   Provider successfully deposits money into his current balance in the
   wallet.

- Fail: Show error message of failure to make a deposit.

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Wallet" tab on the nav bar	Display button options within the wallet management page, including: deposition, view transaction history and current balance.
3	Provider clicks on the "Deposition" button.	The system displays a wizard form including: - Amount to deposit - Default method, which can be chosen at the discretions of the provider.
4	Provider fills in the required fields and hits "Deposit".	- The system saves the deposition information, and calls an API to the third-party transaction apps (like Momo or VNPay) to make the transaction. The system updates the new balance of the account. - The system also creates a new transaction entry in the Transaction table, which is

| | | retrieved later in the transaction history. |

Exceptions:

No	Cause	System Response
1	Internal error calling third-party API	The system displays a message or error and requires the provider to reattempt later.
2	The provider enters an amount greater than balance in his bank account	The system replies with a message warning that the amount deposited surpasses the balance in his bank account and the system fails to make such a transaction.

Relationships: N/A

Business Rules:

● Providers can deposit an amount of money into his wallet for booking commission fees and subscription fee payment.

● The system relies on third-party transaction apps like Momo or VNPay for real money transactions on the provider's account.

● The amount of money deposited to the wallet is equal to the exact money subtracted in the provider's bank account.

● The history of deposition is saved in the transaction history for later retrieval.

3.2.3.16. <Provider> Get Notified When Balance is Low
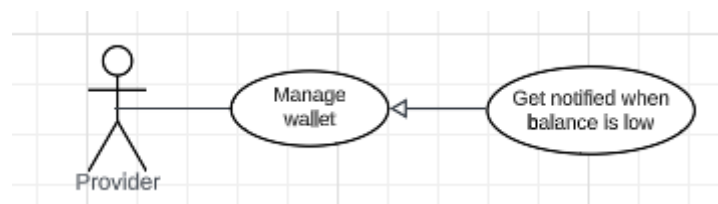
Figure: <Provider> Get Notified When Balance is Low

USE CASE – UC_29			
Use Case No.	**UC_29**	**Use Case Version**	2.0
Use Case Name	Get Notified When Balance is Low		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to get notified when his balance in the wallet is low and might not be sufficient for future charges.

Goal:

- Provider successfully gets notified when his wallet balance drops to the defined threshold.

Triggers:

- The current balance drops to or lower than the defined threshold.

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

- The current balance is lower or equal to the threshold.

- Provider's email is viable.

Post conditions:

- Success:

    Provider successfully gets notified for his low balance in the wallet.

- Fail: Show error message of failure to notify the provider when his balance is low.

Main Success Scenario:

Step	Actor Action	System Response
1	Conducts some fee-involved activities	Update the balance of the provider's account. If the balance is lower than the defined threshold, the system will send

112

		an email and a popup notification as he logs in the system.
2	The provider gets a notification of low balance from the system in his mailbox or logs in to receive the notification. He will subsequently need to deposit some money into his wallet.	

Exceptions:

No	Cause	System Response
1	The provider does not get notified of his low balance.	The system will fix the warning on the header of the provider's homepage.

Relationships: N/A

Business Rules:

● Providers get notified if the balance is lower than the first threshold (10000 by default) and second threshold (5000) through registered email and popup notification on the app.

● The message will remain until the balance is brought higher than the first threshold.

● In case the balance is lower than -10000, his account will be deactivated and must resolve the problem by mail to the system admin.

3.2.3.17. <Provider> View Subscription Plan

Figure: <Provider> View Subscription Plan

USE CASE – UC_30

Use Case No.	**UC_30**	Use Case Version	2.0
Use Case Name	View Subscription Plan		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to view his subscription plans that he has registered up to now.

Goal:

- Provider successfully views the subscription plans that he has registered.

Triggers:

- The provider clicks on the "Registered Plans" button on the subscription plan main page.

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

Post conditions:

- Success:

    Provider successfully views a list of subscription plans he has contracted with the latest plan being placed on top.

- Fail: Show error message of failure to retrieve subscription history.

Main Success Scenario:

Step	Actor Action	System Response
1	Provider navigates to the nav bar and clicks on the "Subscription" tab	The system displays the subscription plan main page, containing specific pages for "Register" and "View Subscription History" and the subscription plan that the provider is currently on.

2	Provider clicks on the "View Subscription History" button.	The system displays a list of subscription plan registration that the provider has contracted, the order of which is chronologically managed. Each registration contains the following fields: - Subscription plan registered - Amount of charge - Date - Extend option if it is the latest registration (for subscription extension)
3	The provider can additionally perform some filtration or sorting (by date/by subscription plan) according to his desire.	The system sorts/filters/ the results based on the requirements of the provider.

Exceptions:

No	Cause	System Response
1	Internal error in sorting or filtering results.	The system will prompt a message notifying the trouble and ask the provider to try again later. The provider can email system admin about this contingency.

Relationships: N/A

Business Rules:

- Providers can view the subscription history that he has registered over the past 1 year.

- The history is not subject to modification neither by the provider nor the admin. The background worker of the system would periodically remove the past history older than 1 year old.

- The latest registration exhibits an extra feature: extend registration to allow the provider to automatically extend the plan when it reaches the end of the period.

3.2.3.18. <Provider> Register a Subscription Plan

Figure: <Provider> Register a Subscription Plan

USE CASE – UC_31			
Use Case No.	**UC_31**	Use Case Version	2.0
Use Case Name	Register A Subscription Plan		
Author	Nguyen Hoang Nam		
Date	17/06/2023	Priority	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to register a subscription plan available in the system.

Goal:

- Provider successfully registers a subscription plan.

Triggers:

- Provider clicks the "Register" button on the selected subscription plan on the subscription page.

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

- The current balance in the provider's wallet must exceed the subscription charge of the plan he intends to register.

Post conditions:

- Success:

  Provider successfully registers a subscription plan and his day count increments by the duration of the selected plan.

- Fail: Show error message of failure to register a plan.

Main Success Scenario:

Step	Actor Action	System Response
1	User logins as provider	Activate provider's availability status and displays main view board for provider
2	Clicks on the "Subscription" tab on the nav bar	Display the current subscription plan that the provider has registered, including day count left (due date). On the side bar, there are 2 options: "Register" and "View subscription history"
3	Provider clicks on the "Register" button.	The system displays the current balance in the provider's wallet and a list of available subscription plans, each of which includes details: - Plan name - Plan description - Duration - Price - "Register" button.
4	Provider selects a plan and clicks on the "Register" button of the selected plan.	- The system checks if the current balance is sufficient to make a registration. If it is,

		the system proceeds to save the subscription fee information in the database, and automatically creates a transaction entry in the provider's transaction history, subtracting an amount equal to the plan price from the current balance of the provider.
		- The day count is extended with the duration of the plan. The provider will be sent back to the subscription main page.
5	The provider can optionally set the latest subscription plan as auto-extend so that the system will automatically register a new same-kind subscription plan when the day count reaches the threshold.	- Save the auto-extend subscription and trigger the extension of the subscription plan automatically as the day count reaches the threshold.

Exceptions:

No	Cause	System Response
1	Internal error adding subscription fee and transaction entry.	The system displays a message or error and requires the provider to reattempt later.
2	The provider's current balance is lower than the price of the selected subscription plan.	The system replies with a message warning that the current balance is lower than the charge, hence, it fails to make a successful registration. The system will ask the provider to deposit money into his wallet or

		pick another subscription plan that fits in his current balance.

Relationships: N/A

Business Rules:

● Providers can register a subscription plan, which keeps him present on the system.

● There are 4 subscription plans for providers to choose. The longer the duration, the higher the price.

● The provider will be notified when his day count is lower than 5 days. If the provider's day count is lower than -3, he will be suspended automatically by the system.

3.2.3.19. <Provider> Get Notified When Subscription is Close to End

Figure: <Provider> Get Notified When Subscription is Close to End

USE CASE – UC_32			
Use Case No.	**UC_32**	**Use Case Version**	2.0
Use Case Name	Get Notified When Subscription is Close to End		
Author	Nguyen Hoang Nam		
Date	17/06/2023	**Priority**	High

Actor:

- Provider (primary), system (secondary)

Summary:

- This use case allows a provider to get notified when day count (plan subscription) is near the expiration.

Goal:

- Provider successfully gets notified when his day count is below the defined thresholds.

Triggers:

- The day count drops to or lower than the defined thresholds.

Preconditions:

- User must login to the system as provider

- Provider account is active in the system

- The day count is lower or equal to the threshold.

- Provider's email is viable.

Post conditions:

- Success:

    Provider successfully gets notified for his near expiration of subscription.

- Fail: Show error message of failure to notify the provider when the subscription is about to end.

Main Success Scenario:

Step	Actor Action	System Response
1		-The system scheduler conducts daily decrements on the day count of all providers and triggers notification to counts that are below the thresholds.
2	The provider gets a notification of low balance from the system in his mailbox or logs in to receive the notification. He will subsequently need to register a subscription plan or let the system automatically extend the subscription.	

Exceptions:

No	Cause	System Response

1	The provider does not get notified of his day count.	The system will fix the warning on the header of the provider's homepage.

Relationships: N/A

Business Rules:

● Providers get notified if the day count is equal or lower than the first threshold (3 days), and the second threshold (1 day) through registered email and popup notification on the app.

● The message will remain until the day count is brought higher than the first threshold.

● In case the day count is lower than -1, his account will be deactivated and must resolve the problem by mail to the system admin.

3.2.3.20. <Provider> Logout
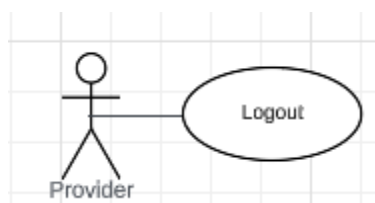
Figure: <Provider> Logout

USE CASE - UC33			
Use Case No.	**UC33**	**Use Case Version**	1.1
Use Case Name	Logout		
Author	Nguyen Ba Huy		
Date	23/06/2023	**Priority**	Normal

Actor:

-        Provider

Summary:

-        This use case allows the provider to logout to the system.

Goal:

-        Provider can logout from the system

Trigger:

-        Provider sends a logout command.

Precondition:

-        Provider must log in before logging out.

Post Conditions:

-        Success: Provider logout to the system successfully

Main Success Scenario:

Step	Actor Action	System Response
1	Provider clicks "Logout" on the navigation menu.	Return the user to the home screen.

Relationship: N/A.

Business rule:

● After logged out, user access the system as role "Provider"

● Return the user to the home screen after logged out.

3.2.3.21. <Provider> Update account information
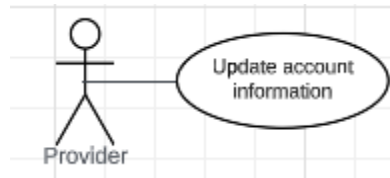
Figure: <Provider> Update account information

USE CASE-34 SPECIFICATION			
Use-case No.	UC34	**Use-case Version**	1.0
Use-case Name	Update account information		
Author	Nguyen Ba Huy		
Date	23/06/2023	**Priority**	Normal

Actor:

-		Provider

Summary:

-		This use case allows Provider to update their account information

Goal:

-		Provider can update their account information.

Triggers:

-		Provider sends an update account information command to the system.

Preconditions:

-		Provider must login into the system with the role Provider.

Post conditions:

-		Success: update the account information successfully

Main Success Scenario:

Step	Actor	System Response
1	Provider	System re

	clicks "Account" on navigation menu	direct to the account screen with account information of the provider.

2	Provider clicks on "Update" button	System show a form pop-up for the provider to fill in.
3	Provider fills in	System will show the mes

	details (name, email, phone) and hits the "Submit" button.	sage "Update account information successfully" and return to the account screen.

Alternative Scenario:

Step	Actor	System Response
1	Provider is not logged in or their account is invalid	Provider is redirected to the login/registration page.

Exceptions:

No	Actor	System Response
1	The provider's login session expires during the	They will be prompted to login again before e acc

	pr oc ess	ess ing the ir bo oki ng his tor y

Relationships: N/A

Business Rules:

- Displays provider's information based on provider's id

- Provider's account information must have:

  1. Name of provider

  2. Email

  3. Phone

3.2.4.   <Admin> Overview Use Case
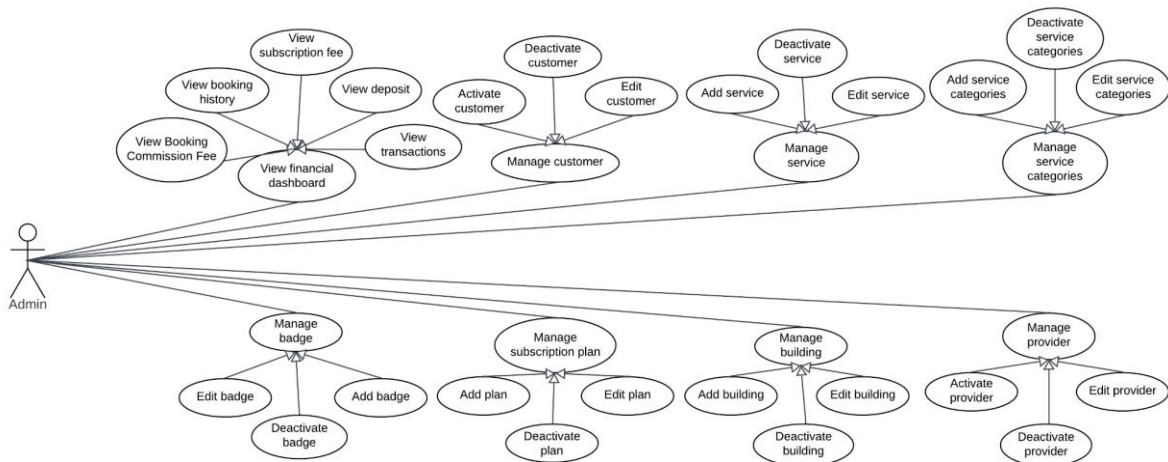
Figure: <Admin> Overview Use Case

3.2.4.1. <Admin> Add service categories
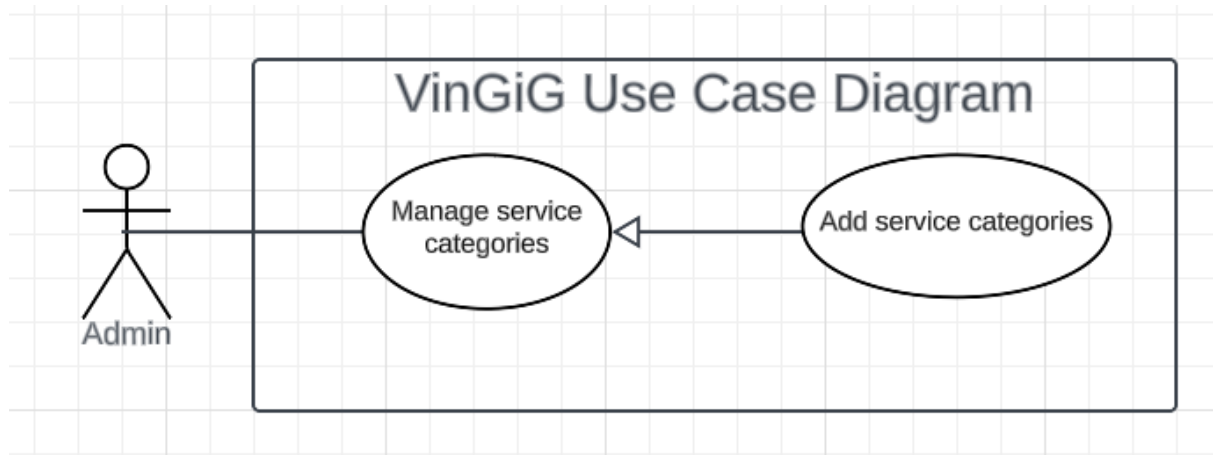
Figure: <Admin> Add service categories

USE CASE – UC35			
Use Case No.	**UC35**	**Use Case Version**	2.0
Use Case Name	Add service categories		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	High
Actor:			

- Admin

Summary:

- This use case allows the admin to create a new services category

Goal:

- The admin successfully create a new service category

Triggers:

- The admin clicks on "Add service categories"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

 Admin successfully creates a new services category

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "service categories" tab on nav bar	Displays a list of existing service categories
2	Selects the option to add a new service category	Show a form to enter the new service category's details: ● Name ● Description
3	Enter the details of the new Service category	System shows a message that the service category is created. [Exception 1] [Exception 2]

Exceptions:

No	Cause	System Response
1	User input invalid fields	System shows message with corresponding fields: Service type name: Must have 6 to 20 characters

No	Cause	System Response
2	Service categories already exist	displays an error message indicating that the Service categories already exists

Relationships: N/A

Business Rules:

● Service type name: Must have 6 to 20 characters

● Admin can view the details of service before clicking "Finish".

● When Admin click "Finish":

    ○ Shows message "Added service category".

    ○ Redirects to the "Service categories list" screen.

3.2.4.2. <Admin> Deactivate service categories

Figure: Deactivate service categories

USE CASE – UC36			
Use Case No.	**UC36**	**Use Case Version**	2.0
Use Case Name	Deactivate service categories		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal

Actor:

 - Admin

Summary:

 - This use case allows admin to deactivate service categories

Goal:

 - Admin successfully deactivates service categories

Triggers:

 - Admin clicks " deactivated"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

- Success:

Admin successfully deactivates service categories

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Loại dịch vụ" tab on nav bar	Displays a list of existing Service categories
2	Selects the Service categories to be deactivated from the list	System shows a message that the service categories are deactivated and services are also deactivated. [Exception 1]

Exceptions:

No	Cause	System Response
1	There are services depending on this category	The system displays a message stating, "There are services dependent on this category."

Relationships: Manage services

Business Rules:

● Admin can view the details of service before clicking "deactivate".

● When Admin click "deactivate":

      ○ Show message "Service categories deactivated".

      ○ Redirect to the "Service categories list" screen.

3.2.4.3. <Admin> Edit service categories
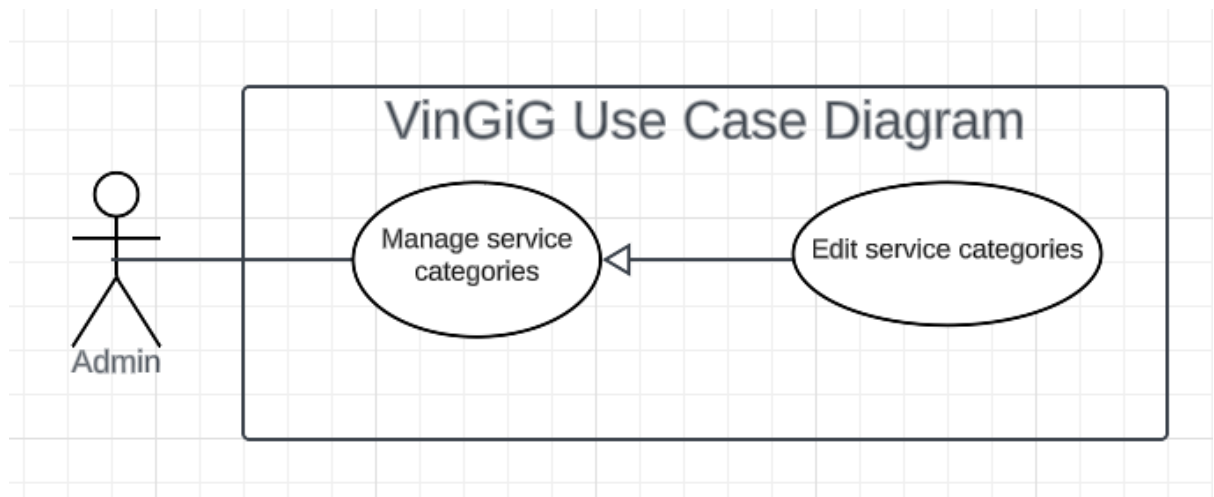
Figure: Edit service categories

USE CASE – UC37			
Use Case No.	**UC37**	**Use Case Version**	2.0
Use Case Name	Edit service categories		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal

Actor:
   - Admin
 Summary:
   - This use case allows admin to edit a service category's details			

Goal:

- The admin successfully edits a service category's details

Triggers:

- The admin clicks on " Edit"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

- Success:

> Admin successfully edits the selected services category's details

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Service Categories" tab on the nav bar	Displays a list of existing service categories
2	Selects the option to edit a service category	Shows a form to enter the selected service category's details: ● name ● description
3	Enters the details of the new service category	System shows a message that the service category is edited. [Exception 1] [Exception 2] [Exception 3]

Exceptions:

No	Cause	System Response
1	User enters invalid inputs	System shows message with corresponding fields: Service type name: Must have 6 to 20 characters

No	Cause	System Response
2	The service category already exists	displays an error message indicating that the Service categories already exists

No	Cause	System Response
3	There are services depending on this category	The system displays a message stating, "There are services dependent on this category."

Relationships: N/A

Business Rules:

● Service type name: Must have 6 to 20 characters

- Admin can view the details of service before clicking "finish".

- When Admin click "Finish":

    o Shows message "Service categories edited".

    o Redirects to the "Service categories list" screen.

3.2.4.4. <Admin> Add services
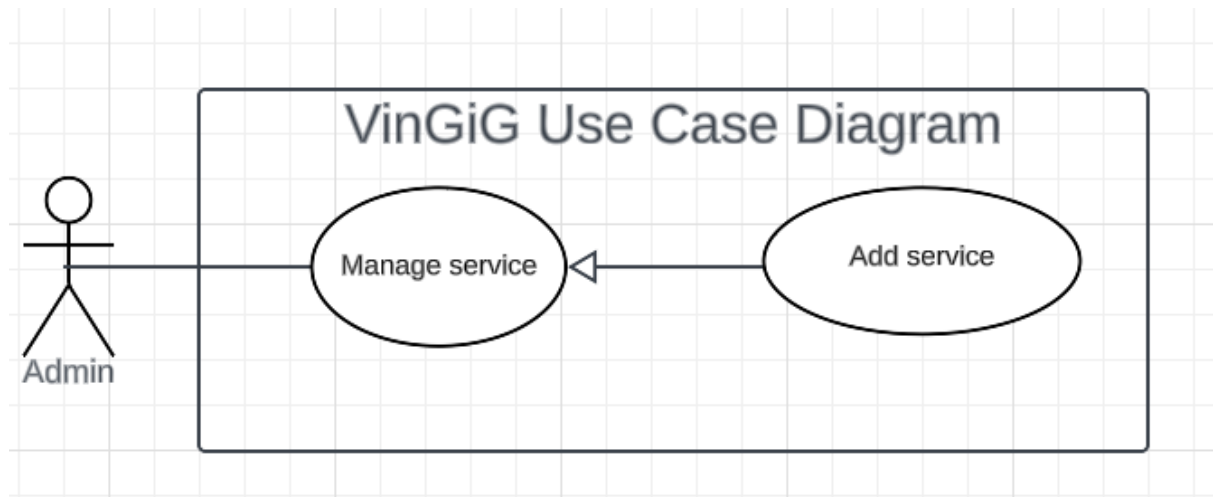
Figure: Add services

USE CASE – UC38			
Use Case No.	**UC38**	Use Case Version	2.0
Use Case Name	Add services		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	High

Actor:

- Admin

Summary:

- This use case allows the admin to add a new service

Goal:

- The admin successfully adds a new service

139

Triggers:

- The admin clicks on "Add" button

Preconditions:

- User must login to the system as system admin

- Admin account must be activated

Post conditions:

- Success:

    Admin successfully creates a new service

-

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Services" tab on the nav bar	Displays a list of existing services
2	Selects the option to add a new service	Shows a form to enter new service details: name,service categories, description, unit, min price, max price, fee.
3	Enters the details of the new service	Shows a message that the service is created. [Exception 1] [Exception 2]

Exceptions:

No	Cause	System Response

1	User enters invalid inputs	System shows message with corresponding fields: Service Name: Must be at least 6 characters. The service category name must exist in the system Max price must be greater than min price Min price is greater than 0 Fees greater than 0

No	Cause	System Response
2	The service already exists	displays an error message indicating that the service already exists.

Relationships: N/A

Business Rules:

● Admin can view the details of service before clicking "Finish".

● When Admin click "Finish":

    o Shows message "Added service".

    o Redirects to the "Service list" screen.

3.2.4.5. <Admin> Deactivate services
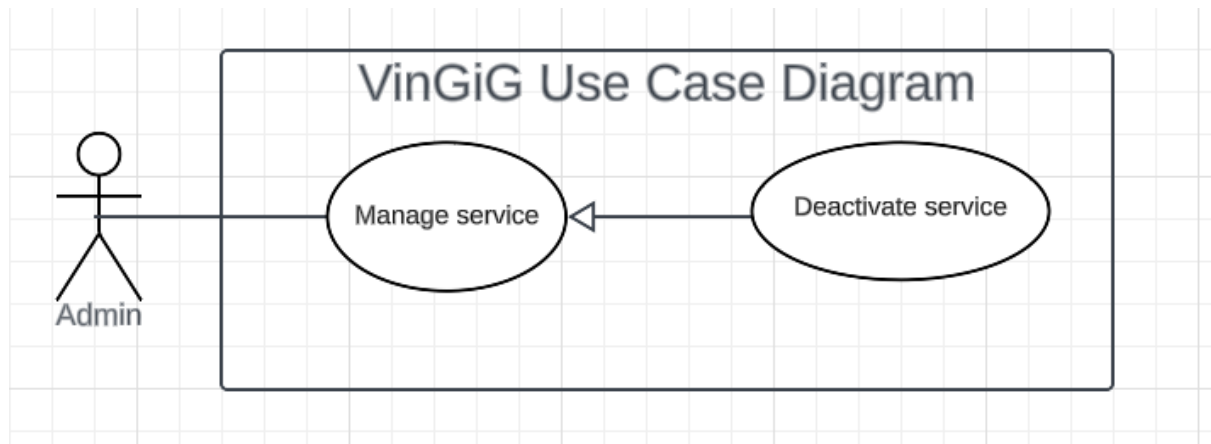
Figure: Deactivate services

USE CASE – UC39			
Use Case No.	**UC39**	**Use Case Version**	2.0
Use Case Name	Deactivate services		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal
Actor: - Admin **Summary:** - This use case allows admin to deactivate a service **Goal:** - The admin successfully deactivates the selected service **Triggers:** - The admin clicks on "Delete" button **Preconditions:**			

- User must login to the system

- Admin account must be activated

Post conditions:

Admin successfully deactivates a service

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "services" tab on nav bar	Displays a list of existing services
2	Selects the service to be deactivated from the list	System shows a message that the service is deactivated.

Exceptions: N/A

Relationships: N/A

Business Rules:

● Admin can view the details of service before clicking "deactivate".

● When Admin click "Finish":

    ○ Shows message "Service deactivated".

    ○ Redirects to the "Service list" screen.

3.2.4.6. <Admin> Edit services
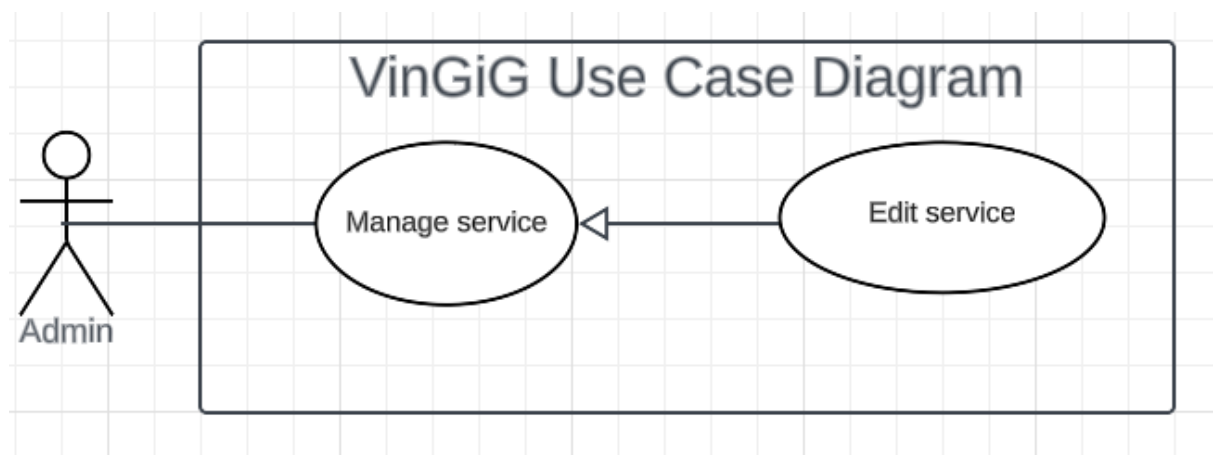
Figure: Edit services

USE CASE – UC40			
Use Case No.	**UC40**	**Use Case Version**	2.0
Use Case Name	Edit services		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal
Actor: - Admin **Summary:** - This use case allows admin to edit a service's details			

Goal:

- The admin successfully edit the selected service's details

Triggers:

- The admin clicks on " Edit"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

Admin successfully edits the selected service's details

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "services" tab on nav bar	Displays a list of existing services
2	Selects the option to edit a service category	Shows a form to enter the service's details: ● Name ● service categories ● description
3	Enters the details of the new service	System shows a message that the service is edited. [Exception 1] [Exception 2]

Exceptions:

No	Cause	System Response

No	Cause	System Response
1	User inputs invalid fields	System shows message with corresponding fields: Service Name: Must be 6 to 20 characters. Service type name must exist in the system Max price must be greater than min pride Min pride is greater than 0 Fees greater than 0

No	Cause	System Response
2	Service already exists	displays an error message indicating that the service already exists

Relationships: N/A

Business Rules:

Service Name: Must be 6 to 20 characters.

Service type name must exist in the system

Max price must be greater than min pride

Min pride is greater than 0

Fees greater than 0

- The admin can view the details of the service before clicking "Finish".

- When the admin clicks on "Finish":

    o   Show message "Service edited".

    o   Redirect to the "Service list" screen.

3.2.4.7. <Admin> Add buildings
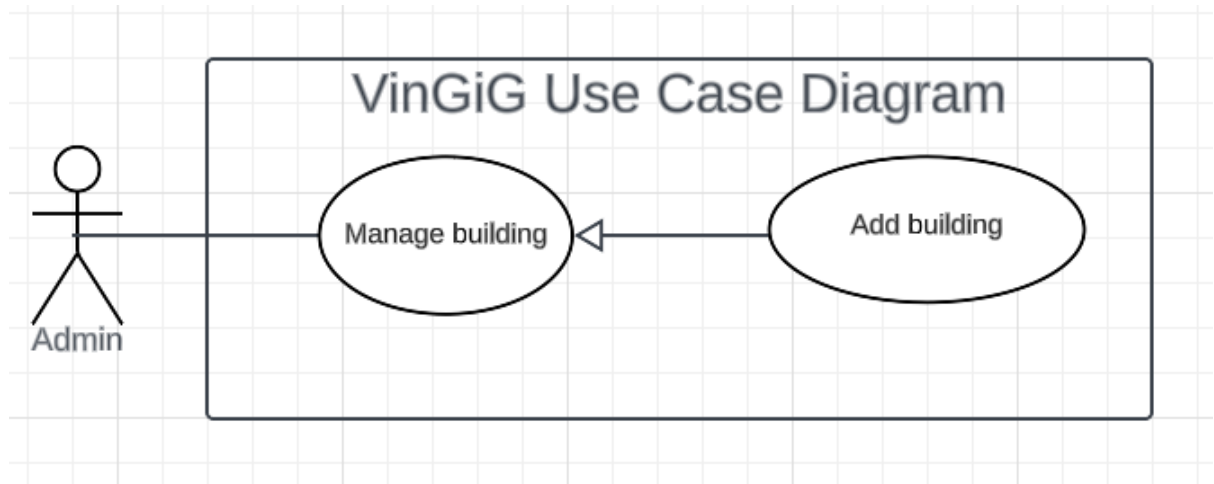
Figure: Add building

USE CASE – UC41			
Use Case No.	**UC41**	**Use Case Version**	2.0
Use Case Name	Add buildings		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	High
Actor: - Admin **Summary:** - This use case allows admin to add a new building **Goal:**			

- Admin successfully adds a new building

Triggers:

- Admin clicks on "Add buildings"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

Admin successfully adds a new building

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Building" tab on nav bar	Displays a list of existing buildings
2	Selects the option to add a new building	Show a form to enter the new building's details (name, description.)
3	Enter the details of the new building	System shows a message that the building is created. [Exception 1] [Exception 2]

Exceptions:

No	Cause	System Response
1	User input invalid fields	System shows message with corresponding fields:

No	Cause	System Response
		Building name: Must have 6 to 20 characters.

No	Cause	System Response
2	The building already exists	displays an error message indicating that the building already exists

Relationships: Manage provider, Manage customer

Business Rules:

● Building name: Must have 6 to 20 characters.

● Admin can view the details of the building before clicking "finish".

● When Admin click "finish":
   ○ Show message "Building Added".
   ○ Redirect to the "Building list" screen.

3.2.4.8. <Admin> Deactivate buildings
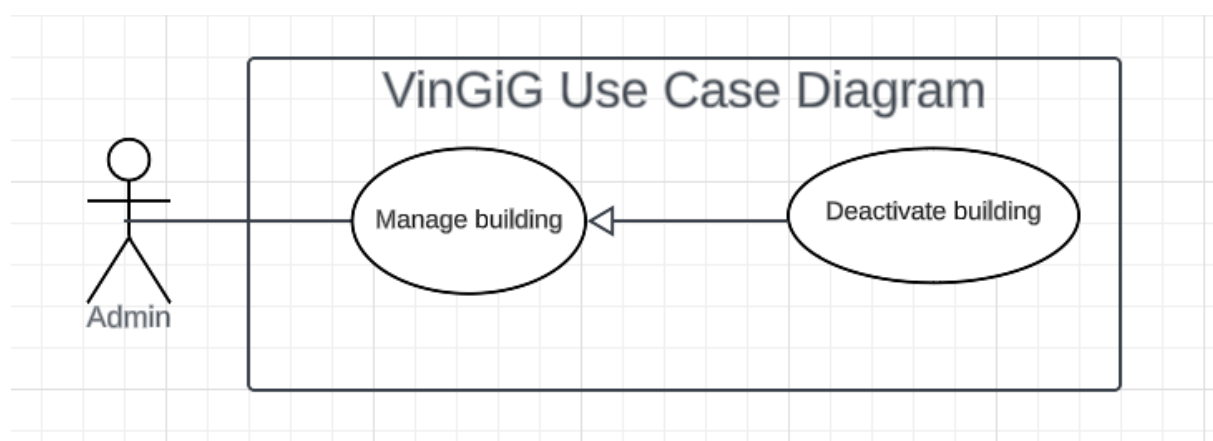
Figure: Deactivate buildings

USE CASE – UC42			
Use Case No.	**UC42**	**Use Case Version**	2.0
Use Case Name	Deactivate buildings		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal

Actor:

- Admin

Summary:

- This use case allows admin to deactivate a building

Goal:

- Admin successfully deactivate the selected building

Triggers:

- Admin click " Deactivate"

Preconditions:

- User must login to the system
- Admin account must be activated

Post conditions:

    Admin successfully deactivate the selected building

-

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "building" tab on nav bar	Displays a list of existing buildings

2	Selects the building to be deactivated from the list	System shows a message that the building is deactivated.

Exceptions:

No	Cause	System Response
1	There are providers and customers dependent on this building	Shows a message indicating, "There are providers and customers dependent on this building."

Relationships: Manage provider, Manage customer

Business Rules:

● Admin can view the details of service before clicking "finish".

● When Admin click "finish":

    o Show message "Building deactivated".

    o Redirect to the "Building list" screen.

3.2.4.9. <Admin> Edit buildings
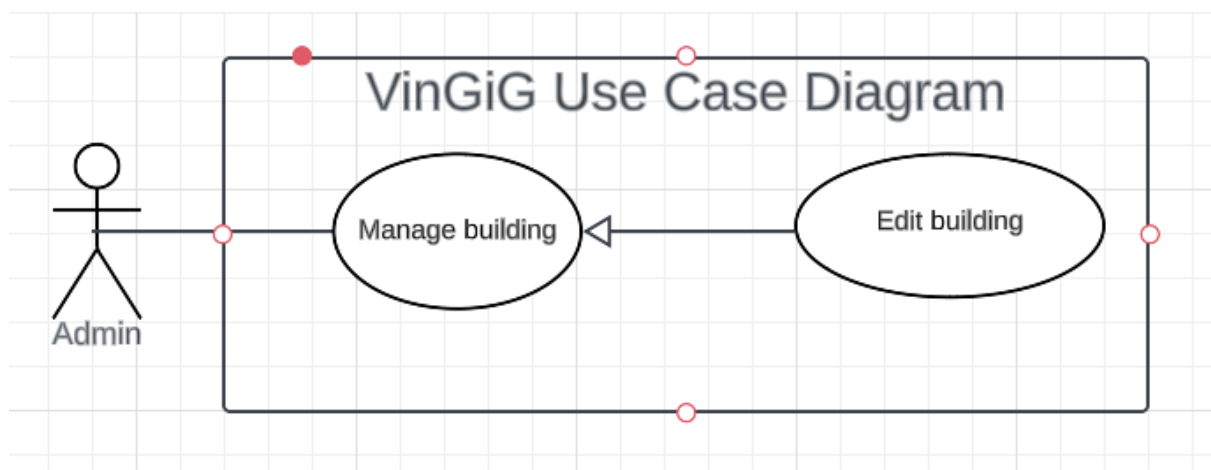
Figure: Edit buildings

USE CASE – UC43			
Use Case No.	**UC43**	**Use Case Version**	2.0
Use Case Name	Edit buildings		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal

Actor:

- Admin

Summary:

- This use case allows admin to edit a building's details

Goal:

- Admin successfully edit the selected building's details

Triggers:

- Admin clicks on "Manage building"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

      Admin successfully edits the selected building's details

-

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Building" tab on nav bar	Displays a list of existing buildings

152

2	Selects the option to edit a building's details	Shows a form to enter the building details
3	Enters the details of the new building (name, description, etc.)	System shows a message that the building's details are edited. [Exception 1] [Exception 2]

Exceptions:

No	Cause	System Response
1	User inputs invalid fields	System shows message with corresponding fields: Building name: Must have 6 to 20 characters.

No	Cause	System Response
2	The building already exists	Displays an error message indicating that the building already exists

No	Cause	System Response
3	There are providers and customers dependent on this building	Shows a message indicating, "There are providers and customers dependent on this building."

Relationships: N/A

Business Rules:

Building name: Must have 6 to 20 characters.

● Admin can view the details of the building before clicking "Finish".

● When Admin clicks "Finish":

  ○ Show message "Building edited".

  ○ Redirect to the "Building list" screen.

3.2.4.10. <Admin> Activate providers
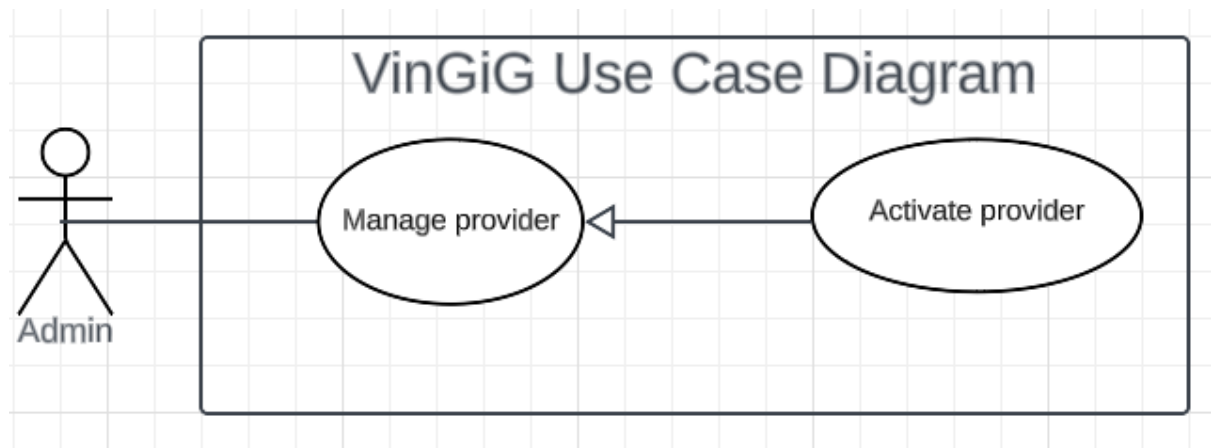
Figure: Activate providers

USE CASE – UC44			
Use Case No.	**UC44**	**Use Case Version**	2.0
Use Case Name	Activate providers		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	High
Actor:			
- Admin			

Summary:

- This use case allows the admin to activate a provider

Goal:

- Admin successfully activates a provider

Triggers:

- Admin clicks on " Activate"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

Admin successfully activates the provider

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Providers" tab on nav bar	Displays a list of existing providers
2	Selects the option to activate a provider	System shows a message that the provider is activated.

Exceptions: N/A

Relationships: N/A

Business Rules:

- Admin can view the details of the provider before clicking "Finish".

- When Admin Clicks "Finish":

    o Show message "Provider is activated".

    o Redirect to the "Provider list" screen.

3.2.4.11. <Admin> Deactivate providers

Figure: Deactivate providers

USE CASE – UC45			
Use Case No.	**UC45**	**Use Case Version**	2.0
Use Case Name	Deactivate providers		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal

Actor:

- Admin

Summary:

- This use case allows admin to deactivate a provider

Goal:

- Admin successfully deactivates the selected provider

Triggers:

- Admin click "Deactivate"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

- Success:

    Admin successfully deactivate the provider

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "nhà cung cấp" tab on nav bar	Displays a list of existing provider
2	Selects the option to deactivate a provider	System shows a message that the provider is deactivated.

Exceptions: N/A

Relationships: N/A

Business Rules:

- Admin can view the details of the provider before clicking "Finish".

- When Admin clicks "Finish":

    o Show message "Provider deactivated".

    o Redirect to the "Provider list" screen.

3.2.4.12. <Admin> Edit providers
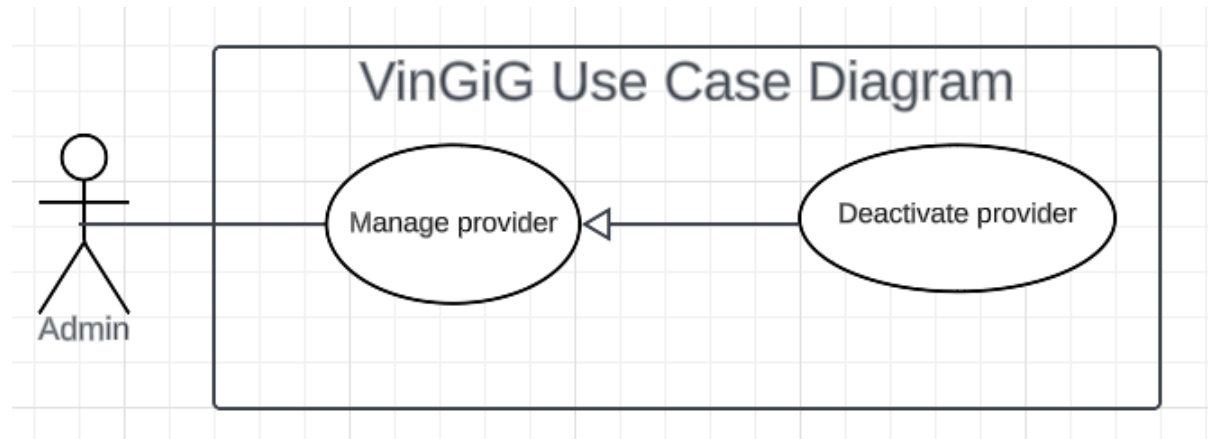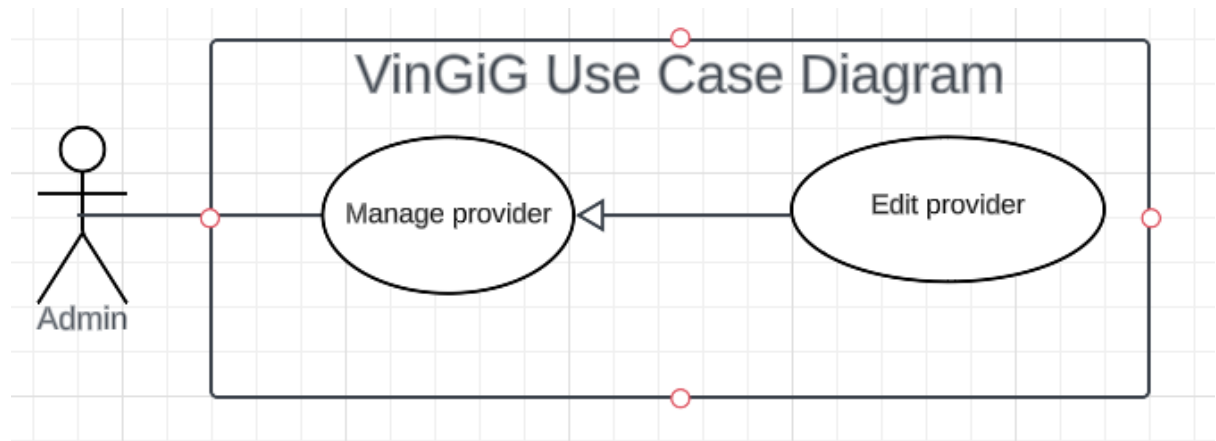

Figure: Edit providers

USE CASE – UC46			
Use Case No.	**UC46**	**Use Case Version**	2.0
Use Case Name	Edit a providers		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal

Actor:

- Admin

Summary:

- This use case allows admin to edit a provider's details

Goal:

- Admin successfully edits provider

Triggers:

- Admin clicks on "Edit"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

Admin successfully edits provider

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Providers" tab on nav bar	Displays a list of existing provider
2	Selects the option to edit a Provider	Shows a form to enter the provider's details (username, password, gender, building, badge, avatar, rating, fullname, email, phone, address.)
3	Enters the details of the provider	System shows a message that the provider is edited. [Exception 1] [Exception 2]

Exceptions:

No	Cause	System Response
1	User enters invalid inputs	System shows message with corresponding fields: Username: Must be 6 to 20 characters. Password: Must have 6 to 20 characters. Building : must exist in the table building in the database

		Badge: must exist in
		Table badge in database
		Avatar: it's a picture
		Rating: must be less than 5
		Full name: Must be 6 to 20 characters.
		Email: must have email a format
		Phone: must have 10 digits
		Address: Must be 6 to 20 characters.

No	Cause	System Response
2	Provider already exists	Displays an error message indicating that the provider already exists

Relationships: N/A

Business Rules:

Username: Must be 6 to 20 characters.

Password: Must have 6 to 20 characters.

Building : must exist in the table building in the database

Badge: must exist in

Table badge in database

Avatar: it's a picture

Rating: must be less than 5

Full name: Must be 6 to 20 characters.

Email: must have email a format

Phone: must have 10 digits

Address: Must be 6 to 20 characters.

● Admin can view the details of the provider before clicking "Finish".

● When Admin clicks "Finish":

> o   Shows message "Provider edited".
>
> o   Redirects to the "Provider list" screen.

3.2.4.13. <Admin> Activate customers
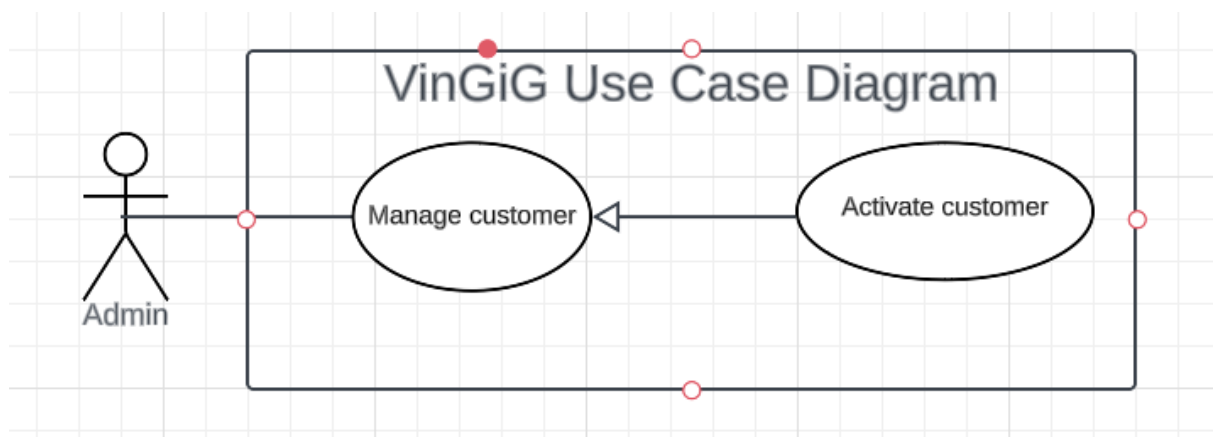
Figure: Activate customers

USE CASE – UC47			
Use Case No.	**UC47**	**Use Case Version**	2.0
Use Case Name	Activate customers		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	High
Actor:			

- Admin

Summary:

- This use case allows admin to activate customers

Goal:

- Admin successfully activates customer

Triggers:

- Admin click "Add"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

- Success:

    Admin successfully activates customer

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Customer" tab on the nav bar	Displays a list of existing customers
2	selects the option to activate a customer	System shows a message that the customer is activated

Exceptions: N/A

Relationships: N/A

Business Rules:

- Admin can view the details of the customer before clicking "Finish".

- When Admin click "finish":

    o Shows message "Customer activated".

    o Redirects to the "Customer list" screen.

3.2.4.14. <Admin> Deactivate customers

Figure: Deactivate customers

USE CASE – UC48			
Use Case No.	**UC48**	**Use Case Version**	2.0
Use Case Name	Deactivate customers		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal
Actor: - Admin **Summary:** - This use case allows admin to deactivate customers **Goal:**			

- Admin successfully deactivates customer

Triggers:

- Admin click "Deactivate"

Preconditions:

- User must login to the system

- Admin account must be activated

Post conditions:

- Success:

Admin successfully detective customer

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Customer" on the nav bar	Displays a list of existing customer
2	Selects the option to deactivate a customer	System shows a message that the customer is deactivated.

Exceptions: N/A

Relationships: N/A

Business Rules:

● Admin can view the details of the customer before clicking "Finish".

● When Admin click "Finish":

    o Show message "Customer is deactivated".

    o Redirect to the "Customer list" screen.
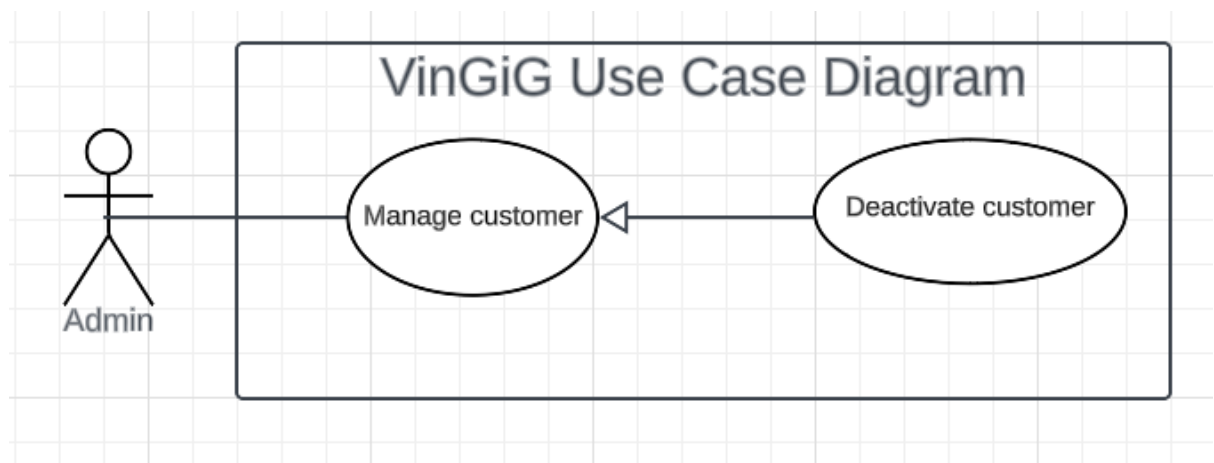
3.2.4.15. <Admin> Edit customers

Figure: Edit customers

USE CASE – UC49			
Use Case No.	**UC49**	**Use Case Version**	2.0
Use Case Name	Edit customers		
Author	Le Trung Duc		
Date	07/06/2023	**Priority**	Normal

Actor:

  - Admin

Summary:

  - This use case allows admin to edit customer

Goal:

  - Admin successfully edits customer

Triggers:

  - Admin clicks "Edit"

Preconditions:

  - User must login to the system

- Admin account must be activated

Post conditions:

- Success:

Admin successfully edit customer

Main Success Scenario:

Step	Actor Action	System Response
1	Clicks on the "Customer" tab on nav bar	Displays a list of existing customer
2	Selects the option to edit the customer.	Shows a form to enter new customer details(username, password, gender, building, avatar, rating, fullname, email, phone, address)
3	Enters the details of the new customer	System shows a message that the customer is edited. [Exception 1] [Exception 2]

Exceptions:

No	Cause	System Response
1	User inputs invalid fields	System shows message with corresponding fields: Username: Must be 6 to 20 characters. Password: Must have 6 to 20 characters.

		Building : must exist in the table building in the database
		Avatar: it's a picture
		Rating: must be less than 5
		Full name: Must be 6 to 20 characters.
		Email: must have email a format
		Phone: must have 10 digits
		Address: Must be 6 to 20 characters.

No	Cause	System Response
2	Provider already exists	Displays an error message indicating that the customer already exists

Relationships: N/A

Business Rules:

Username: Must be 6 to 20 characters.

Password: Must have 6 to 20 characters.

Building : must exist in the table building in the database

Avatar: it's a picture

Rating: must be less than 5

Full name: Must be 6 to 20 characters.

Email: must have email a format

Phone: must have 10 digits

Address: Must be 6 to 20 characters.

● Admin can view the details of the customer before clicking "finish".

● When Admin click "finish":

    o Show message "Customer edited".

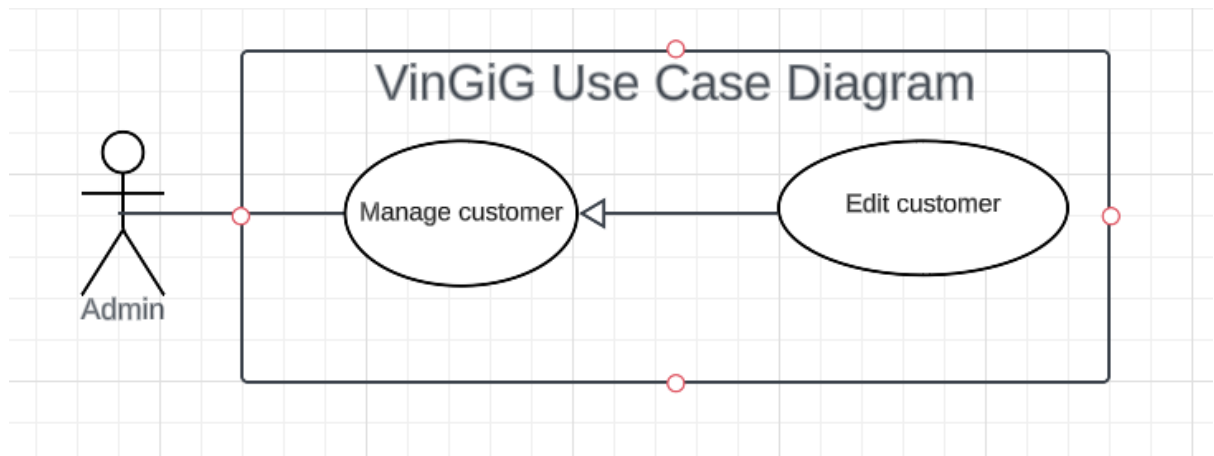   o Redirect to the "Customer list" screen.

3.2.4.16. <Admin> Add a badge
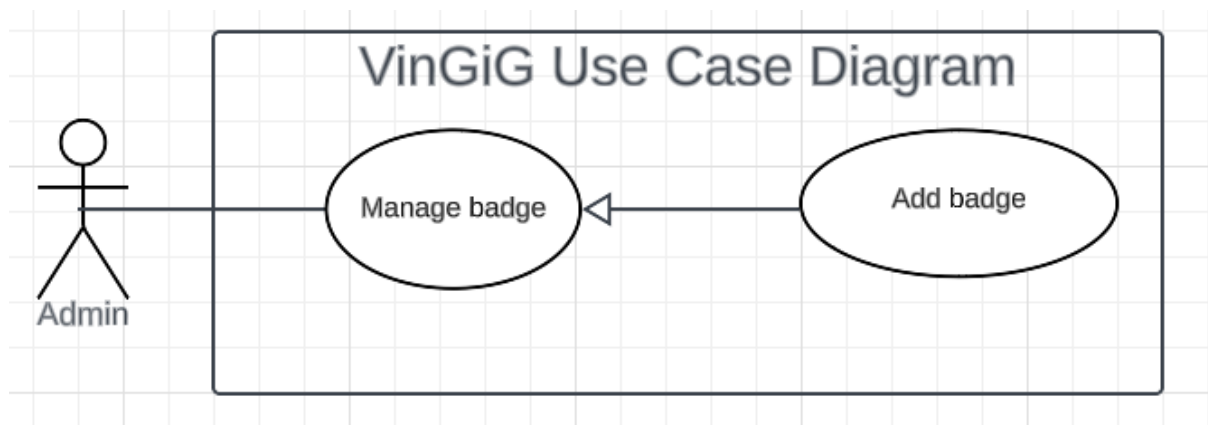
Figure: Add a badge

USE CASE-50 SPECIFICATION			
Use-case No.	UC050	Use-case Version	1.0
Use-case Name	Add a badge		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

    Admin

Summary:

The system allows the admin to add a badge.

Goal:

The admin is able to add a new badge.

Triggers:

    The admin clicks on the "Add a badge" option.

Preconditions:

    The user has to log in successfully as an admin.

Post Conditions:

    The badge is added to the system.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged-in user selects the "Manage badges" option on the admin homepage.	The system navigates the user to "Badges management" screen
2	The user clicks on "Add a badge" option	The system shows a form. The form includes input fields for: The

		badge's name, description and its corresponding benefits.
3	The user enters the badge's details and clicks "Submit" button	The system added the newly created badge. [Exception]

Alternative Scenario:

   N/A

Exceptions:

No.	Cause	System's response
1	The user enters wrong a format input	The system displays a pop-up message: the entered data is invalid.

Relationships:

   This use case relates to the "Manage badges" functionality in the user system.

Business Rules:

   The system validates the user's input with the following constraints:

   o   Badge's name: has less than 20 characters.

   o   Description: has less than 200 characters.

   o   Benefit: has less than 200 characters.

3.2.4.17. <Admin> Edit a badge

Figure: Edit a badge

USE CASE-51 SPECIFICATION			
Use-case No.	UC51	Use-case Version	1.0
Use-case Name	Edit a badge		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

    Admin

Summary:

The system allows the admin to edit badge details, including: the badge's name, its description and corresponding benefits.

Goal:

The admin is able to edit a selected badge's details.

Triggers:

The admin clicks on the "Edit" option on a chosen badge.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

The chosen badge's details are edited successfully.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged user selects the "Manage badges" option on the admin homepage.	The system navigates the user to "Badges management" screen
2	The user clicks on "Edit a badge" option	The system displays a list of active badges.
3	The user selects the "Edit" button on a badge.	The system shows a form. The form includes input fields for: Badge name, description and benefit. The fields are already filled with the badge's previous

		details. Input validation is required.
4	The user enters the badge's details and clicks "Submit" button	The system updates the badge's details. [Exception]

Alternative Scenario:

N/A

Exceptions:

No.	Cause	System's response
1	The user enters wrong a format input	The system displays a pop-up message: the entered data is invalid.

Relationships:

This use case relates to the "Manage badges" functionality in the user system.

Business Rules:

The system validates the user's input with the following constraints:

- o Badge name: has less than 20 characters.

- o Description: has less than 200 characters.

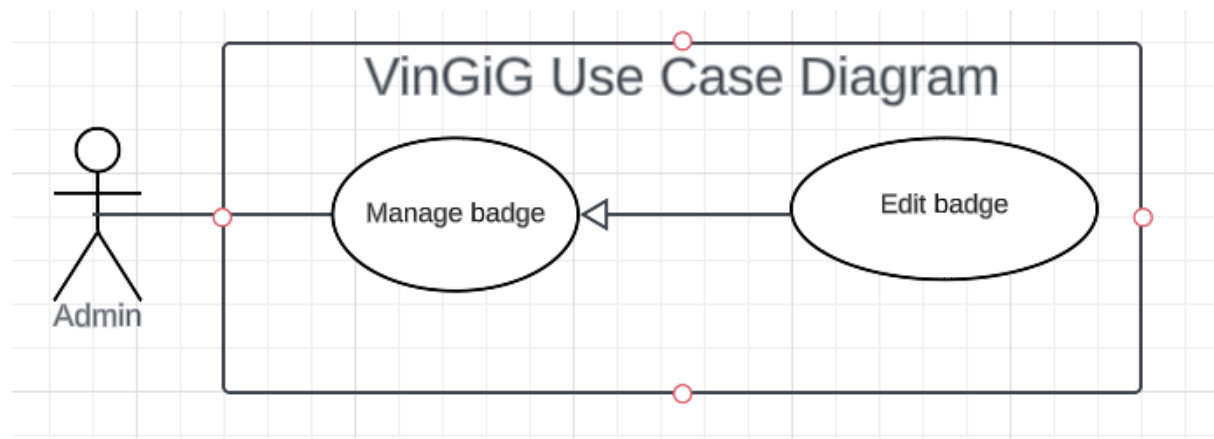- o Benefit: has less than 200 characters.

3.2.4.18. <Admin> Deactivate a badge
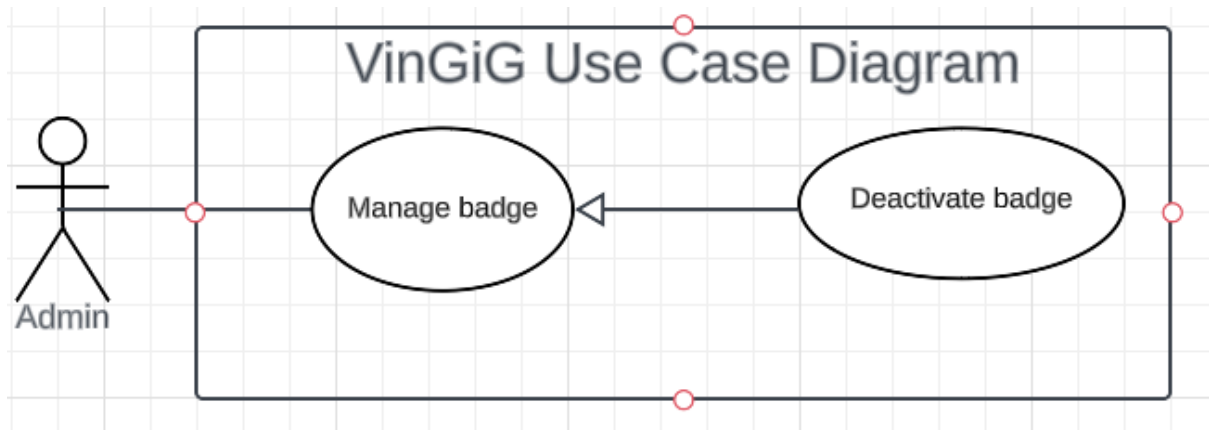
Figure: Deactivate badge

USE CASE-52 SPECIFICATION			
Use-case No.	UC52	Use-case Version	1.0
Use-case Name	Deactivate a badge		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

    Admin

Summary:

The system allows the admin to deactivate a badge.

Goal:

The admin is able to deactivate a selected badge.

Triggers:

The admin clicks on the "Deactivate a badge" option.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

The badge is deactivated successfully.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged admin selects the "Badges" tab on the admin's homepage nav bar.	The system navigates the user to "Badges management" screen
2	The user clicks on "Deactivate a badge" option	The system shows a list of active badges, including badgeID, badgeName and a link to details. [Exception 1]
3	The user selects "Deactivate" button on a selected badge	The system displays a pop-up message: "Are you sure to deactivate this badge?"
4	The user clicks on the "Confirm" button on the pop-up screen.	The system updates the badge's active state. [Exception 2]

Alternative Scenario:

N/A

Exceptions:

N/A

Relationships:

This use case relates to the "Manage badges" functionality in the user system.

Business Rules:

The list of active badges must provide these details: Badge's ID and badge's name.

3.2.4.19. <Admin> Add a plan
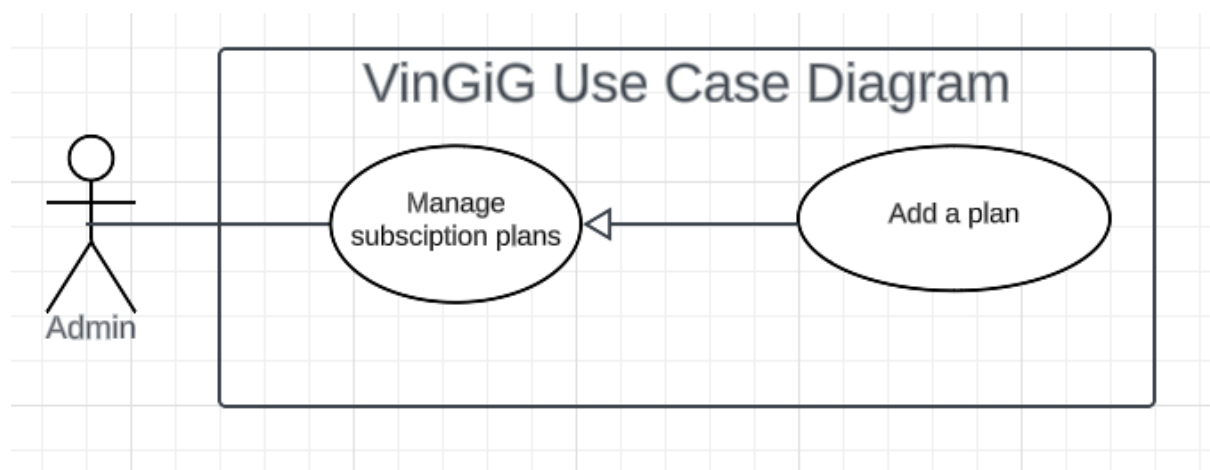
Figure: Add a plan

USE CASE-53 SPECIFICATION			
Use-case No.	UC53	Use-case Version	1.0

Use-case Name	Add a plan		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

Admin

Summary:

 The system allows the admin to add a subscription plan.

Goal:

 The admin is able to add a new subscription plan.

Triggers:

   The admin clicks on the "Add a plan" option.

Preconditions:

   The user has to log in successfully as an admin.

Post Conditions:

    Success scenario: The subscription plan is added to the system.

Main Success Scenario:

No.	Actor's action	System's response

1	The logged user selects the "Manage subscription plans" option on the admin homepage.	The system navigates the user to "Subscription plans management" screen
2	The user clicks on "Add a plan" option	The system shows a form with input fields: name, description, duration and price. These fields require validation.
3	The user enters the details and clicks "Submit" button	The system added the newly created subscription plan to the database. [Exception]

Alternative Scenario:

   N/A

Exceptions:

No.	Cause	System's response
1	The user enters wrong a format input	The system displays a pop-up message: the entered data is invalid.

Relationships:

   This use case relates to the "Manage subscription plans" functionality in the user system.

Business Rules:

   The system validates the user's input with the following constraints:

- Name: has less than 20 characters.

- Description: has less than 200 characters.

- Duration: Date a format, dd/mm/yyyy.

- Price: has less than 7 digits, with VND as currency.

3.2.4.20. <Admin> Edit a plan

Figure: Edit a plan

USE CASE-54 SPECIFICATION			
Use-case No.	UC54	Use-case Version	1.0
Use-case Name	Edit a plan		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

    Admin

Summary:

 The system allows the admin to edit a subscription plan's details.

Goal:

 The admin is able to edit the subscription plan's details.

Triggers:

    The admin clicks on the "Edit a plan" option.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

The subscription plan's details are edited successfully.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged user selects the "Manage subscription plans" option on the admin homepage.	The system navigates the user to "Subscription plans management" screen
2	The user clicks on "Edit a plan" option	The system displays a list of active plans. [Exception 1]
3	The user selects the "Edit" button on a selected plan.	The system displays a pop-up form, including: the plan's description, its duration and price. Input validation required.
4	The user enters the details and clicks "Submit" button	The system updates the subscription plan's details. [Exception 2]

Alternative Scenario:

N/A

Exceptions:

No.	Cause	System's response
1	The system fails to load the data	The system displays an error message: "Currently unable to retrieve data". The user can click on the button "Back to homepage" to navigate back to the homepage.
2	The user enters wrong a format input	The system displays a pop-up message: the entered data is invalid.

Relationships:

This use case relates to the "Manage badges" functionality in the user system.

Business Rules:

- The system validates the user's input with the following constraints:

- o Name: has less than 20 characters.

- o Description: has less than 200 characters.

- o Duration: Date a format, dd/mm/yyyy.

- o Price: has less than 7 digits, with VND as currency.

- Each subscription plan is displayed with basic details: the plan's ID, name, duration and price.

3.2.4.21. <Admin> Deactivate a plan
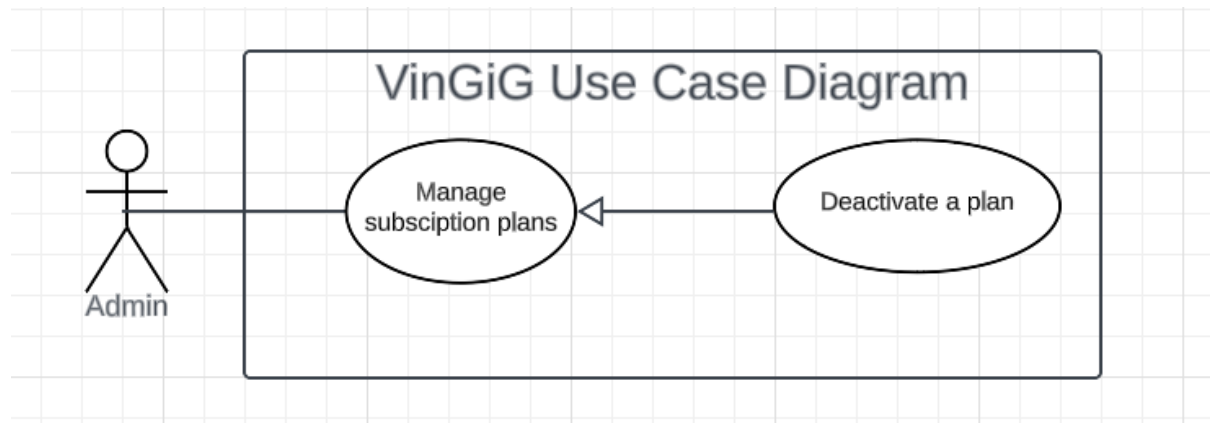
Figure: Deactivate a plan

USE CASE-55 SPECIFICATION			
Use-case No.	UC55	Use-case Version	1.0
Use-case Name	Deactivate a plan		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

Admin

Summary:

 The system allows the admin to deactivate a subscription plan instead of hard-deletion.

Goal:

 The user is able to deactivate a selected subscription plan.

Triggers:

The admin clicks on the "Deactivate a plan" option.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

 The subscription plan is deactivated successfully.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged user selects the "Manage subscription plans" option on the admin homepage.	The system navigates the user to the "Subscription plans management" screen
2	The user clicks on "Deactivate a plan" option	The system shows a list of subscription plans. [Exception 1]
3	The user selects "Deactivate" button on a selected subscription plan	The system displays a pop-up message: "Are you sure to deactivate this subscription plan?"
4	The user clicks on the "Confirm" button on the pop-up screen.	The system updates the subscription plan's active state. [Exception 2]

Alternative Scenario:

    N/A

Exceptions:

No.	Exception	System's response
1	The system fails to retrieve data	The system displays an error message
2	The system fails to update the selected subscription plan's state.	The system displays an error message: "Failed to update, please try again". After that, the user is navigated back to the "Subscription plans management" screen.

Relationships:

  This use case relates to the "Manage subscription plans" functionality in the user system.

Business Rules:

 All displayed subscription plans on the "Subscription plans management" screen must be in active state as default.

  When the system displays a pop-up message: "Are you sure to deactivate this subscription plan?", the admin is provided an option to cancel or deactivate the selected subscription plan.
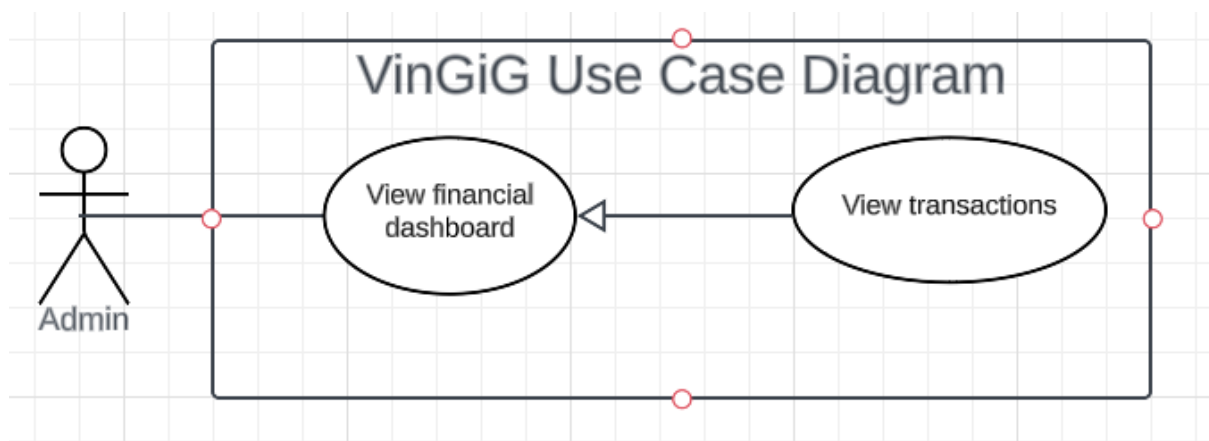
3.2.4.22. <Admin> View transactions

Figure: View transactions

USE CASE-56 SPECIFICATION			
Use-case No.	UC56	Use-case Version	1.0
Use-case Name	View transactions		

Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

Admin

Summary:

 The system allows the user to view transaction history. The user also can view some basic details of each transaction: transactionID, amount and date.

Goal:

 The user is able to view the transactions.

Triggers:

 The admin clicks on the "View transactions" option.

Preconditions:

 The user has to log in successfully as an admin.

Post Conditions:

 A list of transactions has to be displayed by the system.

Main Success Scenario:

No.	Actor's action	System's response

1	The logged user selects the "Financial dashboard" option on the admin homepage.	The system navigates the user to "Financial dashboard" screen
2	The user clicks on "View transactions" option	The system shows a list of transactions. [Exception]

Alternative Scenario:

　　N/A

Exceptions:

No.	Exception	System's response
1	The user hasn't made any purchase.	The system displays a message: "There is no transaction". The user can click on the button "Back to homepage" to navigate back to the homepage.

Relationships:

This use case relates to the "View financial dashboard" functionality in the user system.

Business Rules:

Each transaction is displayed with basic details: transactionID, amount and date (datetime a format: dd/mm/yyyy hh:mm:ss).
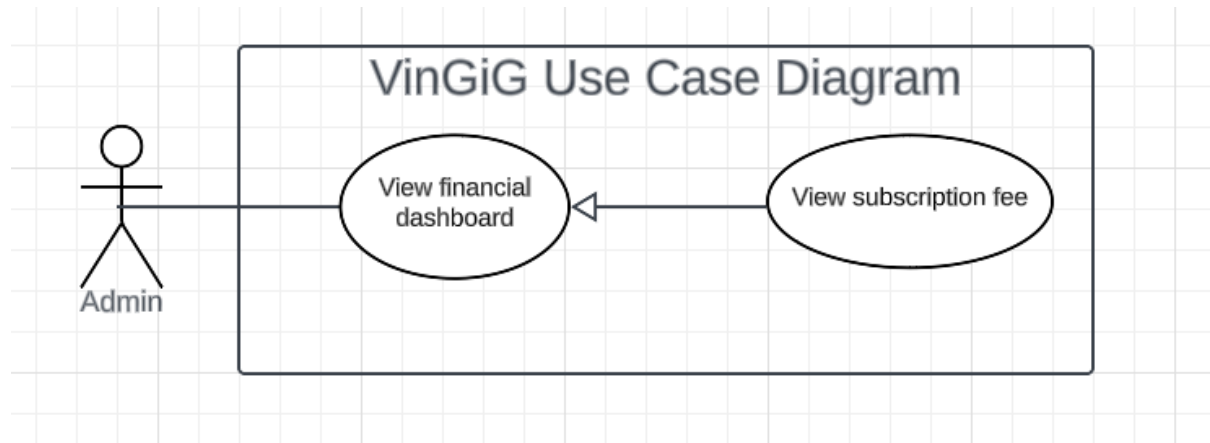
3.2.4.23. <Admin> View subscription fee

Figure: View subscription fee

USE CASE-57 SPECIFICATION			
Use-case No.	UC57	Use-case Version	1.0
Use-case Name	View subscription fee		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

    Admin

Summary:

    The system allows the user to view subscription status of the providers.

Goal:

The user adds a new service successfully.

Triggers:

The admin clicks on the "View subscription fee" option.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

A table of subscriptions is displayed.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged user selects the "Financial dashboard" option on the admin homepage.	The system navigates the user to "Financial dashboard" screen
2	The user clicks on "View subscription fee" option	The system shows the "Subscription fee" screen. The screen displays a list of subscriptions. [Exception]

Alternative Scenario:

N/A

Exceptions:

No.	Exception	System's response
1	The customer doesn't subscribe to any plan after all, or the system failed to load the data.	The system displays a message: "There is no subscription fee". The user can click on the button "Back to homepage" to navigate back to the homepage.

Relationships:

This use case relates to the "View financial dashboard" functionality in the user system.

Business Rules:

For each subscription shown on the screen, its detailed information is provided, including: ID of the provider, ID of the subscribed plan, and total amount of fee and subscription date (datetime a format: dd/mm/yyyy hh:mm:ss).
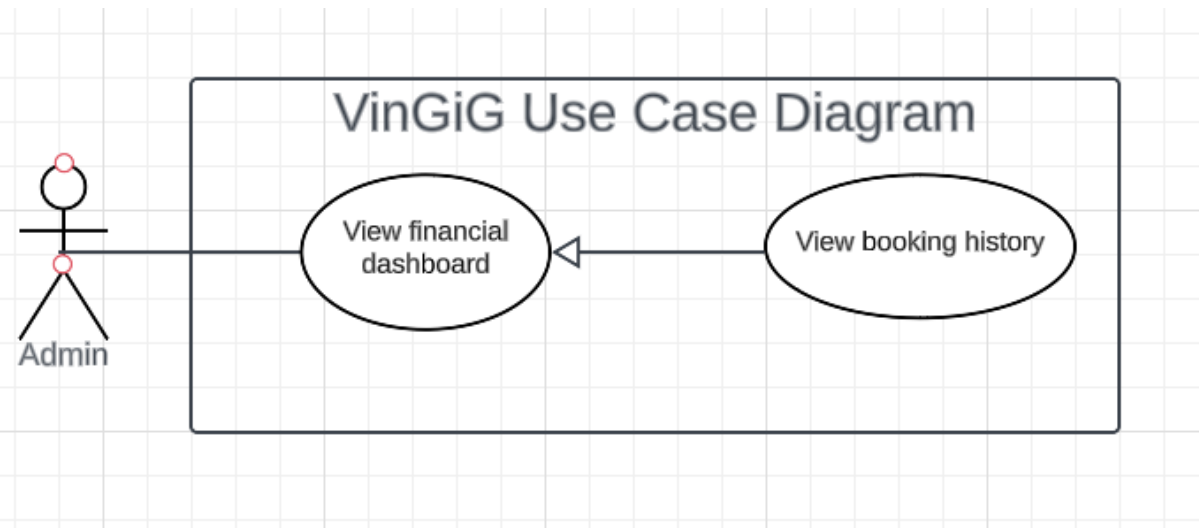
3.2.4.24. <Admin> View booking history

Figure: View booking history

USE CASE-58 SPECIFICATION			
Use-case No.	UC58	Use-case Version	1.0
Use-case Name	View booking history		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

Admin

Summary:

The system allows the user to view booking history.

Goal:

The user can view booking history.

Triggers:

The provider clicks on the "View booking history" option.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

A table of booking history is displayed.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged user selects the "Financial dashboard" option on the admin homepage.	The system navigates the user to "Financial dashboard" screen
2	The user clicks on "View booking history" option	The system shows the "Booking history" screen. The screen displays a list of bookings made by customers. [Exception]

Alternative Scenario:

N/A

Exceptions:

No.	Exception	System's response
1	The customer doesn't make any booking, or the system fails to load the data.	The system displays a message: "There is no subscription fee". The user can click on the button "Back to homepage" to navigate

		back to the homepage.	

Relationships:

This use case relates to the "View financial dashboard" functionality in the user system.

Business Rules:

For each booking, detailed information is provided, including the IDs of: the booking, the customer booked, the provider's service, the customer's rating and the booking date (datetime a format: dd/mm/yyyy hh:mm:ss).

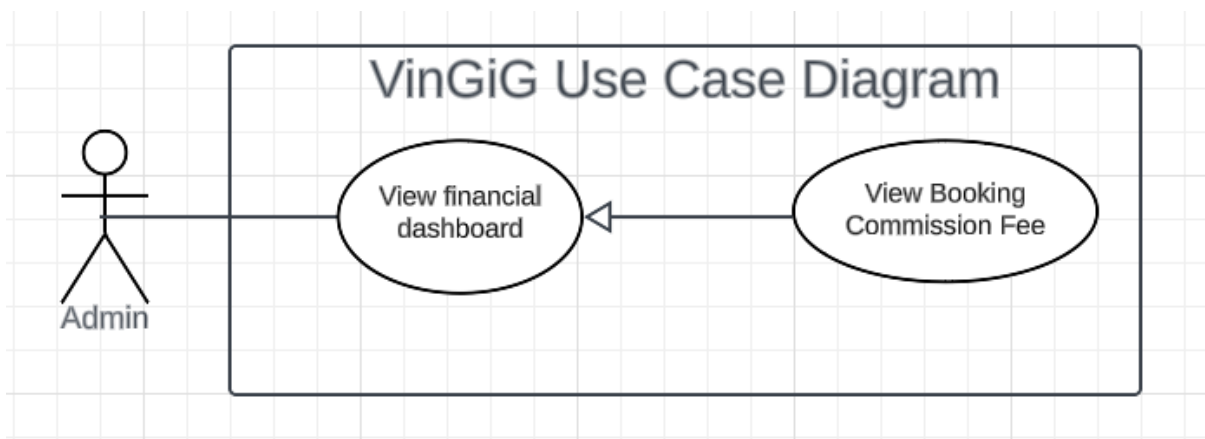## 3.2.4.25. <Admin> View booking commission fee

Figure: View booking commission fee

USE CASE-59 SPECIFICATION			
Use-case No.	UC59	Use-case Version	1.0

Use-case Name	View booking commision fee		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

Admin

Summary:

The system allows the user to view the commission fee for corresponding booking.

Goal:

The admin can view the commission fee of each booking.

Triggers:

The admin clicks on the "View booking commission fee" option.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

Information of each commission fee is successfully displayed.

Main Success Scenario:

No.	Actor's action	System's response
1	The logged user selects the "Financial dashboard" option on the admin homepage.	The system navigates the user to "Financial dashboard" screen
2	The user clicks on "View booking commission fee" option	The system shows the "Booking commission fee" screen. The

		screen displays a list of every commission made by customers.

Alternative Scenario:

    N/A

Exceptions:

No.	Exception	System's response
1	The system fails to retrieve data	The system navigates the user to an error page. The page has a message "requested data is currently unavailable" and a button to navigate back to the homepage.

Relationships:

This use case relates to the "View financial dashboard" functionality in the user system.

Business Rules:

For each commission, detailed information is provided, including the commission's ID, the booking ID, the amount of money and commission's date (datetime format: dd/mm/yyyy hh:mm:ss).
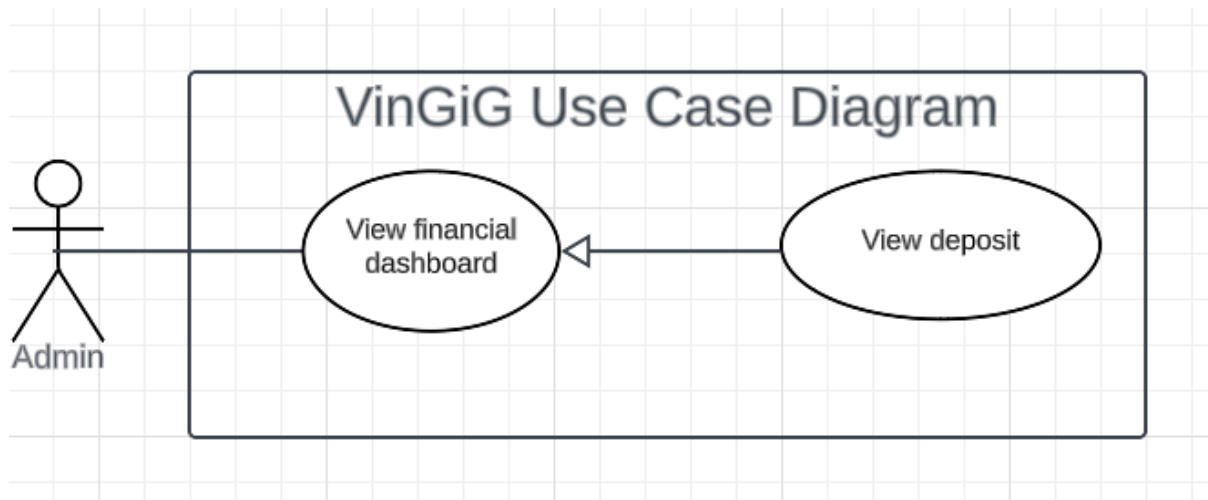
3.2.4.26. <Admin> View deposit

Figure: View deposit

USE CASE-60 SPECIFICATION			
Use-case No.	UC60	Use-case Version	1.0
Use-case Name	View deposit		
Author	Le Nguyen Tien Nhat		
Date	14/06/2023	Priority	Normal

Actor:

   Admin

Summary:

   The system allows the user to view the amount of money prepaid by the providers (deposit).

Goal:

The admin is able to view the deposits' basic details.

Triggers:

The admin wants to view the deposits.

Preconditions:

The user has to log in successfully as an admin.

Post Conditions:

Information of each deposit is successfully displayed

Main Success Scenario:

No.	Actor's action	System's response
1	The logged user selects the "Financial dashboard" option on the admin homepage.	The system navigates the user to "Financial dashboard" screen
2	The user clicks on "View deposit" option	The system shows the "View deposit" screen. The screen displays a list of every deposit paid by customers. [Exception]

Alternative Scenario:

N/A

Exceptions:

No.	Exception	System's response

1	The system fails to retrieve data	The user is navigated to an error homepage. The page has a message "Failed to retrieve the requested data" and a button to navigate to the homepage.

Relationships:

This use case relates to the "View financial dashboard" functionality in the user system.

Business Rules:

For each deposit, detailed information is provided, including the deposit's ID, the provider's ID that paid the deposit, the amount of money, payment method (credit card or online banking) and purchase date (datetime a format: dd/mm/yyyy hh:mm:ss).

4. Non-Functional Requirements

4.1. Usability

- UI fits for all common browsers and sizes.

- A customer/guest can be familiar with the flow of the application after 30 minutes of perusing.

- A provider should feel free to use the application skillfully after 2 hours of training.

- Admin should be able to use the application effectively after 2 days of experience.

4.2. Reliability

- The system runs smoothly without any crash, especially during rush hours.

- In the event of system failure, it takes no more than 1 hour to recover the operation of the application.

- Downtimes occur fewer than 3 times per month.

4.3. Performance

- Response time for any request should take less than 3 seconds for non-booking events.

- For booking-involved events, it should take less than 5 seconds for the latencies.

- The system can handle 1000 simultaneous users.