SECP3204: Software Engineering

Software Requirements Specification (SRS)

Akmal Rental System

Version 1.0

Date:

3rd June 2023

Faculty of Computing

Prepared by: McByte

Revision Page

a. Overview

Describe the content of the current version of SD (see below – the note in red).

b. Target Audience

There are two main targeted audiences which are administrators and investors.

c. Project Team Members

Member Name	Role	Task	Status
MUHAMMAD IMRAN HAKIMI BIN MOHD SHUKRI	Leader	Manage Tenant	Incomplete
AFIF HAZMIE ARSYAD BIN AGUS	Member	Manage Room	Incomplete
MADINA SURAYA BINTI ZHARIN	Member	Manage Finance	Incomplete
NAYLI NABIHAH BINTI JASNI	Member	Authentication and Manage Attendance	Incomplete
NUR SYAMALIA FAIQAH BINTI MOHD KAMAL	Member	Manage Inventory and Manage Investor	Incomplete

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	MUHAMMAD IMRAN HAKIMI BIN MOHD SHUKRI	All Chapter 2.2 are yet to be completed in terms of development, however, the development processes has started	03/06/2023

Table of Contents

Introduction

Akmal Rentals is a room rental service company based in Johor Bahru, serving individuals commuting between Johor and Singapore. They currently operate in Larkin and Taman Nusantara, Gelang Patah, with a total of 115 rooms across 9 floors. However, their existing manual system, involving spreadsheet data entry, phone reminders, and Google Drive storage, is complex and time-consuming. To enhance their services, they require a system that streamlines operations for administrators and investors. The proposed system will enable efficient renter tracking, payment reminders, inventory management, level information, cleaner attendance tracking, sales and profit calculations, and profit sharing with investors.

1.1 **Purpose**

The purpose of the Akmal Room Rental System Documentation (SD) is to provide a comprehensive guide that documents the development process of the system. The SD encompasses key documents such as the System Requirements Specification (SRS), System Design Document (SDD), and System Testing Document (STD) specific to the Akmal Room Rental System.The primary objective of the SD is to serve as a reference for stakeholders involved in the development of the Akmal Room Rental System. This includes administrators and investors who are the target users of the system. The SD ensures clear communication and collaboration among these stakeholders, providing them with the necessary information and guidelines for efficient planning, design, implementation, and testing.

1.2 **Scope**

This System Documentation (SD) pertains to the software product called "Akmal Rental System" and is designed to provide a comprehensive overview of the system development process, including the System Requirements Specification (SRS), System Design Document (SDD), and System Testing Document (STD). Akmal Rental System aims to facilitate the management of room rentals by administrators of Akmal Rental company. The primary purpose of the system is to enable administrators to efficiently handle tenant management, including adding, viewing, updating, and deleting tenant details. The system will also facilitate rent payment reminders for tenants, which occur twice a month, and record the payment transactions with the respective building landlords. Additionally, the system will assist in managing inventory, allowing administrators to track borrowed items and their

availability for tenants' use. The system will further support attendance tracking for hired cleaners, who attend the premises three times per month.

The scope of the Akmal Rental system includes the following:

1. Administrators able to manage tenant details, deposit, room requirement, investor list, cleaner attendance and financial.
2. The rental system will be developed using the agile-driven approach, ensuring that it is scalable and maintainable.
3. The software product will be developed following best practices in software engineering, including coding standards, testing procedures, and quality assurance processes.
4. The software product will be delivered with a user manual and installation guide to assist users in the setup and use of the software product.
5. The software product will be tested using a combination of manual and automated testing methods to ensure that it meets all specified requirements and performs as expected.

1.3 **Definitions, Acronyms and Abbreviation**

Definitions of all terms, acronyms and abbreviations used are to be defined here.

Term	**Definition**
Akmal Rental System	A software system designed to manage the room rental services provided by Akmal Rentals, a company based in Johor Bahru
SD	System Documentation - Comprehensive collection of documents that provide detailed information about the system
SDD	System Design Document - Outlines the detailed design specifications and architecture of the system
STD	System Testing Document - Contains the test cases, test scenarios, and procedures for validating and verifying the functionality, performance, and quality of the system.
SRS	System Requirement Specification - Defines the functional and

	non-functional requirements of the system.
Stakeholders	Refer to individuals or entities with an interest or involvement in the system.
Administrator	An individual who is responsible for managing and overseeing the operations of a system or organization.

1.4 **References**

Title	Date	Publishing Organisation	Sources
System Requirement	21 May 2023	Akmal Rentals	Akmal Rentals
Sales AR 2023	18 May 2023	Akmal Rentals	Akmal Rentals

1.5 **Overview**

The Sequence Diagram (SD) provides a visual representation of the interactions between different entities, controllers, and models within a system. It depicts the flow of messages and the order of actions in a specific scenario.

The SD is organized as follows:
Actors: This section identifies the participants or actors involved in the sequence diagram. These can be users, system components, external systems, or any other entities interacting with the system

Message Flow: This section illustrates the sequence of messages exchanged between the participants. Each message represents a specific action or communication between participants, indicating the direction of the message flow.

Lifelines: This section includes the lifelines of each participant, representing their existence throughout the sequence diagram. Lifelines are depicted as vertical lines with the participant's name at the top. Activation and Deactivation: This section shows the activation and deactivation of participants during the message flow. An activation represents when a participant is actively engaged in performing an action, while deactivation indicates when the participant has completed their task or is temporarily inactive.

The SD serves as a documentation and communication tool for understanding the interactions between different components of a system. It helps stakeholders visualize the message flow and understand the order of operations in a particular scenario. The SD should be consulted throughout the software development process to ensure accurate implementation and testing of the system's functionality.

2. Specific Requirements

2.1 **User characteristics**.

2.1.1 Admin

- Able to login into the system
- Manage tenants by adding, viewing, updating and deleting tenants information.
- Manage rooms by adding rooms, viewing rooms, and updating room status or details.
- Manage inventory by adding, viewing, updating and deleting the inventory list.
- Manage attendance by adding the cleaners attendance, view and delete attendance for each location.
- Manage investor by adding, viewing, editing and deleting the investors

2.1.2 Investor

- Able to login into the system
- The investors are expected to only view all the modules except room, cleaner attendance and inventory details.

2.2 **System Features**

Akmal Rental is a software system that operates on web applications in localhost. The system helps in managing the rental for tenants according to the rooms. It also provides calculation of the expense and profit.

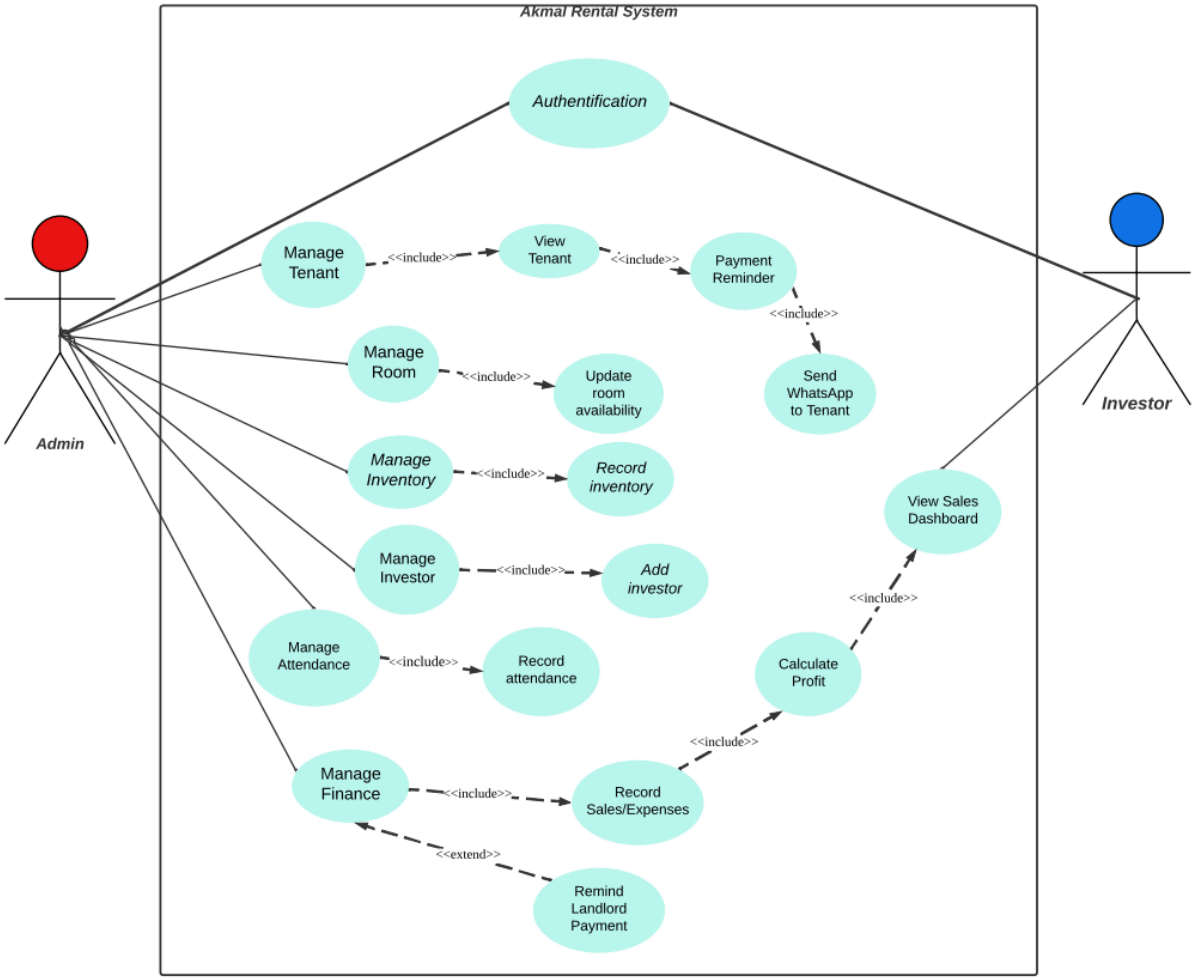

Figure 2.2: Use Case Diagram for Akmal Rental System

Table 2.2: Description of Module and Functions for Akmal Rental System

Module	***Function***	***Description***
Authentification Module	UC01 – Login	This use case allows users to log into the system as the stakeholders
Room Module	UC02 - CRUD Room	This use case allow user(Admin) to Add, Update, Delete and View the detail of room based on locations

Tenants Module	UC03 - Tenant	This use case allows Admin to Add, Update, Delete and View Tenants information and record. See payment reminder and send a WhatsApp message to the tenant using an external link.
Inventory Module	UC04 – Inventory	This use case allows admins to add, update, delete and view the inventory of each floor within 2 different locations.
Investor Module	UC05 – Investor	This use case authorizes admins to add, update, delete and view the investors.
Attendance Module	UC06 - Attendance	This use case allows admin to add, delete and view attendance of the cleaner for each location
Finance Module	UC07 - Finance	This use case allows the admin to record their expenses and investors view the sales summary.

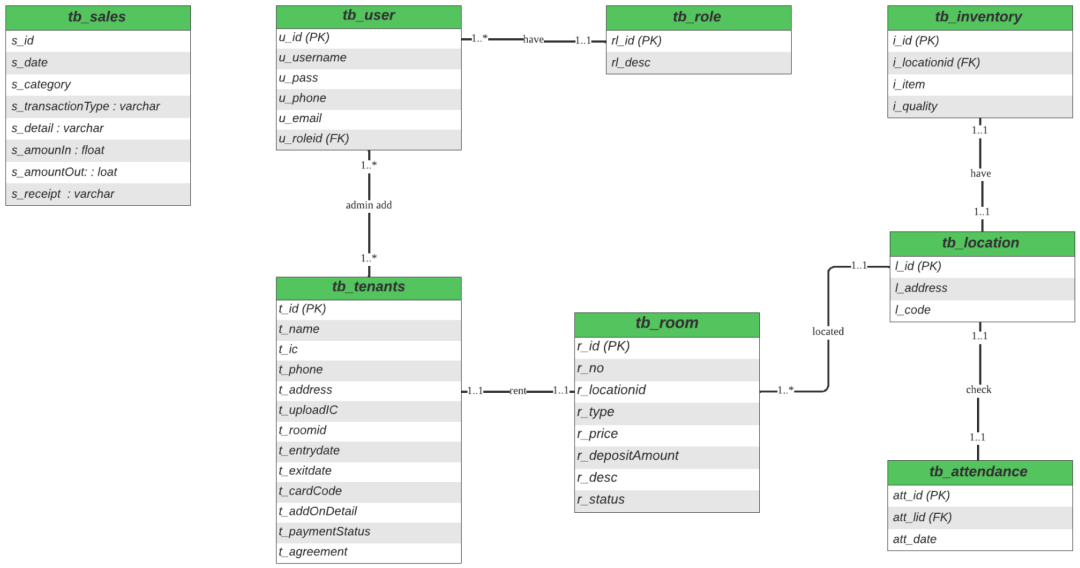

Figure 2.2: ERD for Akmal Rental

2.2.1 UC001: Use Case Authentication

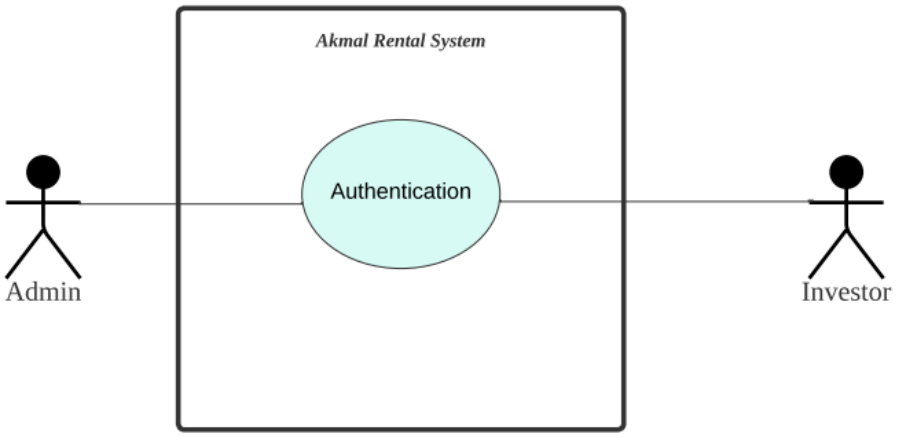

Table 2.2.1: Use Case Description for Authentication

Use case: Authentication
ID: UC1
Actors: Admin and investor
Preconditions: 1. The system will request the actors to register information about user id, username, password, name, contact information, email and role.
Flow of events: 1. Actors need to enter username and password. 2. The system validates the actor's login and password and logs him/her in.
Postconditions:
Exception flow: 1. If the actor inputs an invalid login and/or password 1.2. The system shows an error message.

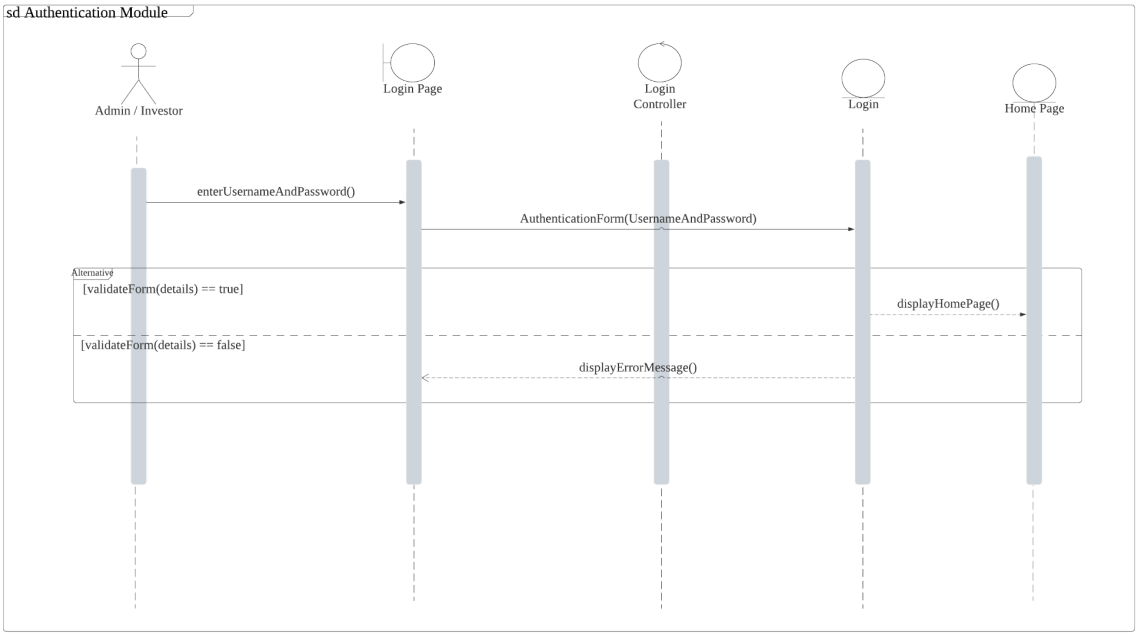

Figure 2.2.1.1: Sequence Diagram for Login

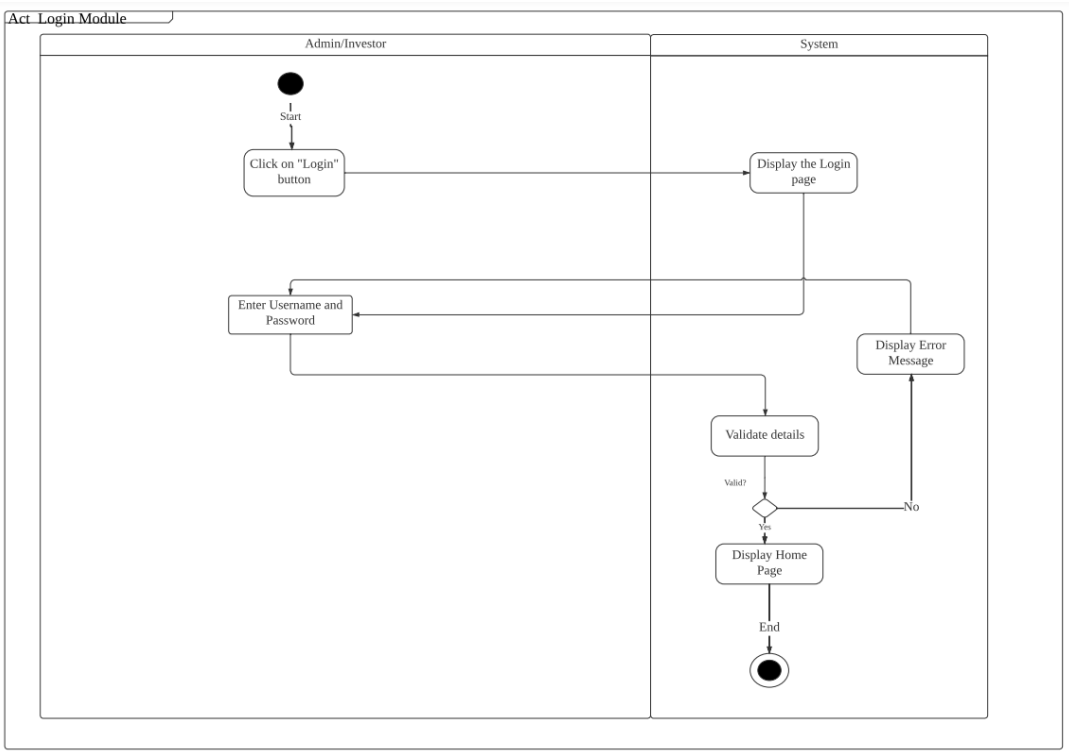

Figure 2.2.1.2: Activity Diagram for Login

2.2.2 UC002: Use Case Room

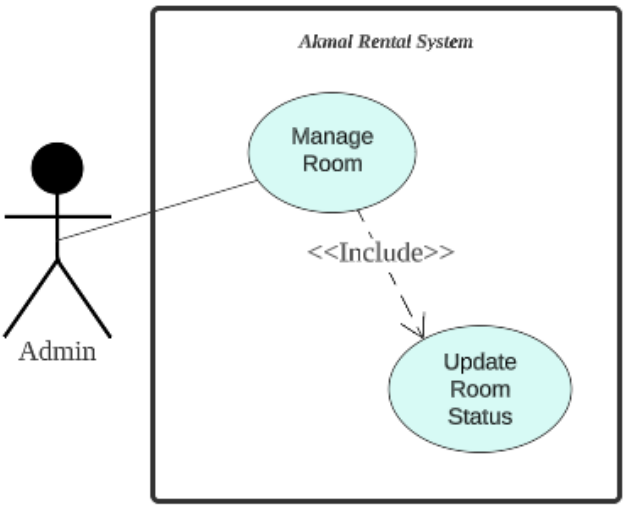

Table 2.2.2: Use Case Description for Room

Use case: Room
ID: UC2
Actors: Admin
Preconditions: 1. Admin login and validate into the system. 2. Locations is available in the system
Flow of events: 1. Admin Login and validate into the system 2. Admin click on “Room” button on the nav-bar 3. Admin view list of room. 4. Admin can perform the following task a. Create/Add new room into the system b. View details of each room c. Update the information of room d. Delete unwanted room 5. System will update the change made into the room detail.
Postconditions: 1. Changes made into the room table will be updated according to action performed by the admin.
Exception flow:

1. If room no already exist according to locations
    a. System display error message.
2. If information is not sufficient
    a. System display error message.

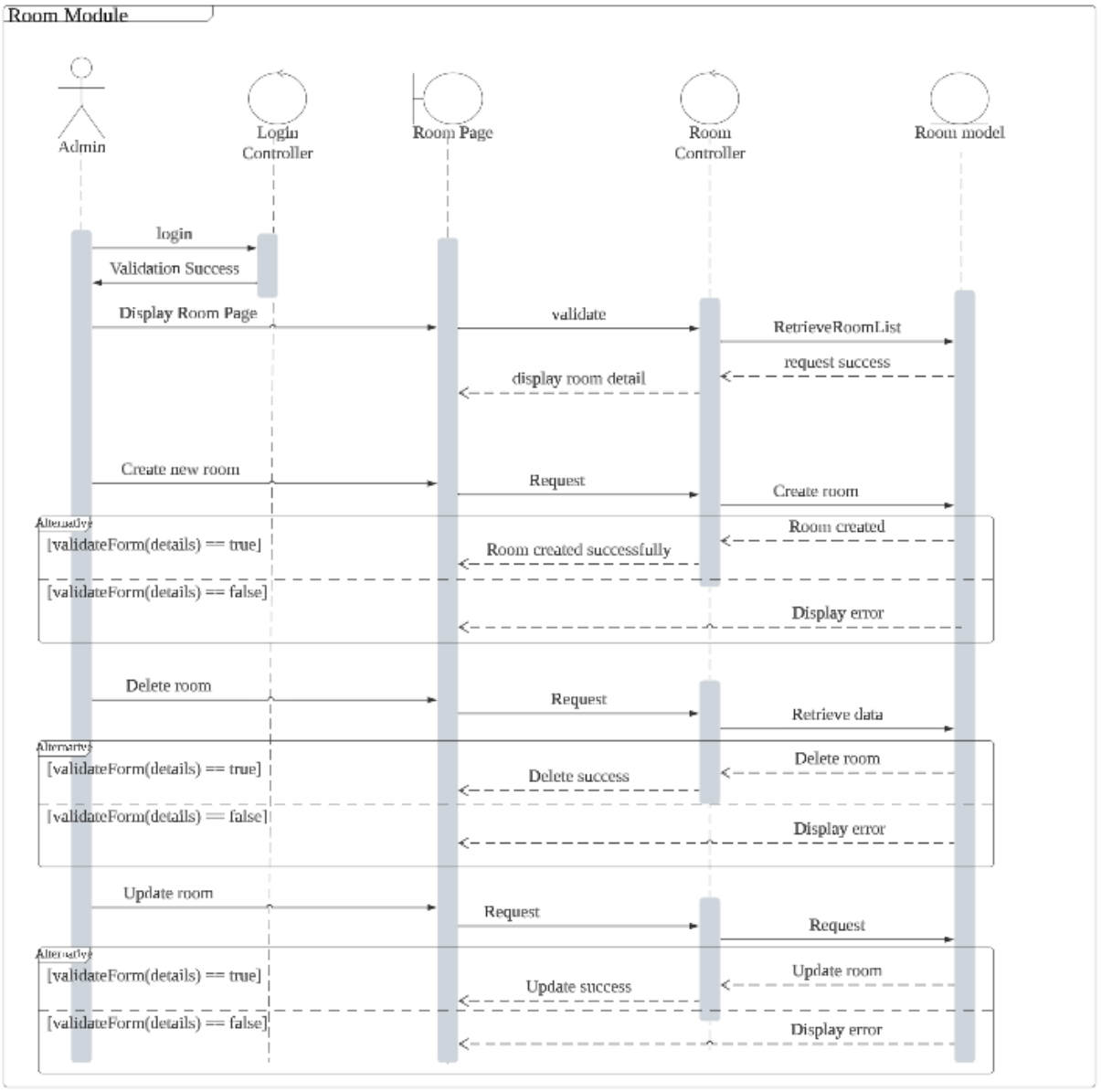

Figure 2.2.2.1: Sequence Diagram for Room

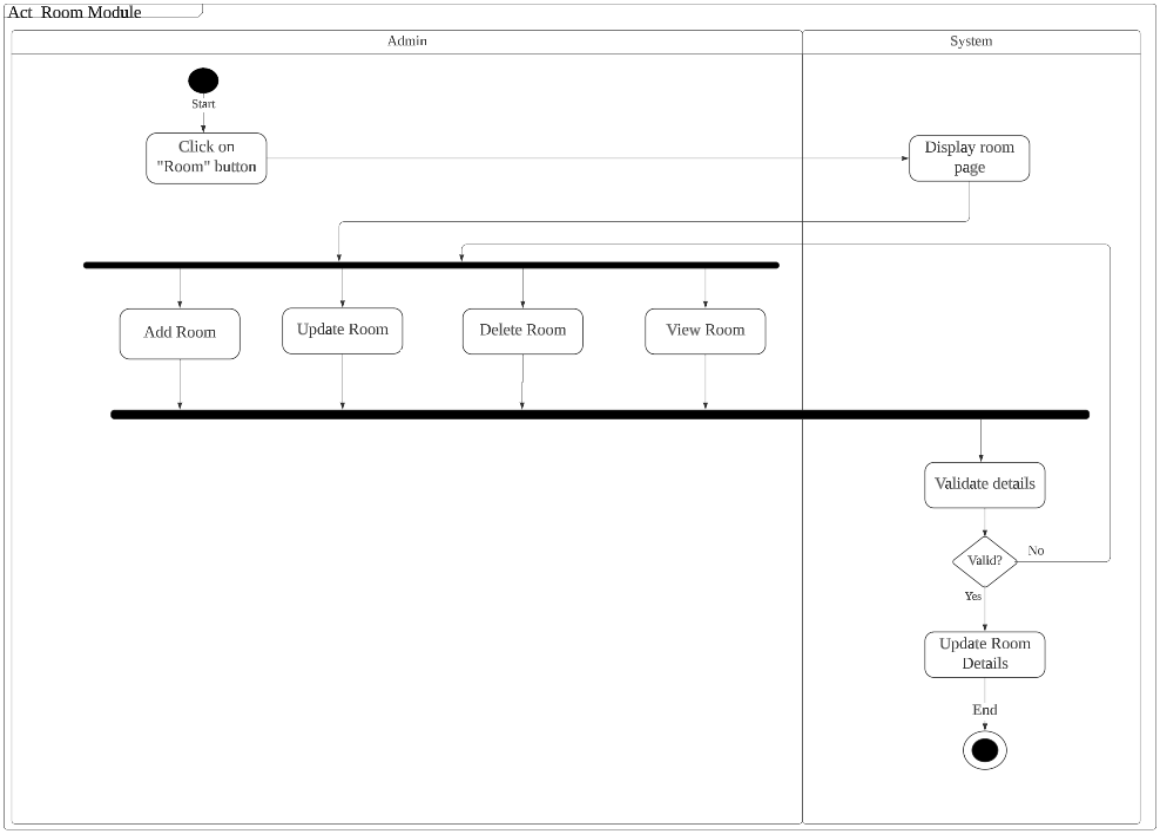

Figure 2.2.2.2: Activity Diagram for Room

2.2.3 UC003: Use Case Tenants

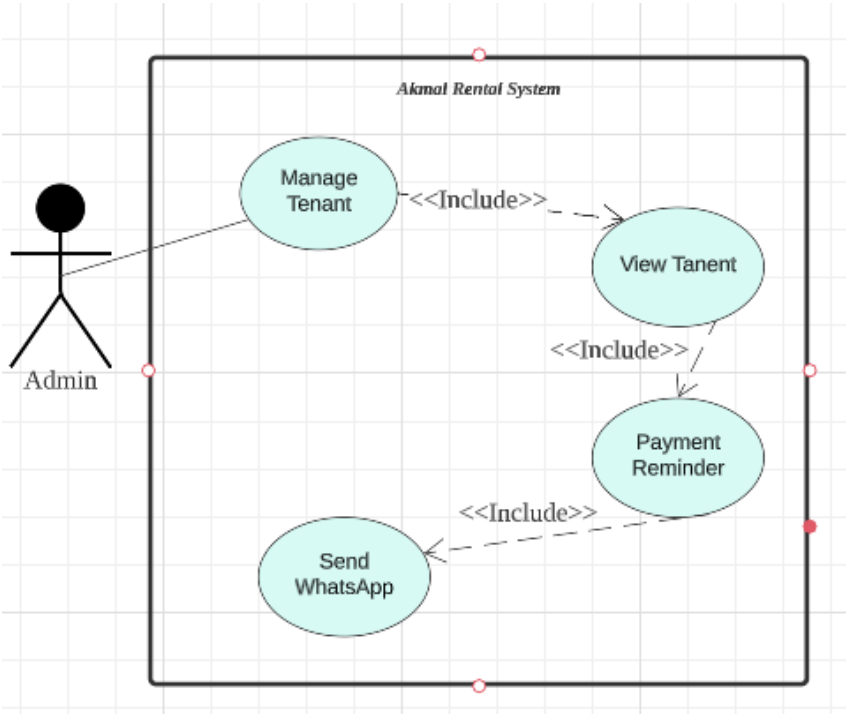

Table 2.2.3: Use Case Description for Tenants

Use case: Tenants
ID: UC3
Actors: Admin
Preconditions: 1. Admin login and validate into the system. 2. Room must be available to select to ensure room availability.
Flow of events: 1. Admin login into the system. 2. Admin click the Tenant button on the sidebar from the Dashboard. 3. Admin view tenant records and lists. 4. From the tenant view, admin can perform: a. Add tenant information with preferred available room. b. Edit tenant information and preferred room. c. Delete tenant information.

5. Admin view reminders regarding tenant payment. 6. From the reminder, the admin sends a WhatsApp message to the tenant.
Postconditions: 1. Changes made from the tenant view will be updated in the table based on actions performed by the admin. 2. The reminder is deleted after payment has been made.
Exception flow: 1. If no room is available for rent: 1.1. Tenants cannot be created. 2. If required field is not filled: 2.1. System will display an error message.

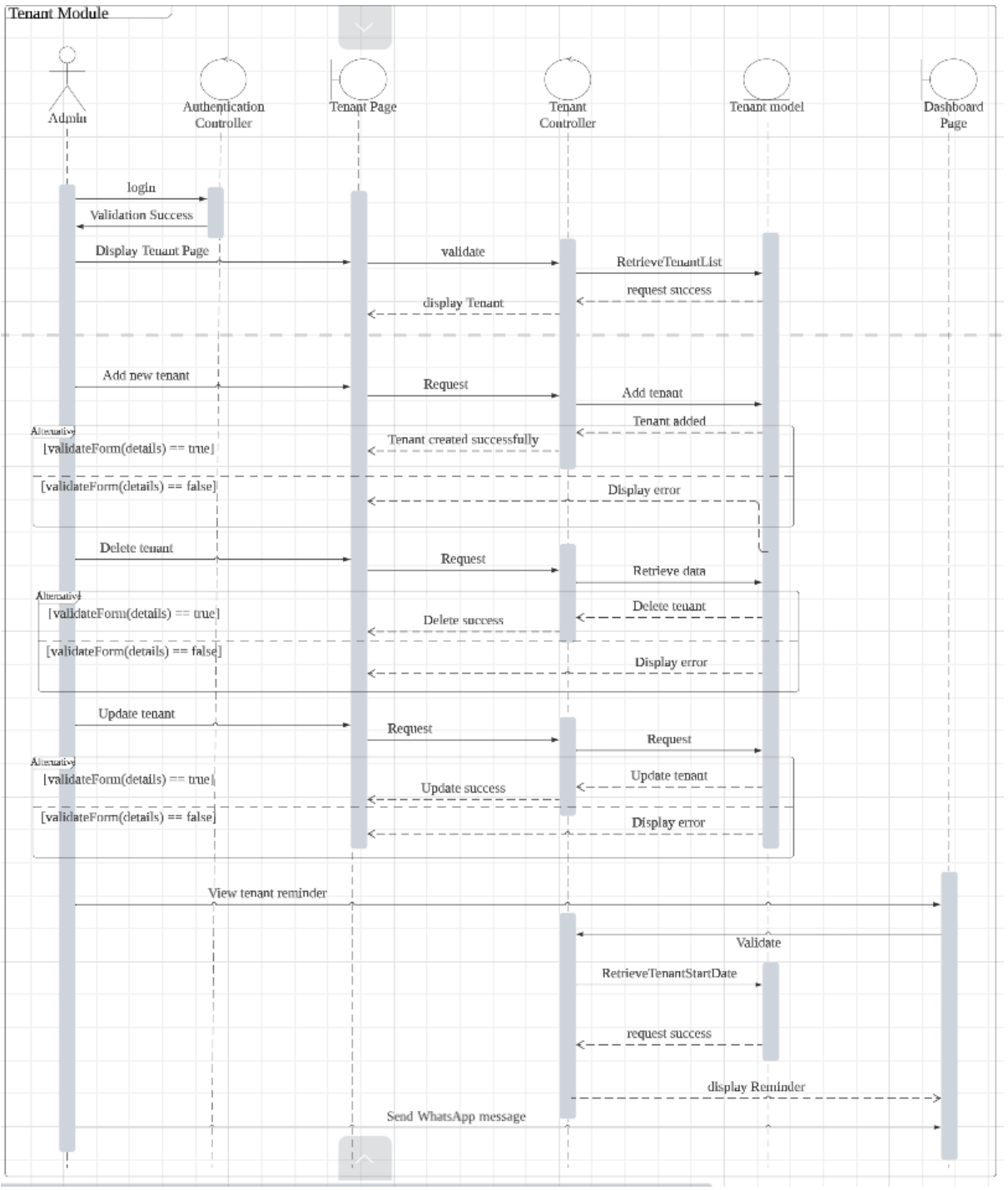

Figure 2.2.3.1: Sequence Diagram for Module Tenant

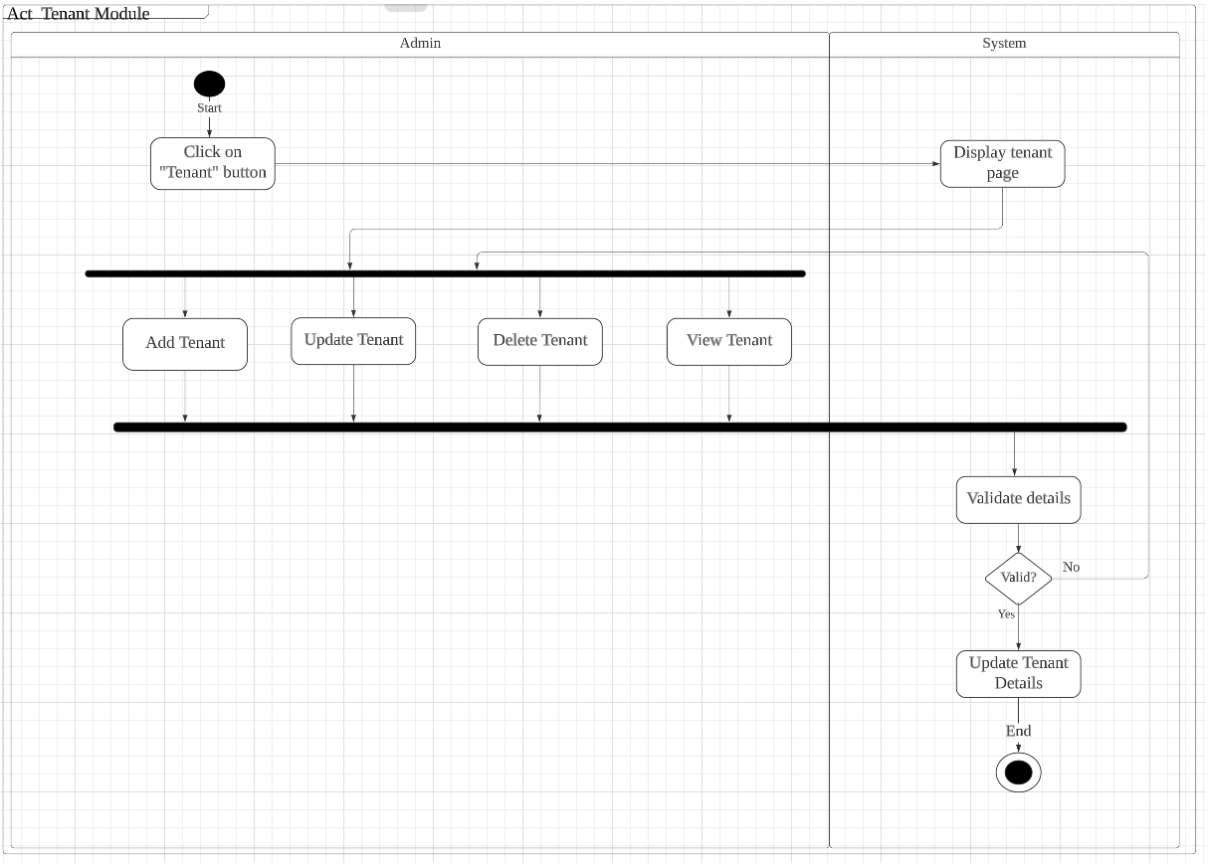

Figure 2.2.3.2: Activity Diagram for Managing Tenant

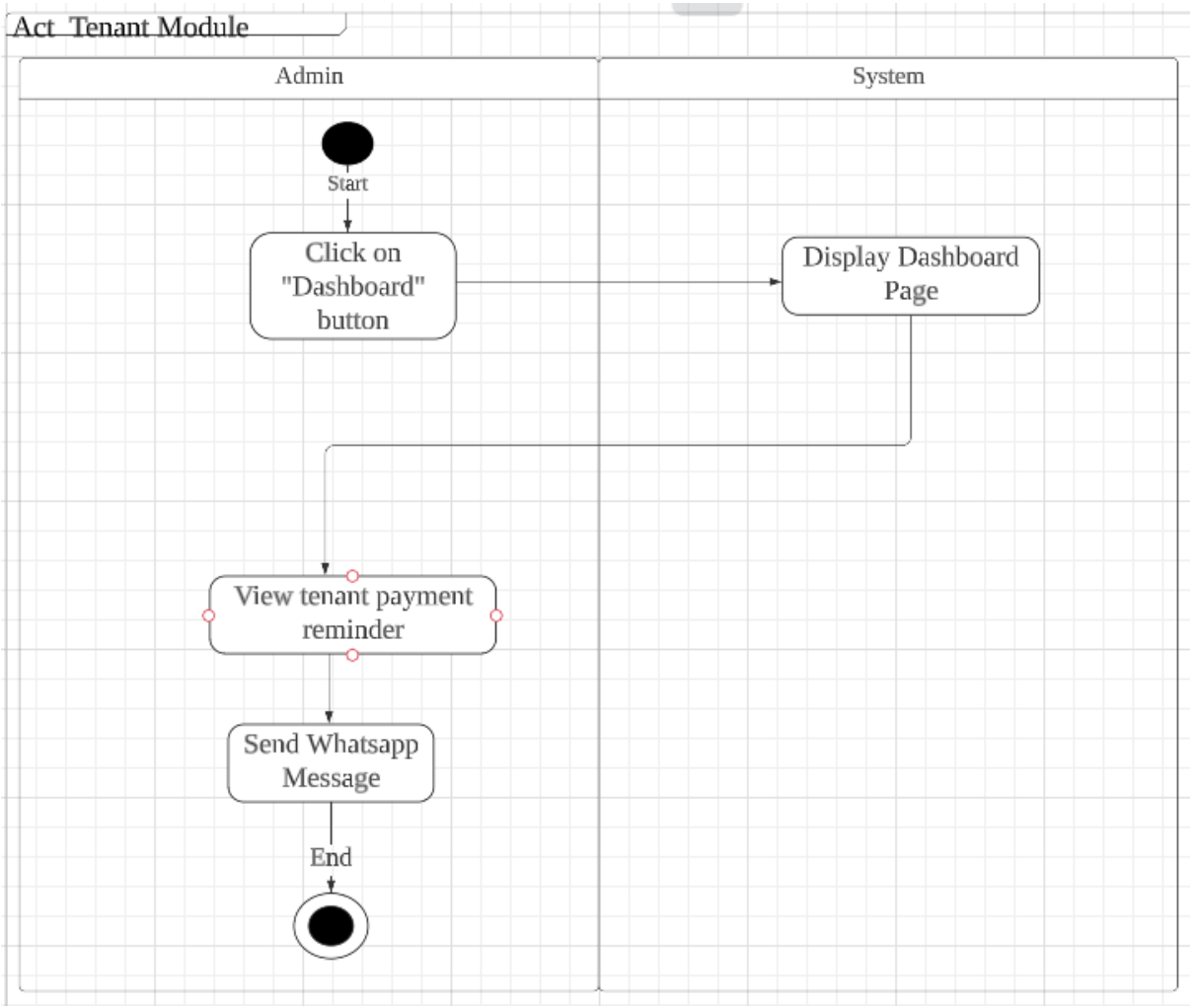

Figure 2.2.3.3: Activity Diagram for Payment Reminder.

2.2.4 UC004: Use Case Inventory

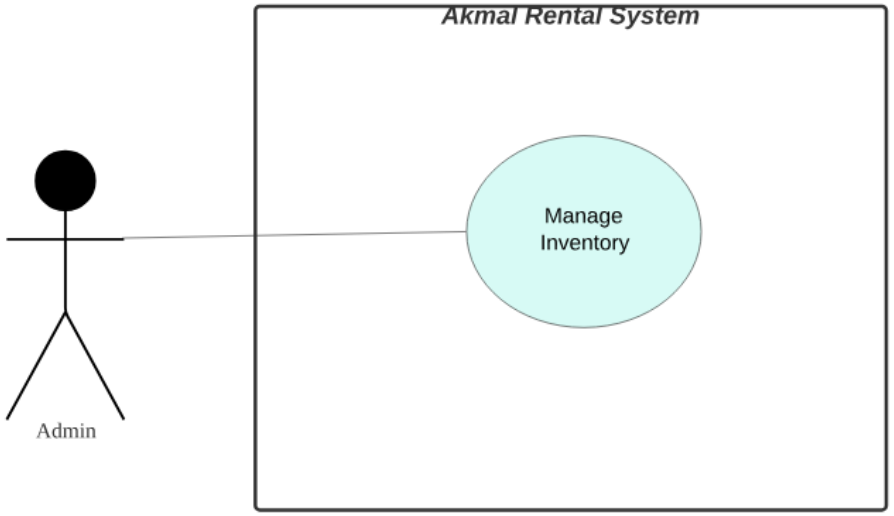

Table 2.2.4: Use Case Description for Inventory

Use case: Inventory
ID: UC4
Actors: Admin
Preconditions: 1. Admin needs to log in and validate into the system.
Flow of events: 1. Admin login and validate into the system. 2. Admin choose the inventory option in the navbar. 3. Admin perform the following actions: a. Add a new item and fill in all the information. b. Update the inventory and supply of the existing item. c. Delete the existing item which will be removed from the store of each location. d. View all inventory lists for every location. 4. After performing any action, the system will update the inventory records and display the new lists.
Postconditions:

1. The inventory records are updated based on the actions performed by the admin.
Exception flow (if any): 1. If the details are incorrectly fill up: 1.1. System display error message.

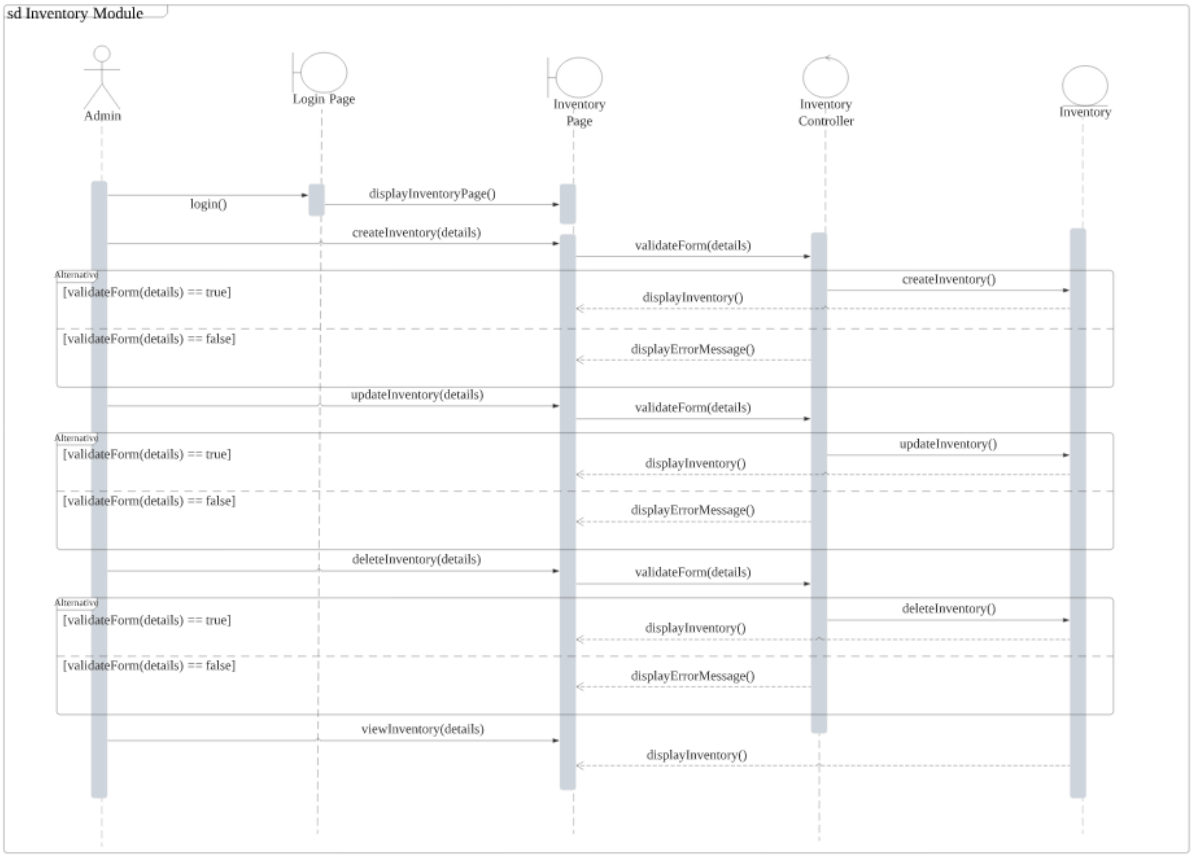

Figure 2.2.4.1: Sequence Diagram for Inventory

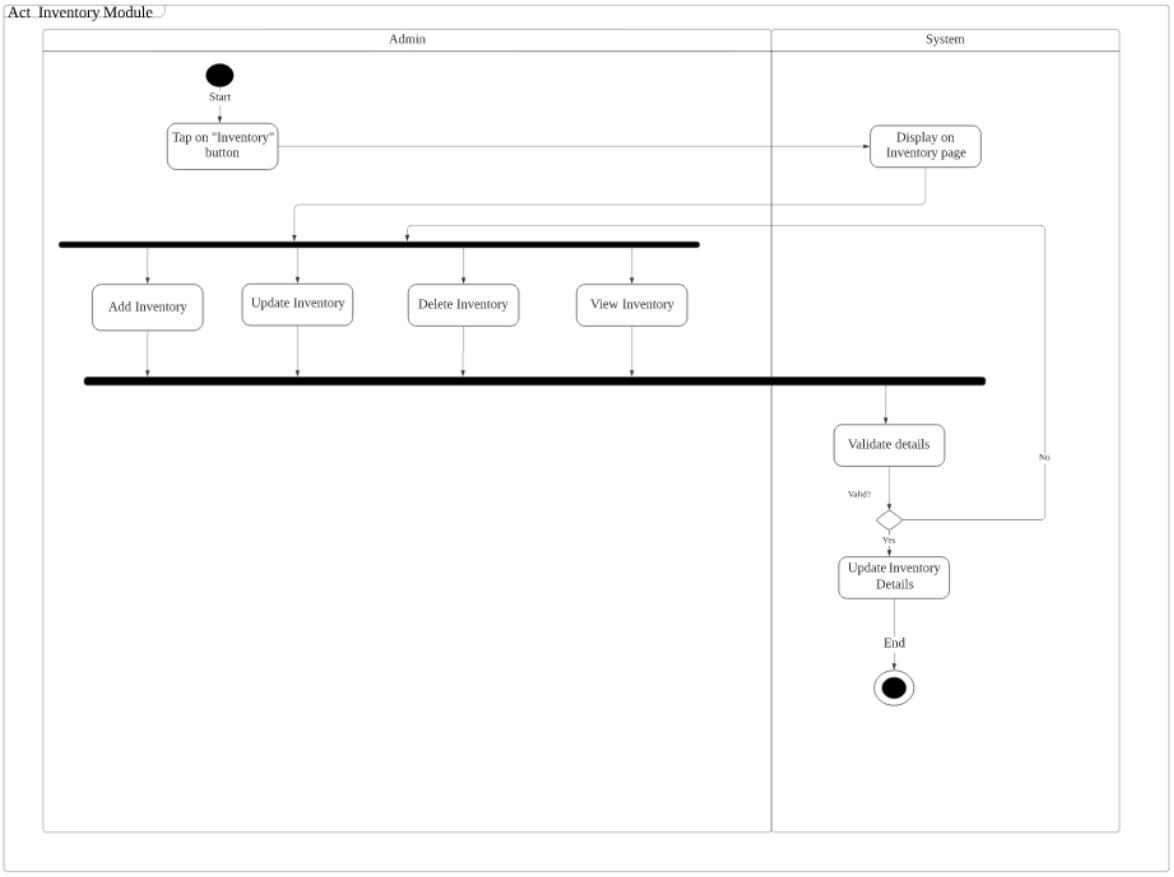

Figure 2.2.4.2: Activity Diagram for Inventory

2.2.5 UC005: Use Case Investor

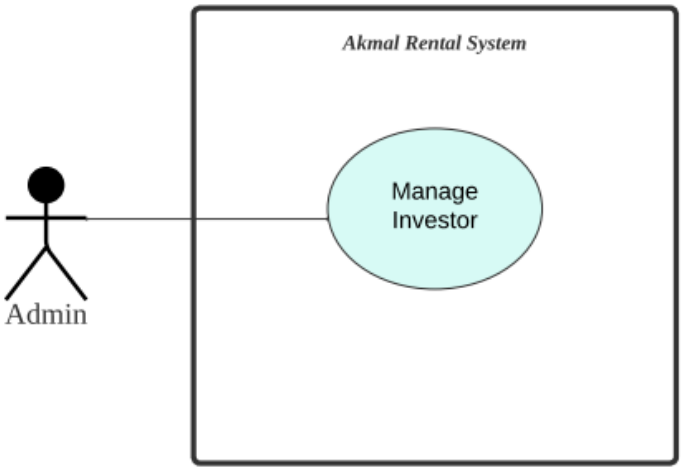

Table 2.2.5: Use Case Description for Investor

Use case: Investor
ID: UC5
Actors: Admin
Preconditions: 1. The admin has logged into the Akmal Rental system.
Flow of events: 1. Admin must log in and verify in the system. 2. Admin select the investor option in the navbar. 3. Admin execute the subsequent operations: a. Admin can create a new investor and fill up the required details. b. Existing investors can be updated. c. Admin can delete the existing investors. d. All the investors' details can be monitored in the list of tables. 4. The system will update the investor records and provide the new listings after any action is taken.
Postconditions: 1. Admin actions are taken into account when updating the investor records.
Exception flow (if any): 1. If the details are incorrectly fill up: 1.2. System display error message.

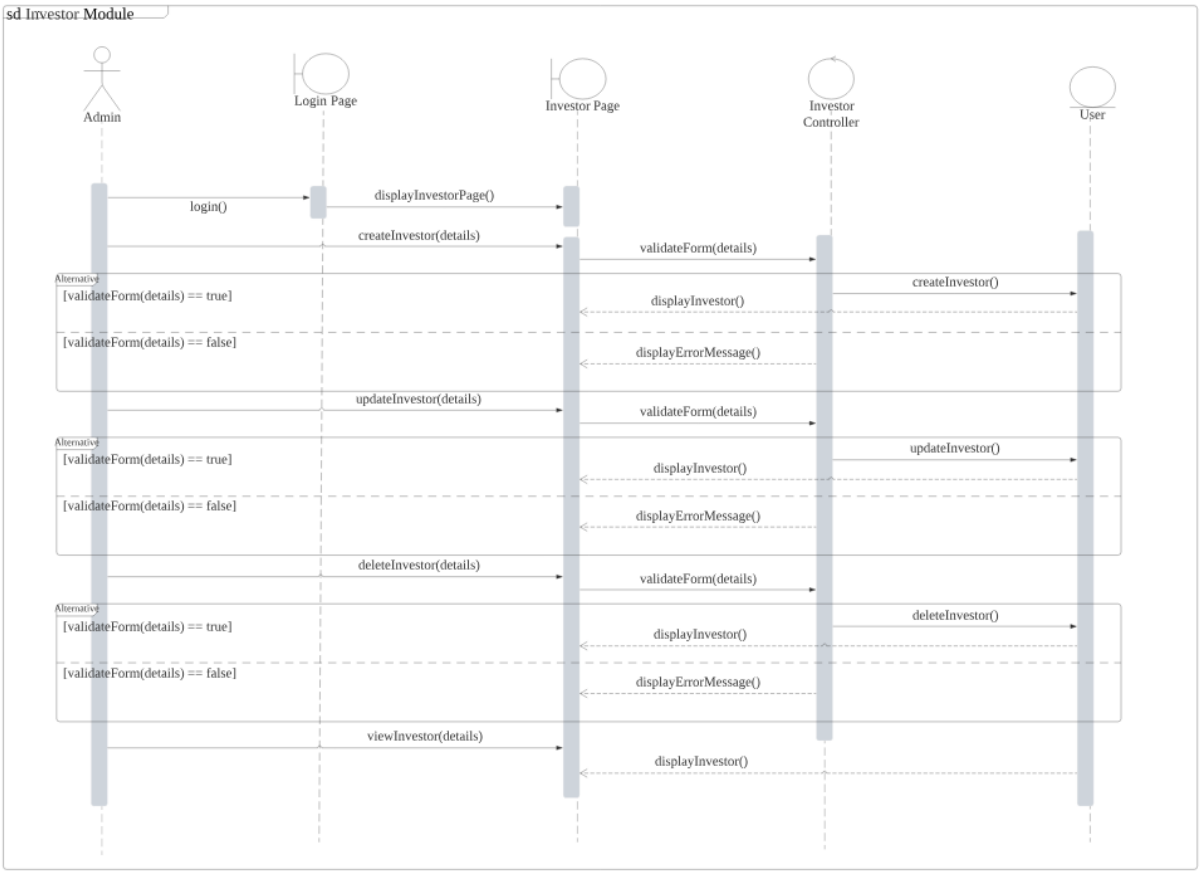

Figure 2.2.5.1: Sequence Diagram for Investor

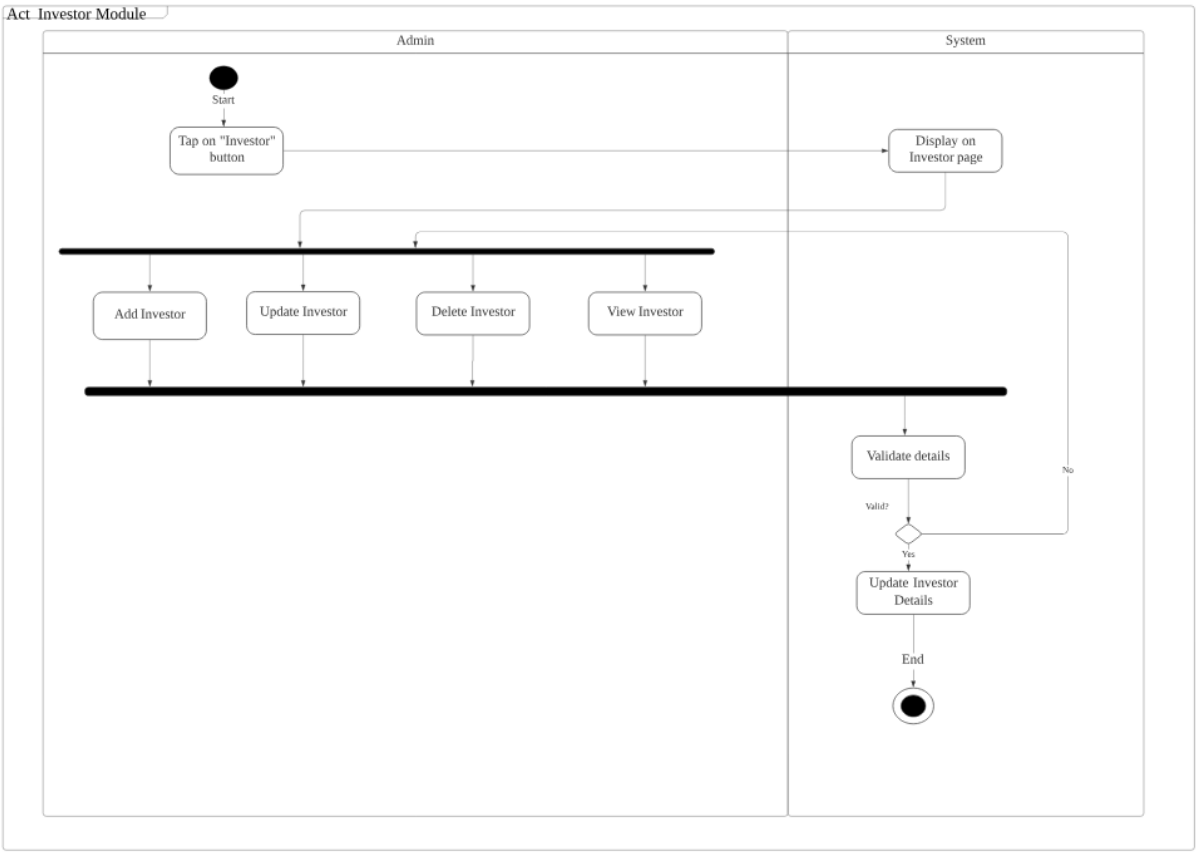

Figure 2.2.5.2: Activity Diagram for Investor

2.2.6 UC006: Use Case Attendance

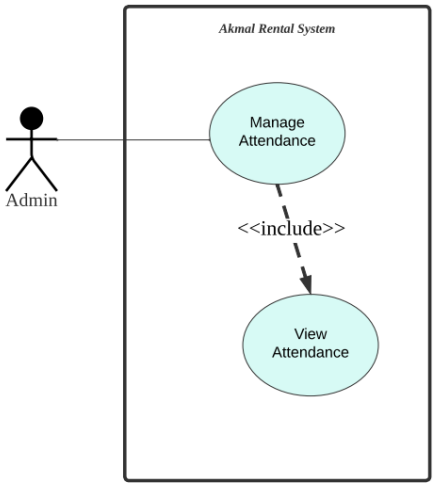

Table 2.2.6: Use Case Description for Attendance

Use case: Attendance
ID: UC6
Actors: Admin
Preconditions: 1. Admin should be logged into the system.
Flow of events: 1. Admin needs to log in and with verified authentication. 2. Admin clicked on the ‘Attendance’ option at the nav-bar. 3. Admin will be directed to the Attendance page. 4. Admin can perform these operations: a. Admin is able to record new attendance of cleaners by clicking on the ‘Present’ button. b. List of attendance records can be viewed. c. Admin can delete the attendance record. 5. The system will display the attendance records according to month and year.
Postconditions: 1. Admin and investor will be able to view the attendance record’s dashboard
Exception flow (if any):

1. If there were more than 3 attendance per month for each location

    1.2. System will display an error message

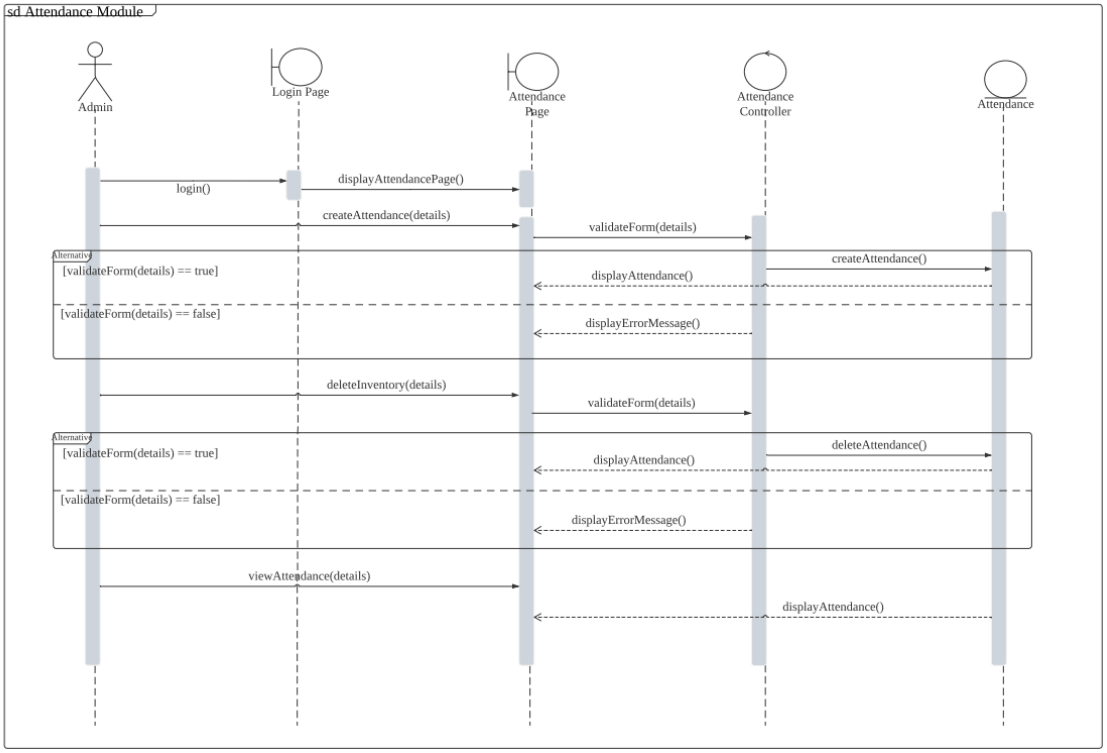

Figure 2.2.6.1: Sequence Diagram for Attendance

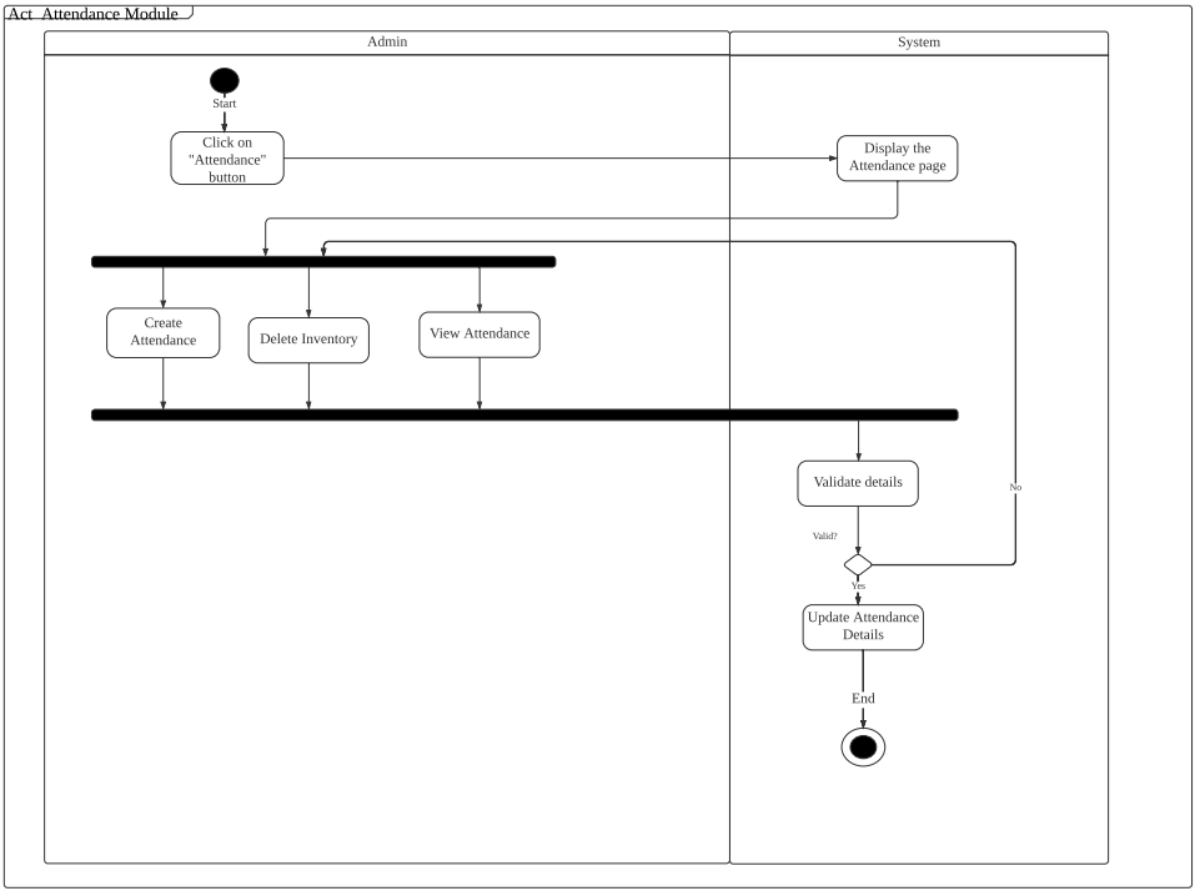

Figure 2.2.6.2: Activity Diagram for Attendance

2.2.7 UC007: Use Case Finance

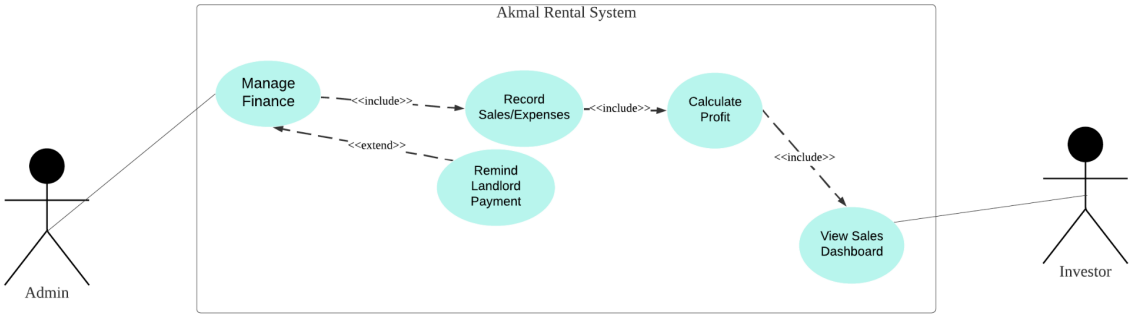

Table 2.2.6: Use Case Description for Finance

Use case: Finance
ID: UC7
Actors: Admin
Preconditions: 2. Admin login into the system
Flow of events: 5. Admin must log in and verify in the system. 6. Admin select sales option in the navbar. 7. Admin select sales category from sales sub-navbar. 6. Admin execute the subsequent operations: a. Admin can create a new sales record by filling up the required details. b. Existing sales records can be updated. c. Admin can delete the sales record. 7. The system will calculate the records according to month and year. 8. The system will determine profit gain or loss. 9. The system will generate a sales dashboard for investors and admin to view.
Postconditions: 2. Admin and investor can view the sales dashboard
Exception flow (if any):

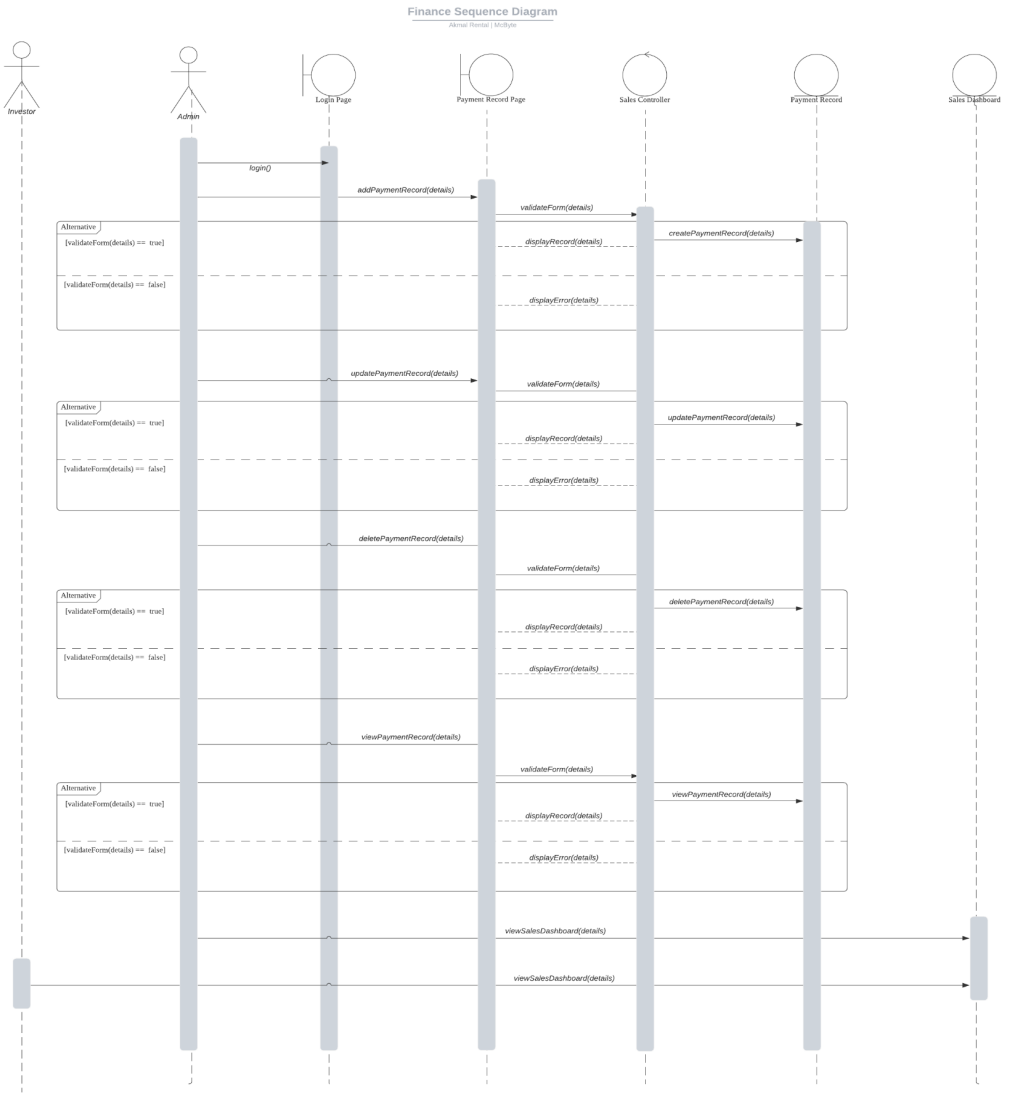

Figure 2.2.7.1: Sequence Diagram for Finance

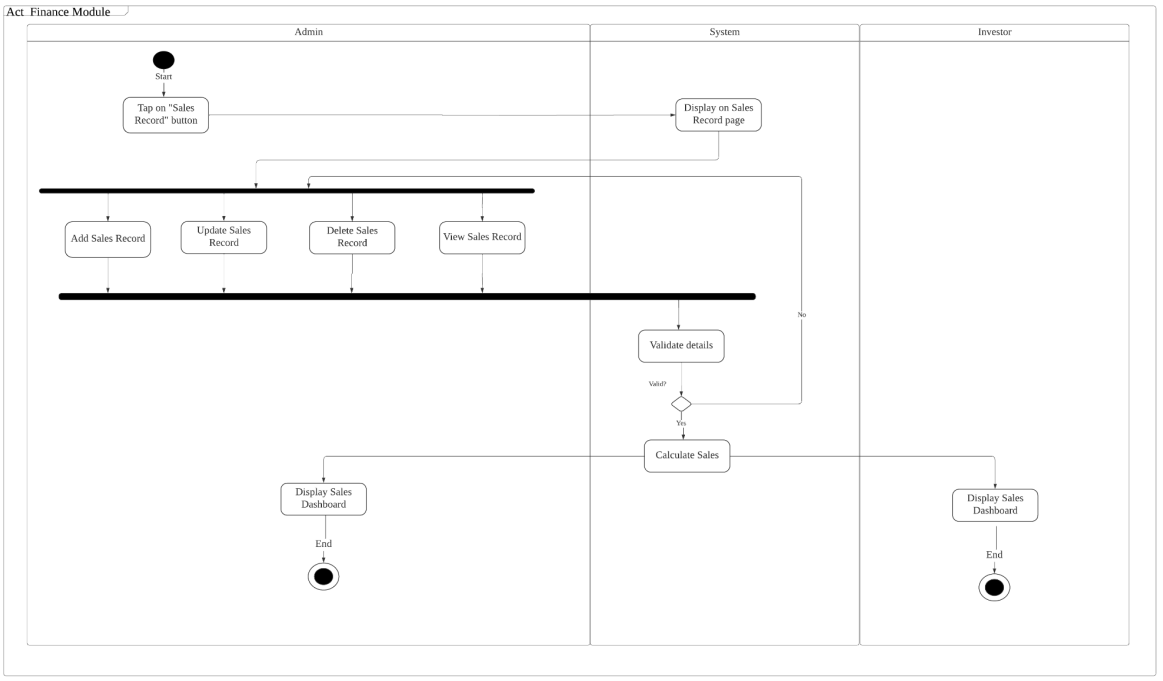

Figure 2.2.7.2: Activity Diagram for Finance

2.3 **Software System Attributes, Performance and Other Requirements**

The attributes of a software system define the overall qualities or characteristics of the software. These characteristics serve as the foundation for the software and include the following:

- Usability: The system should be easy to use and navigate.
- Reliability: The system should be reliable and available 24/7.
- Maintainability: The system should be easy to maintain and update.
- Portability: The system should be portable and easy to deploy on different platforms.
- Compatibility: The system should be compatible with other software systems that are used by customers and employees.

The system's ability to respond to user requests and handle data in a timely manner is defined by performance requirements. Among these requirements are the following:

- Response Time: The time it takes the system to respond to a user's request for a room rental.
- Throughput: This term refers to the system's ability to process a certain number of room rental requests in a given amount of time.
- Capacity: This is the maximum number of users the system can support or the amount of rental data it can effectively handle.
- Availability: This is the percentage of time that the system is operational and accessible to users looking for room rentals.

Other requirements are non-functional requirements that do not fall into one of the categories of software system attributes or performance. Among these requirements are the following:

- Security: Keeping the system and its data safe from unauthorized access and malicious attacks in order to protect the privacy and integrity of user data.
- Safety: The system should be designed to prioritize user safety while minimizing potential harm to users or the environment.

- Legal and Regulatory: Adherence to applicable laws, regulations, and industry standards governing Singapore room rental services.
- Environmental: Assessing and mitigating the system's environmental impact, promoting sustainable practices, and considering energy efficiency measures as appropriate.

2.4 **Design Constraints**

Environmental constraints: The Akmal Rental system should be created to function in that environment. Additionally, it must adhere to any legal or regulatory requirements that are particular to the sector or area of operation. To illustrate, "The system must be run in localhost".

Hardware constraints: The Akmal Rental system should be compatible with the existing hardware infrastructure, including servers, storage devices, and network equipment. Such as, "The system must be designed to run on a minimum of 8GB of RAM."

Security constraints: Strong security measures, such as encryption, access controls, and secure communication protocols, should be implemented by the Akmal Rental system to safeguard user data. For example, "The system must be designed to be authorized by registered users only."

Compatibility constraints: These restrictions deal with the Akmal Rental System's compatibility requirements, such as support for particular software or hardware. For instance, "The system must be designed to be compatible with Windows 11 operating system."

Performance constraints: The response time, throughput, and system availability standards for the Akmal Rental system must be met. For example, "The system must be built to support multiple users operating simultaneously with a response time of under 10 seconds."