Software Requirements Specification

for

Waste Tracker System

Prepared by

Mission Devnath   ASH2125002M
Jarin Thasnin Ahmed BKH2125006F

Institute of Information Technology
Noakhali Science and Technology University

05.03.2024

Table of Contents

List of Figure

List of Tables

1. Introduction

The policy, scope, references, and summary of Software Requirements Specification (SRS) are all included in the SRS introduction. By presenting the problem statement in detail, the purpose of this document is to collect, evaluate, and provide a deeper understanding of the whole **"Waste Tracker System"**. The main services of the **"Waste Tracker System"** are tracking waste and assigning cleaners to clean.

1.1 Problem Statement

This system is built for a university. This system's primary actors are students, teachers, and staff. The secondary actor is the admin(authority). For cleaning purposes, another actor is the cleaner. For modifying any changes another actor is the developer. First of all, every user (primary actor) must create an account to use the system. When a user creates an account system can authenticate the user and create a new account. if a user has an account in this system, then he will be able to access control or log in to the system. If he enters a valid username and password then he enters the system otherwise it shows invalid authentication and show message for forgot password. After login primary actor can post pictures of wastage from any place inside the campus area, with this picture they need to send the proper location of this picture. For every post when it is done admin will get a notification for this post. According to the post, they will assign a cleaner. For getting updates they will track cleaners. Authority will assign tasks for cleaners who are assigned. According to the task, the cleaner performs the task. if the task is complete then they will report to admin that the task is complete. After completing the task admin will approve the post that it is cleared. if the task is not complete primary actors can comment on the relevant post and request for a cleaner from the admin.

1.2 Purpose

The purpose of this report is to evaluate the effectiveness of tracking wastage services and identify any potential challenges or focuses on cleaning the campus environment. This report will provide an in-depth analysis of waste tracking systems, managing cleaners, and maintenance services, and moderate solutions for waste tracking. It will also examine the current environment of the campus area and raise awareness among students, teachers, and others.

1.3 Project Scope

The project scope for the Waste Tracker system entails the following details:

1. **Tracking and Resolution:** Upon receiving a waste incident report, the system will facilitate efficient tracking and resolution by notifying relevant authorities immediately. Designated cleaners or maintenance personnel will be assigned to address reported incidents through the app, with authorities able to monitor the progress and status of tasks in real time.

2. **Backend Integration:** The system will integrate with backend databases and servers to store and manage data related to waste incidents, user accounts, task assignments, and system configurations. This integration will ensure seamless data flow and enable efficient communication between frontend and backend components.

3. **Communication Channels:** The system will establish communication channels between users and authorities to facilitate collaboration and information exchange. Users will receive notifications and updates on the status of reported incidents, while authorities will have the capability to communicate with users and assign tasks through the app.

4. **Security Measures:** Robust security measures will be implemented to protect user data and ensure the integrity of the system. This includes encryption of data in transit and at rest, authentication mechanisms to verify user identities, and access control measures to restrict unauthorized access to sensitive information.

5. **Support and Maintenance:** Ongoing support and maintenance will be provided to ensure the continued functionality and effectiveness of the Waste Tracker system. This includes bug fixes, software updates, and user support services to address any issues or concerns that may arise during operation.

6. **Stakeholder Collaboration:** The project will involve collaboration with various stakeholders, including campus authorities, cleaning staff, students, faculty, and administrative staff. Their input and feedback will be incorporated into the system design and implementation process to ensure that the solution meets the needs and expectations of all stakeholders.

7. **Environmental Awareness:** The project aims to promote environmental awareness and sustainability within the campus community by encouraging responsible waste management practices and fostering a culture of cleanliness and environmental stewardship.

Overall, the project scope encompasses the development of a comprehensive waste management solution that leverages technology to address inefficiencies in campus waste management and promote a cleaner, safer, and more environmentally conscious campus environment.

1.4 Glossary

This section provides definitions for all document names, acronyms, and abbreviations. The application domain's terms and concepts are defined.

GUI - Graphical User Interface
API – Application Programming Interface
SRS – Software Requirement Specification
UI – User Interface
SDLC – Software Development Life Cycle
MB – Megabytes
XML – Extensible Markup Language
RESTful – Representational State Transfer
HTML – Hyper Text Markup Language

1.5 References

- IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

- Elsievier-*A systematic literature review on agile requirements engineering practices and challenges* .

- Bjarnason, E., Wnuk, K., & Regnell, B. (2011b*). Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development*. In 2011 IEEE 19th international requirements engineering conference (pp. 37–46)

- Wolfgang, E. (2011). *Working with user stories. In Agile requirements engineering workshop*, July 2011.

- Bjarnason, E., Wnuk, K., & Regnell, B. (2011b). *Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development*. In 2011 IEEE 19th international requirements engineering conference (pp. 37–46).

1.6 Overview

Universities, teeming with diverse activities, inevitably produce a lot of waste. To understand and tackle this environmental challenge, a waste tracker system can act as a powerful tool. By monitoring the type and amount of waste generated across campus, from dining halls to labs, such a system empowers informed decision-making. Easily identifying areas with high food waste, visualizing paper consumption trends, or pinpointing hazardous materials disposal patterns. Armed with this data, targeted strategies can be implemented to reduce waste, optimize recycling, and ultimately save costs.

The report is divided into several sections, including the project scope, introduction, purpose, methodology, findings, and recommendations. The project scope outlines the key areas covered in the report, while the introduction provides a background to the Waste Tracking System in a university. The purpose of this report is to evaluate the effectiveness of tracking wastage services and identify any potential challenges for improvement. The methodology outlines the research methods used to gather information for the report, while the findings section provides an analysis of the results of the research. The recommendations section provides practical and actionable recommendations for how the Waste Tracking System executes its services.

Beyond mere numbers, the system acts as an educational platform. Real-time data dashboards and insightful reports raise awareness among students, faculty, and staff about their individual and collective impact. This fosters a culture of sustainability, prompting behavioral changes and encouraging responsible waste management practices. Its potential extends further, providing valuable data for research projects and fueling educational campaigns that promote campus-wide sustainability initiatives.

The project scope can be tailored to specific needs, focusing on targeted areas or waste types. Integration with existing systems further enhances its functionality. Careful planning and stakeholder engagement are key to its success, ensuring the system effectively tracks waste, drives informed action, and ultimately paves the way for a greener, more sustainable university campus.

2. Stakeholders and Characteristics

2.1 Stakeholders:

Stakeholders are individuals, groups, or organizations that have an interest or concern in the outcomes or success of a project, initiative, organization, or system. They are those who can affect or be affected by the actions, decisions, policies, or goals of the entity in question. Stakeholders play a crucial role in various contexts, including business, project management, government, and nonprofit organizations. The term "stakeholder" encompasses a wide range of entities with diverse interests, and understanding their perspectives is essential for effective decision-making and project management.

2.2 Identification of Stakeholders:

Internal Stakeholders:

Internal stakeholders refer to individuals or groups within an organization who have a direct interest in or are directly affected by the operations, decisions, and outcomes of that organization. These stakeholders are typically part of the organization and may include various levels of employees, management, and internal departments. Internal stakeholders play a crucial role in shaping the direction and success of the organization.

- **Authority (administrative):** The administrative authority of a university typically handles organizational structure and leadership responsible for managing the day-to-day operations, decision-making, and overall governance of the institution. They have high power to provide acceptance and modify the whole system according to their opinion and control the systems properly.
- **Developer team:** Software developers responsible for designing, implementing, and maintaining the application.

External Stakeholders:

External stakeholders are individuals, groups, or organizations that exist outside of a particular entity (such as a company, organization, or project) but have high interest or low interest in or can be affected by its activities, decisions, or outcomes. These stakeholders are not directly involved in the day-to-day operations of the entity, but their interests may intersect with or be influenced by the entity's actions.

- **Students:** Student can share their thoughts and ideas.
- **Teachers:** Teachers provide their opinions based on the students' opinions.
- **Hall provosts:** They give instructions on how developers can customize the process of waste tracking and managing easily so that halls can maintain an environment of hygiene.
- **Nonprofit organizations:** Nonprofit organizations can share their innovative ideas and work policies to help us.
- **Staff (office, canteen, tongs):** They will cooperate with us on how we can help them.
- **Stationaries:** They can share their problem regarding waste management.
- **Restaurants and hotels:** we can discuss how we can control the waste tracking process easily.

- **Cleaners:** First, we will discuss how they manage waste manually, and then we can identify the disadvantages of manual systems. And then we can figure out the challenges to do it using software.

2.3 Assessment of Stakeholder Influence and Interest:

High power, low interest: This group consists of individuals or entities with significant decision-making power and authority but may have a low personal interest in the day-to-day details of a specific project or activity. They can make important decisions but may not be closely involved in the operational aspects.
- **Authority (administrative)**

Low power, high interest: Stakeholders in this category have a high level of interest or are directly affected by the project or activity but possess relatively less decision-making power. Their involvement is crucial for the success of the project, even though they may not have significant authority.
- **Students**
- **Teachers**
- **Nonprofit organizations**
- **Hall provosts**
- **Staff (office, canteen, tongs)**
- **Stationaries**
- **Restaurants and hotels**

High power, high interest: This category includes stakeholders who have both significant decision-making power and a high level of interest in the project. They are actively engaged in the project, can influence its direction, and are invested in its success.
- **Developer team**

Low power, low interest: Stakeholders in this category have limited decision-making power and low personal interest in the project or activity. They are not heavily involved in the project, and their impact on its success is minimal. Their concerns or involvement may be peripheral or tangential to the main project objectives.
- **Cleaners**

High power, low interest	High power, high interest
Authority (administrative)	**Developer team**
Low power, low interest	Low power, high interest
Students **Teachers** **Nonprofit organizations** **Hall provosts** **Staffs(office, canteen, tongs)** **Stationaries** **Resturants and hotels**	**Cleaners**

Stakeholders

Level of Power

Level of Interest
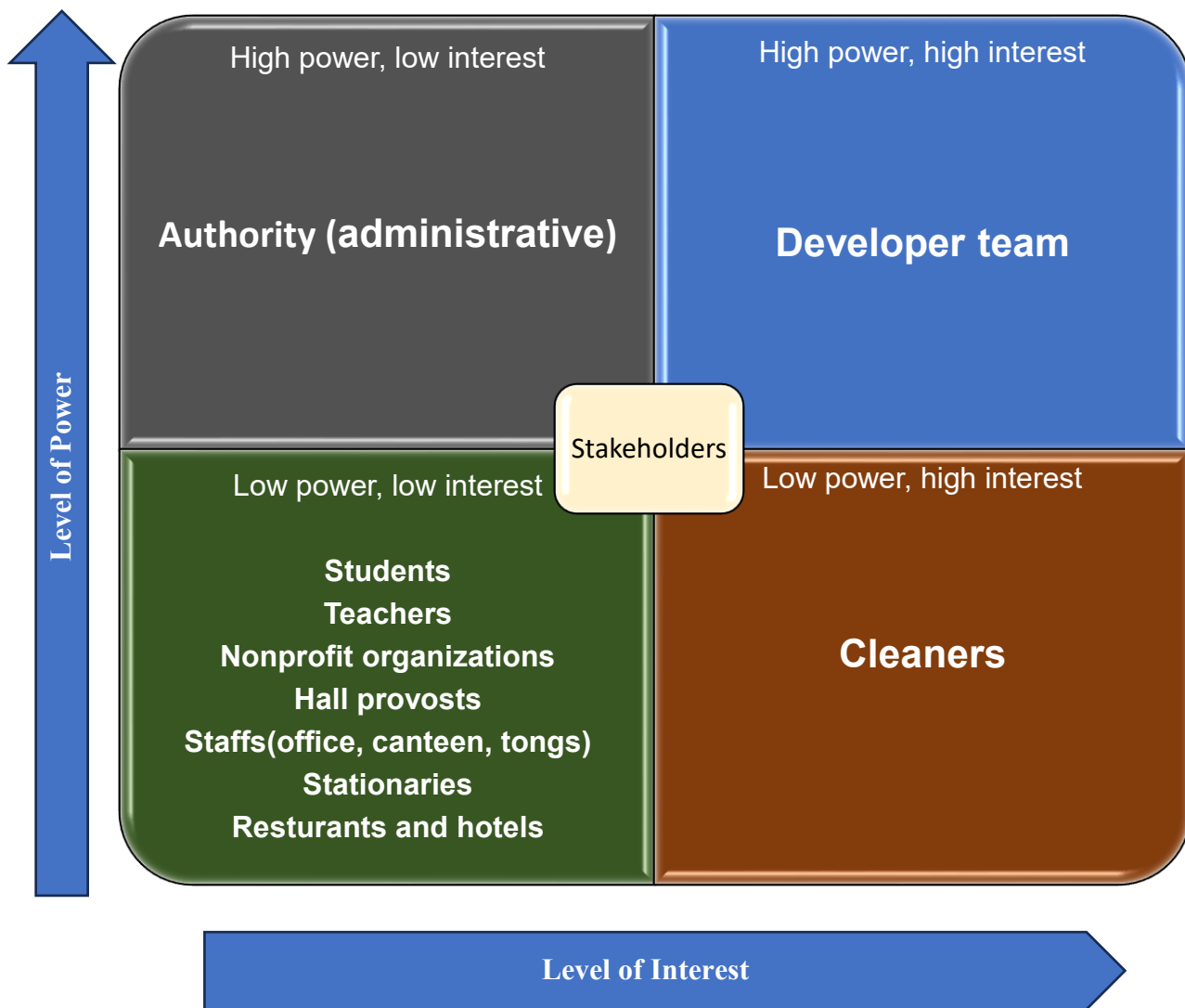
2.4 Conclusion:

In summary, managing waste is crucial for keeping our environment clean and safe. By following proper disposal practices, recycling, and using technology, we can protect our health, conserve resources, and comply with regulations. It's not just about throwing things away; it's about working together, educating communities, and planning for the long term. Responsible waste management benefits everyone and helps create a sustainable future for our planet.

- These are the stakeholders and their key characteristics for the Waste Tracking platform.

3. Design and Implementation Constraints

We have employed design and implementation constraints to ensure the success of this project. It also refers to a tool that allows developers and testers to inspect and interact with the application's user interface (UI) elements.
The design and implementation constraints for the Waste Tracking platform:

1. **Scalability:**
   - The Waste Tracking System must be designed to handle a growing number of users, posts, tasks, and data efficiently.
   - The architecture should allow for easy scalability to accommodate increased demand and usage over time.

2. **Security:**
   - The platform must implement robust security measures to safeguard user authentication data, post information, and sensitive details.
   - Encryption should be applied to protect data during transmission, and secure authentication methods should be employed.

3. **Availability:**
   - The system should maintain high availability to ensure users can access it at all times.
   - Redundancy, load balancing, and disaster recovery mechanisms should be in place to minimize downtime.

4. **User Experience:**
   - The platform should provide an intuitive and user-friendly experience for students, teachers, staff, faculties, and cleaners.
   - Features such as easy navigation, responsive design, and personalized notifications for updates should be incorporated.

5. **Data Management:**
   - The system must efficiently manage and store large volumes of data related to user accounts, wastage posts, tasks, and other relevant information.
   - Database design should be optimized, and measures for data security and backup should be implemented.

6. **Integration:**
   - The platform needs to seamlessly integrate with third-party services, such as notification systems and mapping services.
   - An API-based architecture should be adopted to facilitate smooth integration with other systems.

7. **Technology Stack:**
   - The platform should be developed using appropriate technology stacks based on the project's requirements.
   - Programming languages like Python or Java, frameworks such as Django or Node.js, and databases like PostgreSQL or MongoDB may be considered.

8. **Testing:**
   - Thorough testing processes, including unit testing, integration testing, and acceptance testing, should be implemented to ensure the platform's functionality, security, and performance.

3.1 JavaScript, JSX, and React.js

The visual layout of the components that a user could interact with in a website or technical product is referred to as user interface design or UI design. In other terms, it is a website's visual design.

3.1.1 Programming Language

JavaScript: JavaScript is an ECMAScript-compliant high-level, frequently just-in-time compiled language. It has first-class functions, dynamic typing, and prototype-based object orientation. It's multi-paradigm, allowing you to program in event-driven, functional, or imperative styles.
React is a front-end JavaScript toolkit for creating user interfaces using UI components that are free and open-source. Meta and a community of individual developers and businesses support it. JavaScript XML is abbreviated as JSX. It's just a JavaScript syntactic extension. It allows us to create HTML directly in React (within JavaScript code). It is straightforward to generate a template in React using JSX, but it is not a simple template language; instead, it has all of JavaScript's capabilities.

It is faster than standard JavaScript because it optimizes when converting to standard JavaScript. Rather than dividing the markup and functionality into different files, React makes use of components.

Java: Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, and run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile.Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

3.1.2 CSS Framework

Cascading Style Sheets (CSS) is a language for specifying the appearance of a document written in a markup language like HTML. Along with HTML and JavaScript, CSS is a key component of the World Wide Web. Semantic UI is a website using UI component framework. Developers may

use Semantic UI to create websites with quick and clear HTML, as well as a fully mobile responsive experience. Semantic UI offers a React-integrated version called Semantic UI React, which includes the following functionalities:

- jQuery Free.
- Declarative API.
- Augmentation.
- Shorthand Props.
- Sub Components.
- Auto Controlled State

3.2 Server-Side Technology

Server-side development refers to the actions that take place behind the scenes when an application is used. It primarily focuses on databases, scripting, website architecture, backend logic, APIs, and Servers.

3.2.1 Python, Django Rest API

Python is a dynamically semantic, interpreted, object-oriented high-level programming language. Its high-level built-in data structures, together with dynamic typing and dynamic binding, make it ideal for Rapid Application Development and as a scripting or glue language for connecting existing components. Python's concise, easy-to-learn syntax prioritizes readability, which lowers software maintenance costs.

Python is widely used to create an application's back end. It should come as no surprise that there are a variety of Python frameworks available to help with server-side programming. The Django REST Framework (DRF) is a popular, robust, and versatile framework for creating Web APIs.

To get data from our server to our application, we'll use the Django Rest API. Representational State Transfer is abbreviated as REST. Application Programming Interface is what API stands for.

3.2.2 Database Server

PostgreSQL is an advanced, enterprise class open-source relational database that supports both SQL (relational) and JSON (non-relational) querying. PostgreSQL's speed, security and robustness make it suitable for 99% of applications, so it's a great starting place for our application. It is a dependable, powerful, and stable solution with sophisticated features such as the following:

- User-defined types.
- Table inheritance.
- Sophisticated locking mechanism.
- Foreign key referential integrity.
- Views, rules, subquery.
- Nested transactions (save points)
- Multi-version concurrency control (MVCC)
Asynchronous replication.

3.2.3 Cloud Storage

Amazon S3 is a type of object storage that allows to store and retrieve any quantity of data from any location. It's a straightforward storage solution with industry-leading durability, availability, performance, security, and scalability. So we will use S3 to store audio files.

4. Requirement Specification

All the requirements based on the elicitation process are described in this section.

4.1 Functional Requirement

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and the explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data the system should hold, and the interfaces with the user.

4.1.1 User registration and authentication

FR-1	User Registration and Authentication to registered account.		
Description	Users should register his/her account for the first time and they must provide a valid email address during registration. The system shall authenticate users based on their username and password. Already registered users will not face this stage.		
Stakeholders	Students, Teachers, Staffs	**Priority**	High

4.1.2 User Login

FR-2	User Login
Description	Successful authentication, user will be able to log into the system using their registered credentials. In case of invalid credentials, the system will display an error message and provide an option for password recovery.

Stakeholders	Students, Teachers, Staffs	Priority	Medium

4.1.3 Forget Password

FR-3	Forgot password to recover password.		
Description	This functional requirement describes the process of password recovery for forgotten passwords in this system. The feature allows users to reset their password by providing a verification mechanism, such as an email or SMS, to ensure the user's identity. The system should enforce password strength requirements, display appropriate error messages, and provide a confirmation message to the user after they have successfully reset their password. The system should maintain an audit trail of all password recovery activities and ensure secure communication between the user's browser and the server. After resetting their password, users should be authenticated to the system and redirected to the appropriate page.		
Stakeholders	Students, Teachers, Staffs	Priority	Medium

4.1.4 Post Submission and Send Location

FR-4	Post submission and Send location of wastage		
Description	Users will post which containing pictures of waste inside the campus area. Besides posting users must be informed about the correct location from which place waste will be collected.		
Stakeholders	Students, Teachers, Staffs	Priority	High

4.1.5 Get Notification

FR-5	Get a Notification after post submission and do a comment		
Description	The system will get notified when a user does any new post or does any relevant comment for requesting cleaners. Getting notifications always occurred in real time.		
Stakeholders	Authority	**Priority**	High

4.1.6 Task assign

FR-6	Task assign based on submitted post		
Description	Admin will be able to assign tasks to cleaners based on submitted posts. Task should include details such as location, type of waste and urgency. Task need to complete as soon as possible.		
Stakeholders	Authority	**Priority**	High

4.1.7 Cleaner Tracking

FR-7	Cleaner tracking to ensure work progress		
Description	System will allow admin to track the status and location of assigned cleaners in real time. It will help to maintain work progress continuously.		
Stakeholders	Authority	**Priority**	Medium

4.1.8 Task Completion

FR-8	Report on task completion		
Description	Cleaners will report task completion to the admin through the system. System should update the task status accordingly and help to admin to know about their work status so that it will get approval.		
Stakeholders	Cleaners	**Priority**	Medium

4.2 Performance Requirement

It is important to maintain the performance of the software system. To ensure performance we maintain these steps:

Requirement Name	Requirement Description	Stakeholders	Priority
4.2.1 Response Time	The system must respond to user requests within 2 seconds for 90% of the time.	Users	High
4.2.2 Concurrent Users	The system must support a minimum of 500 concurrent users at any given time.		High
4.2.3 Scalability	The system must be scalable to handle increased traffic and users over time.	Authority	High
4.2.4 Availability	The system must have an uptime of 99.99% to ensure users can access it at any time.	Authority	High
4.2.5 Security	The system must ensure that user data is secure and protected from unauthorized access or data breaches.	Authority, Users	High
4.2.6 Data Backup	The system must regularly back up data to prevent data loss in case of hardware or software failures.	Authority	Medium

Requirement Name	Requirement Description	Stakeholders	Priority
4.2.7 System Maintenance	The system must undergo regular maintenance and updates to ensure optimal performance and minimize downtime.	Authority	Medium
4.2.8 Load Testing	The system must undergo regular load testing to identify and address performance issues.	Authority	Low
4.2.9 Resource Utilization	The system must utilize system resources efficiently to minimize hardware and infrastructure costs.	Authority	Low

4.2.10 Capacity Requirement

Here are some capacity requirements for our project:

4.2.2.1 Bandwidth

The system must have sufficient bandwidth to support high traffic volume during peak periods, such as holiday seasons or promotional events.

4.2.2.2 Storage Capacity

The system must have sufficient storage capacity to store product information, user data, and order history.

4.2.2.3 Processing Power

The system must have sufficient processing power to handle large volumes of data, including transactions, searches, and analytics.

4.2.2.4 Database Capacity

The system must have sufficient database capacity to store user information, transaction data, and inventory information.

4.2.2.5 Server Capacity

The system must have sufficient server capacity to handle user requests and ensure that the system runs smoothly.

4.2.2.6 Network Capacity

The system must have sufficient network capacity to support multiple user connections and ensure that the system is accessible from different locations.

4.2.2.7 Memory Capacity

The system must have sufficient memory capacity to handle multiple transactions and user requests without any delay or interruption.

4.2.3 Safety Critical Requirement

There are no safety-critical requirements for our project.

4.2.4 Robustness or Fault-Tolerance Requirements

4.2.4.1 Error Handling

The system must have proper error-handling mechanisms in place to handle unexpected errors or exceptions and prevent the system from crashing or becoming unresponsive.

4.2.4.2 Redundancy

The system must have redundant components, such as servers, databases, or network connections, to ensure that critical functions are available even in case of failures.

4.2.4.3 Failover

The system must have a failover mechanism that automatically switches to a backup system or component in case of a failure to ensure that the system remains operational.

4.2.4.4 Load Balancing

The system must have load-balancing mechanisms that distribute the load across multiple servers to ensure that no single server is overwhelmed with requests and that the system remains responsive.

4.2.4.5 Performance Monitoring

The system must have performance monitoring mechanisms that track system performance and alert administrators if the system falls below-defined thresholds.

4.2.4.6 Recovery Time Objectives (RTO)

The system must have defined RTOs for each critical component or system function, specifying the maximum acceptable downtime and the required recovery time in case of a failure.

4.2.4.7 Backups

The system must have a backup mechanism that regularly backs up critical data to ensure that data can be recovered in case of a catastrophic failure.

4.3 Maintainability and Supportability

4.3.1 Maintenance Requirements

4.3.1.1 Regular Software Updates

The system must have a mechanism to update the software to the latest version, which includes bug fixes, performance improvements, and security enhancements.

4.3.1.2 Regular Hardware Maintenance

The system must have regular hardware maintenance, which includes cleaning, inspection, and repair or replacement of faulty components.

4.3.1.3 Data Backup and Recovery

The system must have a regular data backup mechanism to ensure that critical data is not lost in case of hardware or software failures.

4.3.1.4 System Monitoring

The system must have a system monitoring mechanism that tracks system performance, usage patterns, and error logs to identify potential issues and optimize system performance.

4.3.1.5 User Support

The system must have a user support mechanism to help users troubleshoot issues, provide guidance, and resolve user complaints or issues.

4.3.1.6 Training

The system must provide training to administrators and users to ensure that they are familiar with the system's features and capabilities, and can use the system effectively.

4.3.1.7 Documentation

The system must have documentation that outlines the system architecture, design, and usage, which is helpful for troubleshooting, maintenance, and training purposes.

4.3.2 Supportability Requirements

This system meets Testability, Maintainability, Compatibility, configure ability, Serviceability, and install ability which are related to supportability requirements.

4.4 Security Requirements

Securing information is much more important for our system to get users dependability. Here are some of them:

4.4.1 Authentication and Authorization

The system must have a mechanism to authenticate users and authorize access to system resources based on the user's role and level of access.

4.4.2 Encryption

The system must use encryption mechanisms to secure sensitive data, such as login credentials, financial data, and personal information, during transmission and storage.

4.4.3 Access Control

The system must have access control mechanisms to restrict access to sensitive data and system resources to authorized personnel only.

4.4.4 Network Security

The system must have network security mechanisms, such as firewalls and intrusion detection systems, to protect against network attacks and unauthorized access to the system.

4.4.5 Audit Trails

The system must have audit trail mechanisms that log user activity and system events, to enable the identification and investigation of security breaches and unauthorized access attempts.

4.4.6 Data Backup and Recovery

The system must have a data backup and recovery mechanism to ensure that critical data can be restored in case of data loss or corruption.

4.4.7 System Monitoring

The system must have system monitoring mechanisms that detect and alert administrators of any suspicious or malicious activity on the system.

4.4.8 Incident Response

 The system must have an incident response plan in place to handle security breaches or attacks and minimize the impact of such incidents.

4.5 Usability and Human Integrity Requirements

This system will provide a more user-friendly environment

4.5.1 Ease of Use Requirements

Our system will be easier to use by any type of people and they don't need any training to use the system.

4.5.2 Accessibility Requirements

To get access to the application, the application provides authorization/authentication. This application will use various modules.

4.5.2.1 Navigation

The system must have a clear and consistent navigation mechanism, which includes keyboard shortcuts, to help users navigate the system effectively.

4.5.2.2 Text Size and Font

The system must have options to increase or decrease the text size and change the font style to help users with visual impairments read the content easily.

4.5.2.3 Color Contrast

The system must have sufficient color contrast between the background and foreground elements to help users with visual impairments distinguish between different elements on the screen.

4.5.2.4 Alternative Text

The system must provide alternative text for images, videos, and other multimedia elements to help users with visual impairments understand the content.

4.5.2.5 Audio and Video Transcripts

The system must provide transcripts for audio and video content to help users with hearing impairments understand the content.

4.5.2.6 Forms and Input Fields

The system must have accessible forms and input fields, which include labels, hints, and error messages, to help users with disabilities fill out the forms accurately.

4.5.2.7 Assistive Technology Compatibility

The system must be compatible with assistive technologies, such as screen readers, text-to-speech software, and voice recognition tools, to help users with disabilities access the system effectively.

4.6.2.8 Standards Compliance

The system must comply with accessibility standards, such as Web Content Accessibility Guidelines (WCAG) 2.1, to ensure that the system is accessible to users with disabilities.

4.6 Look and Feel Requirements

Look and feel requirements mainly refer to how the system will look.

4.6.1 Appearance Requirements

AR-1	Text color and font
Description	Our system has to be different and attractive from other existing system using a better look and feel.

Stakeholders	Developer, User	Priority	High

4.7 Style Requirements

There are no style requirements in our system.

4.8 Legal Requirements

Legal requirements normally refer to the terms and conditions or privacy policies of any organization. The terms and conditions of our application are that no third-party software or person is allowed to use our data for their business purpose.

5. Requirement Engineering Process

Requirements Engineering (RE) determines software requirements according to user requirements or needs. The requirements engineering process includes requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management.

5.1 Requirement Elicitation Techniques

Requirements elicitation is the practice of researching and finding system requirements for users, users, and other stakeholders, also referred to as "requirement gathering". Requirement elicitation can be done by contacting participants directly or by doing some research, analysis and testing.

5.1.1 Hold Interviews

We hold discussions that can be held individually or with a small group of participants. They are an effective way to access services without spending a lot of time with participants because we meet with people to discuss only certain important requirements of this program. Negotiations are useful for obtaining individual requirements for members in organizing workshops where those members of the program come together to resolve any issues or conflicts. We mainly perform our interview based on some specific criteria.

- Short description about goals and objectives
- Registration process
- Searching Audio Files
- Storage system of each account
- Compression size of audio files

5.1.2 Perform Document Analysis

Existing documentation can help to show how systems are currently operating or what they are what I should do. Documents include written information about current programs, business processes, needs specifications, and competitor research. Review once textual analysis can help determine which performance should remain and functionality that isn't in use. After existing document

In the analysis, we found several problems with the existing system.

- Existing systems cannot perform file compression.
- A user cannot share a file with others.
- No cloud storage system is provided by the existing systems.

5.1.3 System Interface Analysis

The first thing to do is to identify which systems the system-to-be shall communicate with. It could be a server on the Internet, a piece of software on the same host as the system-to-be, some hardware or something completely different.

5.1.4 Distribute Questionnaires

The questionnaire is a useful way to investigate styles, changes in attitudes and users' ideas, and user satisfaction with priorities and preferences. Our lists of questions were as short as possible. The respondent may be tired or frustrated. Had a basic reason for all the questions as well as grouped the topic areas together for the respondent to focus on. The main advantage of these survey responses was that they were collected in the usual way. Information was summarized by a large number of people.

5.2 Sample of Requirement Collection

5.2.1 Requirement collection -1

This report summarizes the results of the questionnaire distribution conducted to gather requirements for our system. The objective of the surveys was to identify the key needs and expectations of the stakeholders and to use this information to develop a comprehensive set of requirements for the system.

Interview

Methodology:

The interviews were conducted with stakeholders from various students, and teachers in NSTU Campus. The interviews were conducted in a one-to-one format, lasting approximately 7-10 minutes each.

Participants:

A total of 43 & 50 responses were collected for project requirements & functional requirements respectively.

Findings:

The following are the key findings from the interviews:

- Mobile Application is more comfortable.
- User can post images and send locations in real-time.

- They can create awareness to clean the campus environment.
- They can comment on posts if the place is not clear on time.
- Notification will be sent timely to the authority.

Key Requirements:

Based on the findings from the interviews, the following are the key requirements for Waste Tracking System:

- User-friendly interface with easy navigation.
- Ability to post images.
- Real-time notification sends
- Track cleaner
- Submit Feedback

Assumptions:

It was assumed during the survey process that the waste tracking system will be accessible via the web and we have to develop an Android app for the system.

Limitations:

Responses were collected less than the expectation.

Conclusion:

The stakeholders' questionnaire surveys provided valuable insights into the requirements for the waste tracking system. The key findings and requirements will be used to develop a comprehensive set of requirements for the system.

5.2.2 Requirement collection -2

This report summarizes the results of the stakeholder interviews conducted to gather requirements for our system. The objective of the interviews was to identify the key needs and expectations of the stakeholders and to use this information to develop a comprehensive set of requirements for the system.

Distribute Questionnaires

Methodology:

The Questionnaires were distributed to stakeholders from various cities and roles. It takes 3-4 minutes to complete the questionnaire.

Participants:

A total of 13-16 were interviewed, including:
10-12 students
4-5 teachers
Findings:

The following are the key findings from the interviews:

- Mobile Application is more comfortable.

- User can post images and send locations in real-time.
- They can create awareness to clean the campus environment.
- They can comment on posts if the place is not clear on time.
- Notification will be sent timely to the authority.

Key Requirements:

Based on the findings from the interviews, the following are the key requirements for Waste Tracking System:

- User-friendly interface with easy navigation.
- Language Option.
- Upload Images of wastages.
- Do comment in real time

Assumptions:

It was assumed during the interview process that the Waste Tracking system will be accessible via the web and we have to develop an Android app for the system.

Limitations & Challenges:

Students and teachers should be careful about this kind of thing.

Conclusion:

The stakeholders' interviews provided valuable insights into the requirements for the Waste Tracking system. The key findings and requirements will be used to develop a comprehensive set of requirements for the system.

5.3 Requirement Validation

Requirement validation ensures that the requirements are correct and reflect the quality you want from this program. In the beginning, our requirements looked good but when we read them and tried to work with them, they came out having ambiguities and gaps.

5.3.1 Review the Requirements

Negative peer review, especially the type of rigorous review called evaluation, is unique among the highest-quality software processes available. We had a team of reviewers representing different perspectives and carefully examined written needs, analysis models, and related information on disability.

5.3.2 Test the Requirements

The test creates another view of the requirements. We also performed writing tests regarding assurance of whether the expected performance was found or not. Getting tested by the user needs to document the expected product behavior under specified conditions.

5.3.3 Simulate the requirements

To stimulate requirements, trading tools are available that we have used to simulate a proposed system in place or to add details of written requirements. The simulation takes prototyping to the next level.
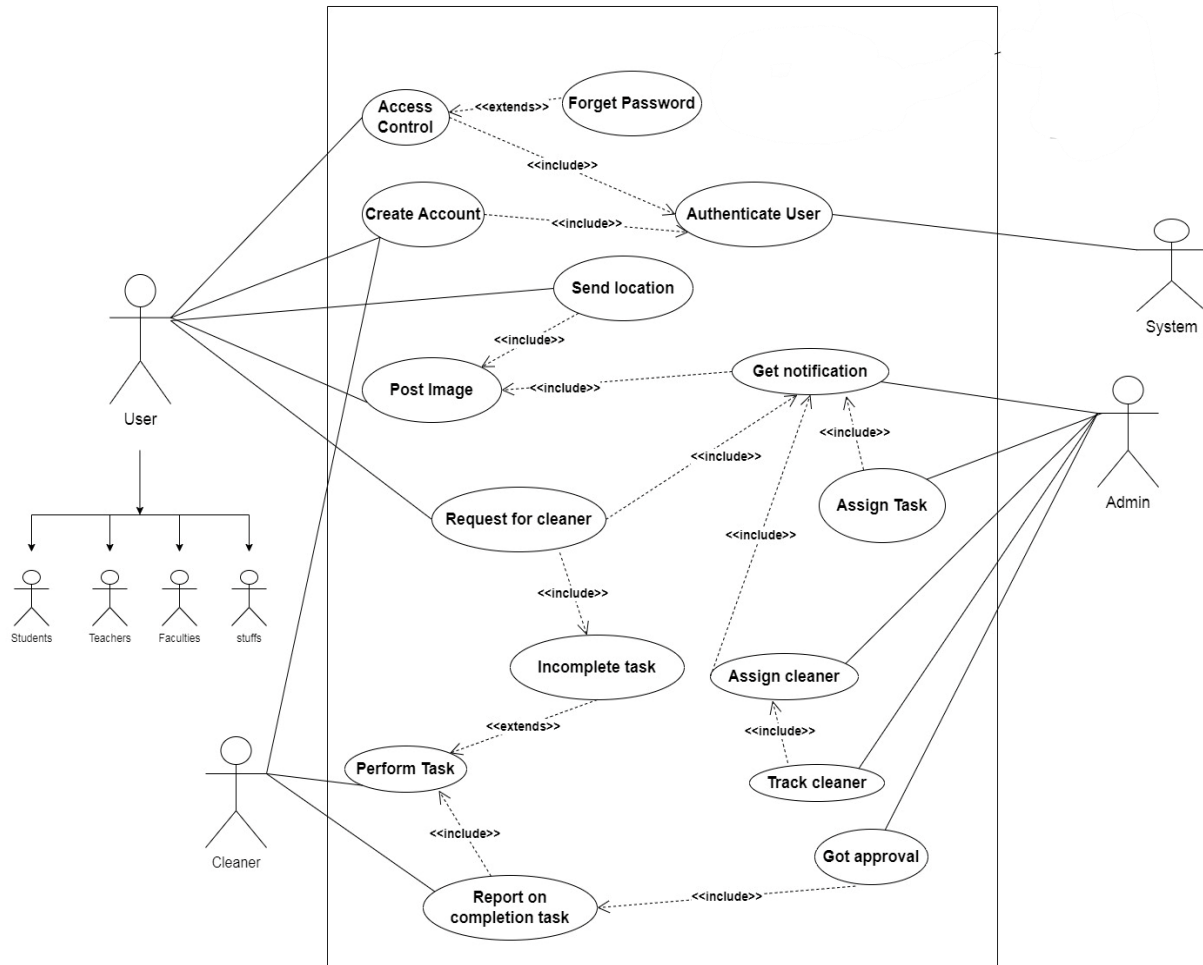
6. Use Case Diagram

Figure 1: Use case Diagram

7. Use Case Description

Table-01:
UC-01: Create Account

Use Case	Create Account
Goal	Allow users to create an account in the Waste Tracking System
Precondition	N/A

Use Case	Create Account
Success End Condition	User account is created and the system confirms the creation
Failed End Conditions	1)The username or email is already in use 2)The User's information is incomplete or invalid
Primary Actor	Students, Teachers, Staffs
Secondary Actor	N/A
Trigger	The user requests to create an account
Main Success Flow	1)The system presents the user with a registration form, 2)The user enters their personal information (e.g. name, address, email, username, password) 3)The user submits the form, 4)The system validates the user's information, 5)The system creates a new user account and assigns a unique ID, 6)The system sends a confirmation email to the user, 7) The system displays a success message to the user.
Alternative Flow	1) If the user enters invalid or incomplete information, the system displays an error message and asks the user to correct the information 2)If the username or email is already in use, the system displays an error message and asks the user to choose a different username or email
Quality Requirements	1)The system should be able to handle a large number of user account creations simultaneously without crashing, 2) The user's information should be securely stored and protected from unauthorized access.

Table-02
UC-02: Access Control

Use Case	Access Control
Goal	Allow users to log in to their account in the Waste Tracking System
Precondition	The user must have an existing account in the system
Success End Condition	User is logged in and has access to their account information
Failed End Conditions	1)The user enters incorrect login credentials, 2) The user's account has been deactivated
Primary Actor	Students, Teachers, Staffs
Secondary Actor	N/A
Trigger	The user requests to log in to their account
Main Success Flow	1)The system presents the user with a login page 2)The user enters their username and password, 3)The user clicks the "Log in" button 4)The system validates the user's login credentials, 5) The system logs the user in and grants access to their account information
Alternative Flow	1)If the user enters incorrect login credentials, the system displays an error message and asks the user to re-enter their credentials 2)If the user's account has been deactivated, the system displays an error message and asks the user to contact user support
Quality Requirements	1)The system should protect the user's login credentials from unauthorized access or disclosure,

Use Case	Access Control
	2) The system should be able to handle a large number of simultaneous login requests without crashing or slowing down.

Table-03:
UC-03: Post Image

Use Case	Post Image
Goal	Allow users to post pictures of wastages from the campus area in the system
Precondition	The user must be logged in to their account
Success End Condition	The picture will upload successfully
Failed End Conditions	None
Primary Actor	Students, Teachers, Staffs
Secondary Actor	None
Trigger	The user sends the picture of wastage so that the authority can get a notification
Main Success Flow	1)The system helps to upload the pictures, 2)The user can see if the picture is uploaded or not, 3)The user can see if anyone else do posts or comment on any post
Alternative Flow	1)If pictures are not uploaded, the system will reply to repost the picture or check internet connections
Quality Requirements	1)The system should be able to handle storing and displaying many posts randomly

Table-04
UC-04: Send location

Use Case	Send location
Goal	Allow users to send the current location of the waste
Precondition	The user must be logged in to their account and must post a picture of waste
Success End Condition	The user has successfully sent the location of wastage to the authority
Failed End Conditions	1) The authority can't assign cleaners on time, 2)wastage can't be tracked and cleaned properly
Primary Actor	Students, Teachers, Staffs
Secondary Actor	None
Trigger	The user sends the proper location of the waste
Main Success Flow	1)The system presents the user with an input box that takes location as an input, 2)The user fills out the input box and clicks the "add" button 3)The system helps to notify the authorities about the proper location,
Alternative Flow	1)If the location is not set then it will show to fill up the location box, otherwise tracking process will stop.
Quality Requirements	1) The Authority will notify with proper location and it will help to assign tasks and clean the place quickly

Table-05
UC-05: Perform Task

Use Case	Perform Task
Goal	The goal of this use case is to perform the assigned task by the authority.
Preconditions	Relevant authorities should assign the task to the cleaners.
Success end condition	Wastes are cleaned from the reported area.
Failed end condition	The area is not cleaned.
Primary Actor	Cleaner
Trigger	The cleaner receives a task assignment through the app.
Main Success Flows	1. Cleaner will go to the location to perform the task. 2. Upon arrival, the cleaner assesses the situation and takes appropriate actions to clean up the waste.
Alternative Flow	1a) If the cleaner finds unexpected difficulties, he can communicate with the authorities. 2a) If the assigned task is no longer feasible or relevant, the cleaner may notify the authority.
Quality Requirements	1. Proper information should be given to the cleaner to perform the required task efficiently.

Table-06:
UC-06: Report on Completion Task

Use Case	Report on Completion Task
Goal	The goal of this use case is to allow the cleaners to report after the completion of the required task to the relevant authorities.
Preconditions	Cleaners have to complete the assigned task.
Success end condition	The cleaner has successfully reported to the authorities.
Failed end condition	Cleaners failed to report.
Primary Actor	Cleaners
Trigger	The cleaner has completed a task assigned to them and needs to report its completion to the relevant authority.
Main Success Flows	1. Cleaners report to admin as they complete their task.
Alternative Flow	1a) Fail to report
Quality Requirements	1. Cleaners must provide clear and accurate information about the completion of the task to the authority.

Table-07
UC-07: Authenticate User

Use Case	Authenticate User
Goal	The goal of this use case is to authenticate the user's identity to ensure secure access to their account and personal information.
Preconditions	The user must have created an account on the platform.
Success end condition	The user successfully logs into their account using their login credentials.
Failed end condition	The user is unable to log in due to incorrect login credentials or the platform experiences an error during the authentication process.
Primary Actor	User
Secondary Actor	Authority
Trigger	The user clicks on the "Login" button and enters their login credentials.
Main Success Flows	1. The platform validates the user's login credentials. 2. The platform grants access to the user's account and personal information. 3. The user can view their account information and access the platform's features and services.
Alternative Flow	2a) The user may choose to reset their password if they have forgotten their login credentials.
Quality Requirements	1. The platform must use secure methods for storing and transmitting login credentials. 2. The platform must provide clear and accurate error messages in case of incorrect login credentials. 3. The platform must provide real-time updates to the user's account information. 4. The platform must ensure the confidentiality of the user's personal information.

Table-08
UC-08: Forgot Password

Use Case	Forgot Password
Goal	The goal of this use case is to allow users to request a password reset.
Preconditions	The user has previously registered an account and is unable to log in to the app due to forgetting their password.
Success end condition	The app confirms the successful password resets and takes the user to the login screen to access their account using the new password.
Failed end condition	Unsuccessful in password reset.
Primary Actor	User
Secondary Actor	System
Trigger	The user selects the "Forgot Password" option provided on the login screen.
Main Success Flows	1. After selecting the "Forgot password" option the app prompts the user to enter their registered email address for password reset. 2. The user inputs the necessary information and requests for a password reset. 3. The app verifies the authenticity of the request and prompts the user to enter a new password. 4. After submitting the new password, the app confirms the successful password reset and redirects the user to the login screen.
Alternative Flow	2. a) If the user inputs the wrong email there will be an error.

Use Case	Forgot Password
	4. a) If the user finds difficulties in resetting the password, he may contact technical support for assistance.
Quality Requirements	1. Proper information should be given for resetting the password. 2. Clear support options should be provided to assist the users.

Table-09:
UC-09: Assign Cleaners

Use Case	Assign Cleaners
Goal	The goal of this use case is to assign the cleaners to perform the given tasks.
Preconditions	1. Authority should have access to the system. 2. A waste incident should be reported through the system.
Success end condition	Successfully assigned cleaners.
Failed end condition	No cleaners are assigned to perform the required task.
Primary Actor	Authority
Secondary Actor	N/A
Trigger	A waste incident is reported and cleaners are required to address the situation.
Main Success Flows	1. Authority will log into the system. 2. The system will display a list of reported incidents, along with their details and current status. 3. The Authority will assign cleaners.
Alternative Flow	1a) If any issue happens to the system access can be denied. 2a) If there is to reported incidents, no need to assign cleaners.
Quality Requirements	1. Cleaners should be available for the specific tasks.

Table-10:
UC-10: Track Cleaner

Use Case	Track Cleaner
Goal	To monitor the progress of the cleaners.
Preconditions	Cleaners have to be assigned.
Success end condition	Successfully monitoring the cleaners.
Failed end condition	The authority has failed to track the cleaners.
Primary Actor	Authority
Secondary Actor	N/A
Trigger	The authority needs to monitor the progress of the cleaners for proper cleaning.
Main Success Flows	1. The authority will monitor the movement of cleaners
Alternative Flow	1a) Tracking multiple cleaners can be difficult.
Quality Requirements	Clear communication channels and support mechanisms should be in place to facilitate effective coordination.

Table-11:
UC-11: Get Notification

Use Case	Get Notification
Goal	To get any update of a post
Preconditions	The user must do post .
Success End Condition	Admin will get a notification over a post
Failed End Condition	Admin doesn't get any notification
Primary Actor	Authority
Secondary Actor	User
Trigger	The notification option will show the number of notifications
Main Success Flow	1. Admin will get notifications and real-time update
Alternative Flow	1a) They may not be able if any post doesn't do.
Quality Requirements	The system will work properly.

Table-12:
UC-13: Got Approval

Use Case	Got Approval
Goal	the goal of this case is to approve a post after a task being completed
Preconditions	Tasks have to be completed.
Success End Condition	Successfully approved
Failed End Condition	Approval unsuccessful.
Primary Actor	Authority
Secondary Actor	N/A
Trigger	Admin approves the posts.
Main Success Flow	1. Authority checks the completion of the task. 2. After completion, approve the post.
Alternative Flow	2a) If the task is not completed and any report is not provided admin won't approve.
Quality Requirements	The system should be able to approve many posts randomly.

8. Activity Diagram
Activity Diagram (Create Account)

Figure 02: Create Account

Activity Diagram (Access Control)
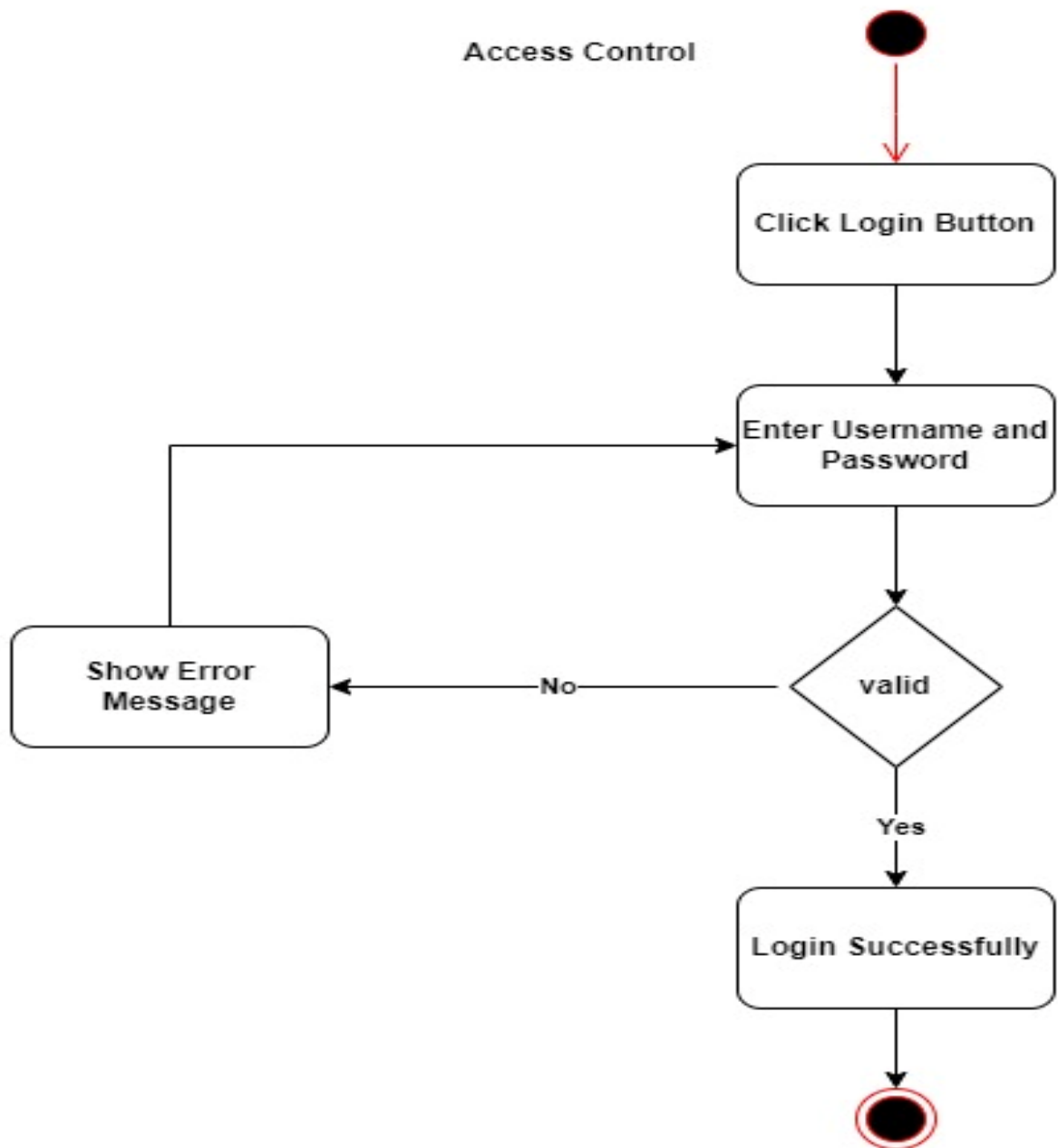
Figure 03: Access Control

Activity Diagram (Post Image)
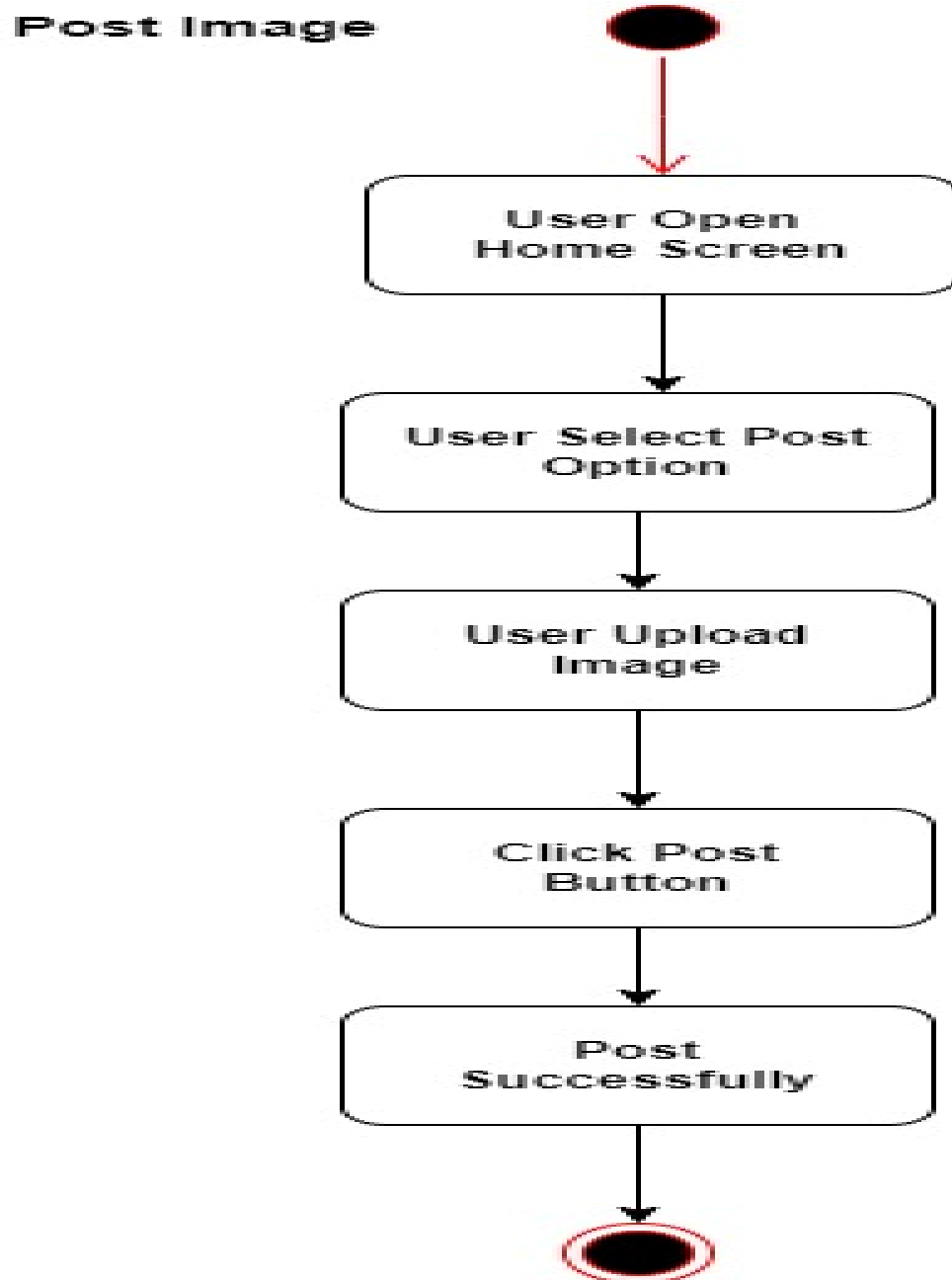
Figure 4: Post Image

Activity Diagram (Send Location)
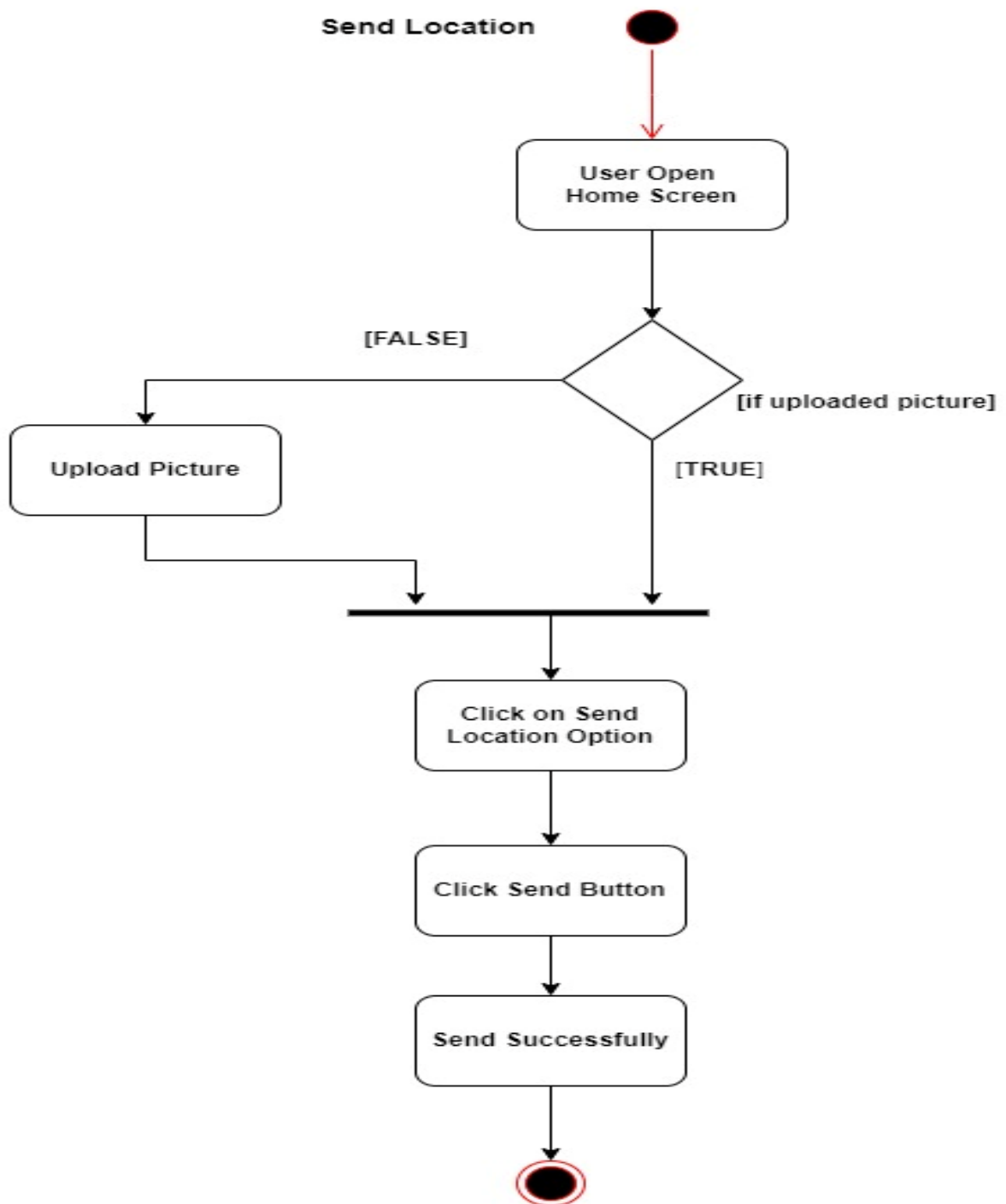
Figure 5: Send Location

Activity Diagram (Get Notification)
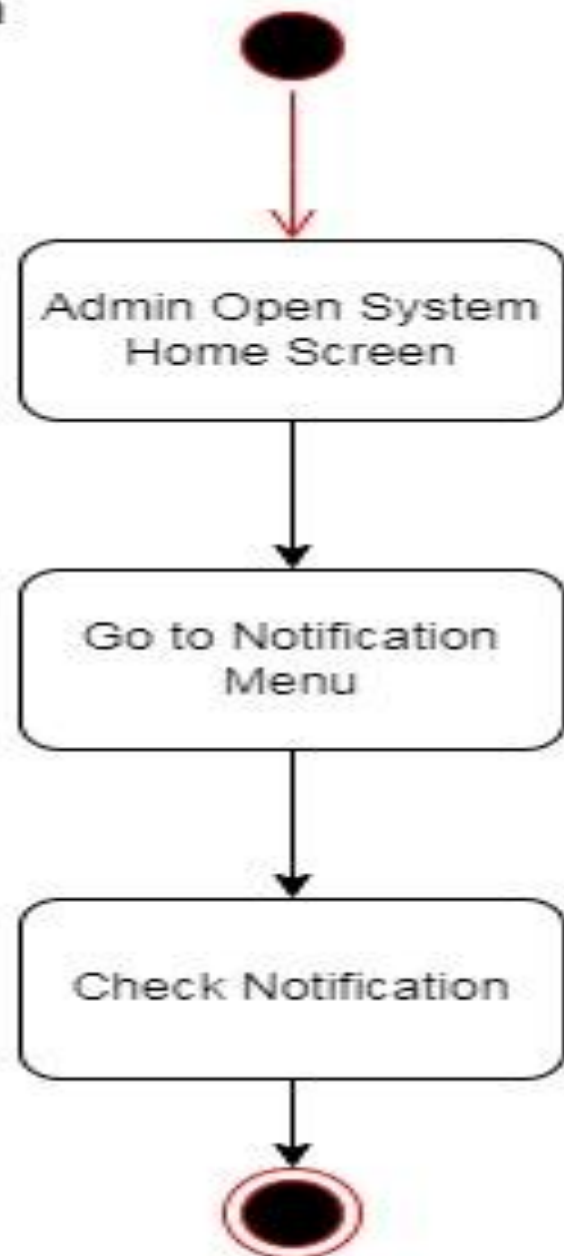
Figure 6: Get Notification

Activity Diagram (Assign Cleaners)
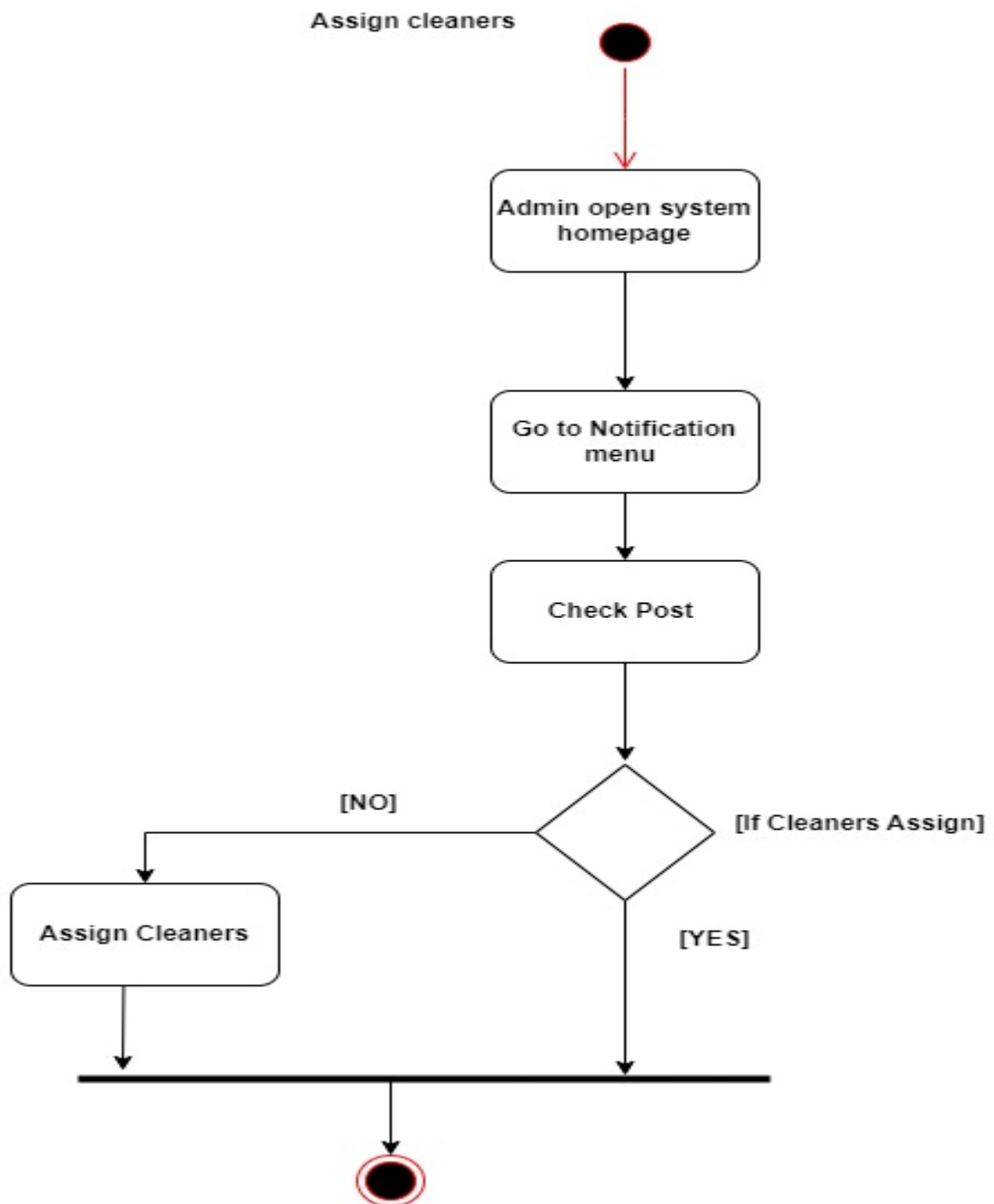
Figure 7: Assign Cleaners

Activity Diagram (Assign Tasks)
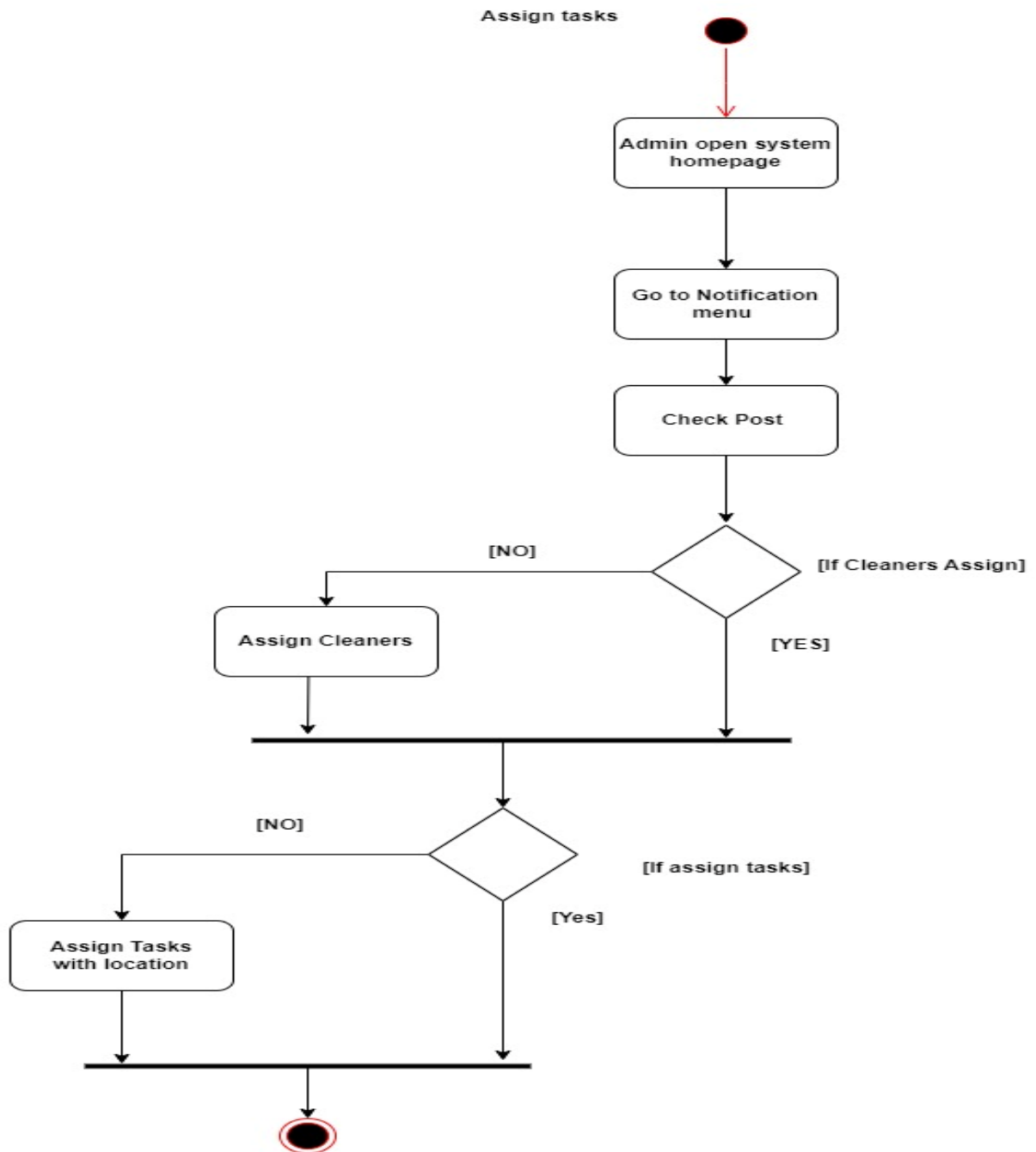
Figure 8: Assign Tasks

Activity Diagram (Task approval)
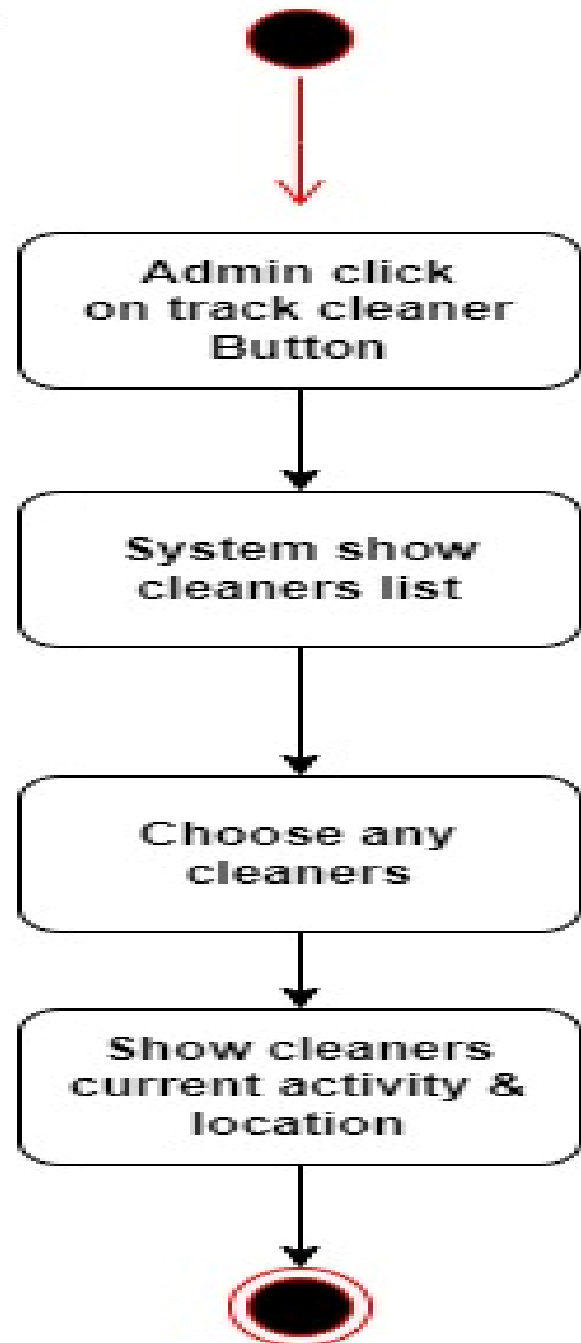
Figure 9: Post Approval

9. Sequence Diagram

Sequence Diagram (Create Account)

Create Account
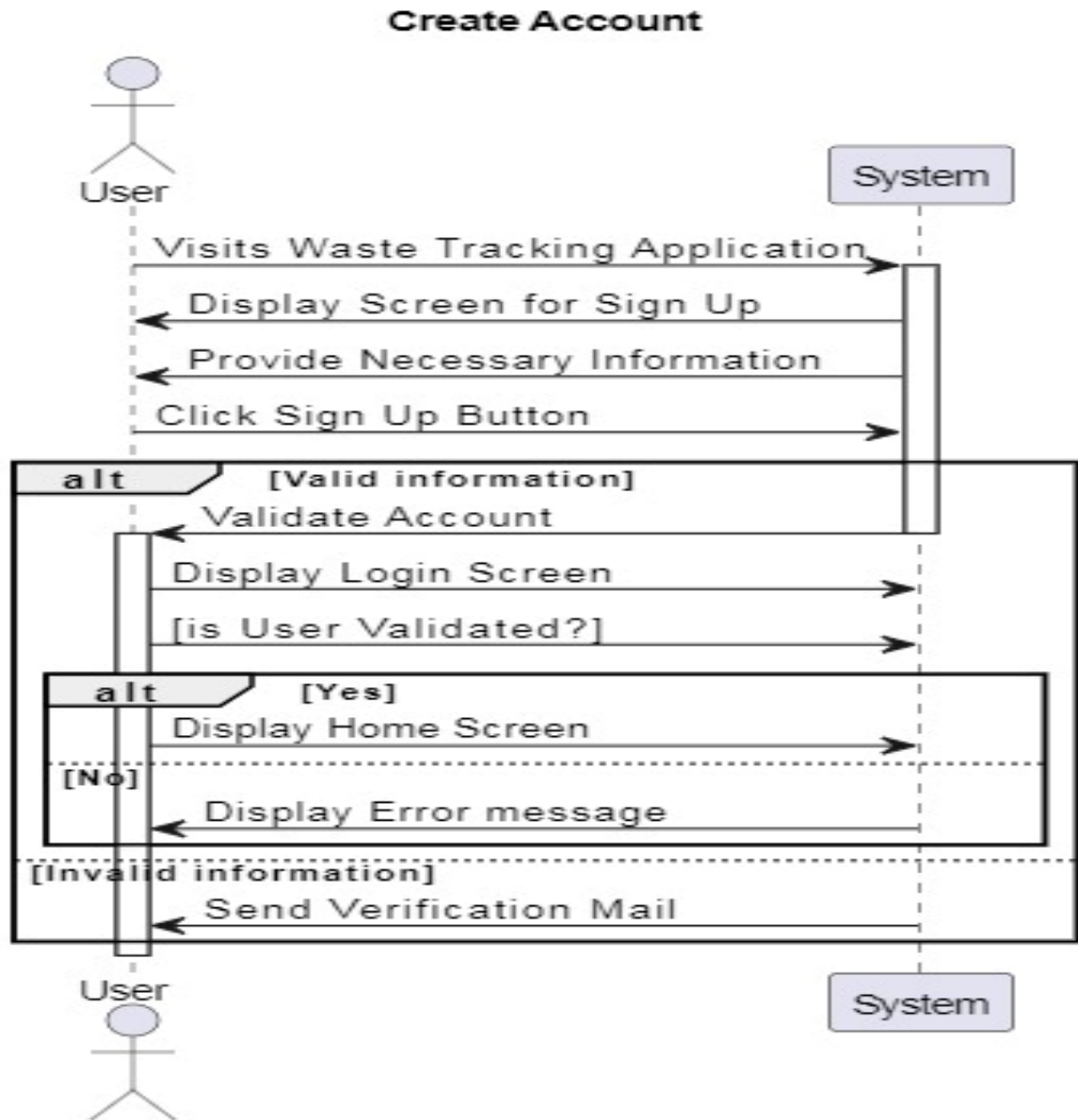
Sequence Diagram (Access Control)

User

Access Control System

Click Login Button

Prompt for Username and Password

Enter Username and Password

Validate Credentials

alt   [Credentials Are Valid?]

Login Successful

Show Error Message

User

Access Control System
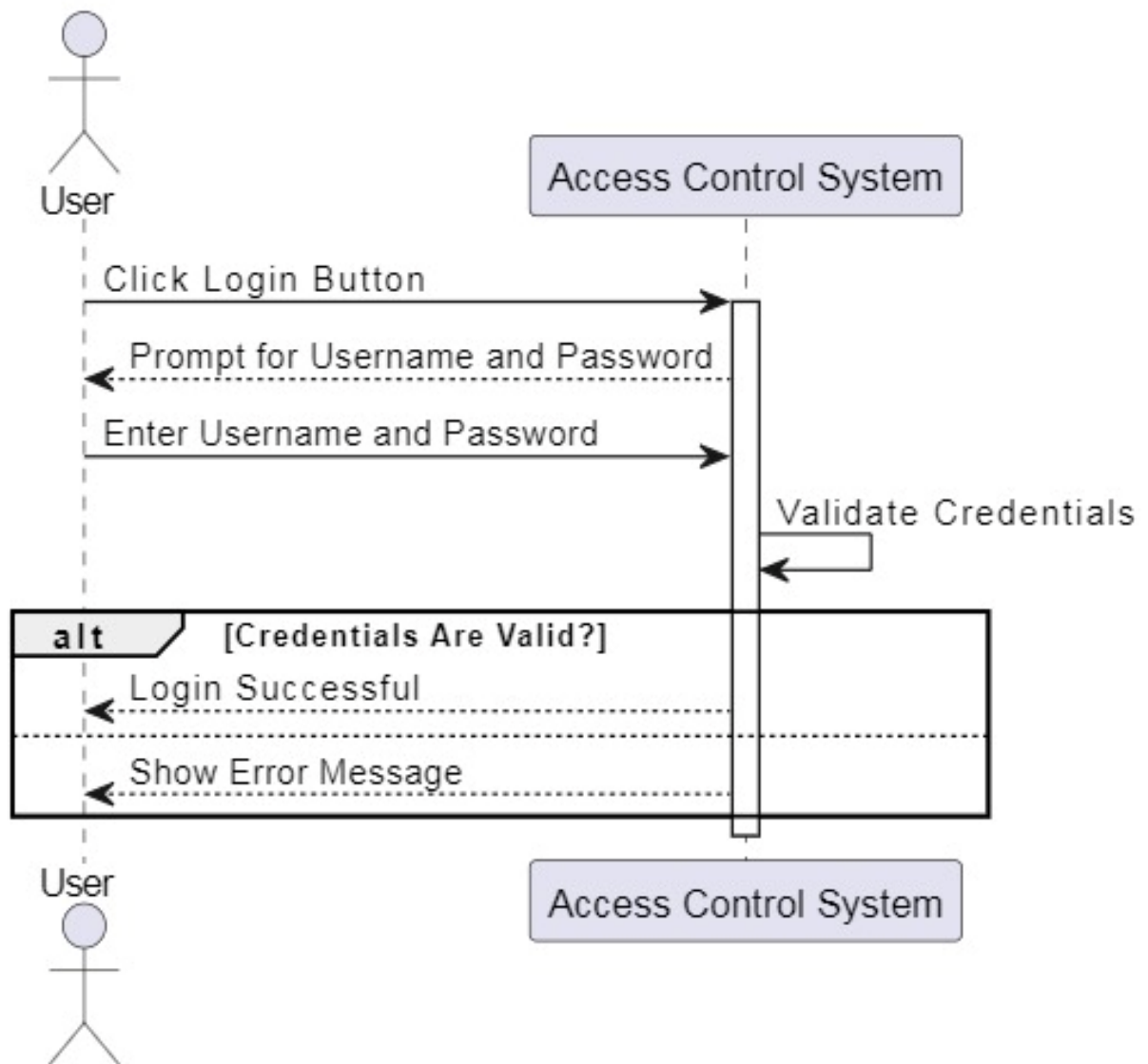
Sequence Diagram (Post Image)
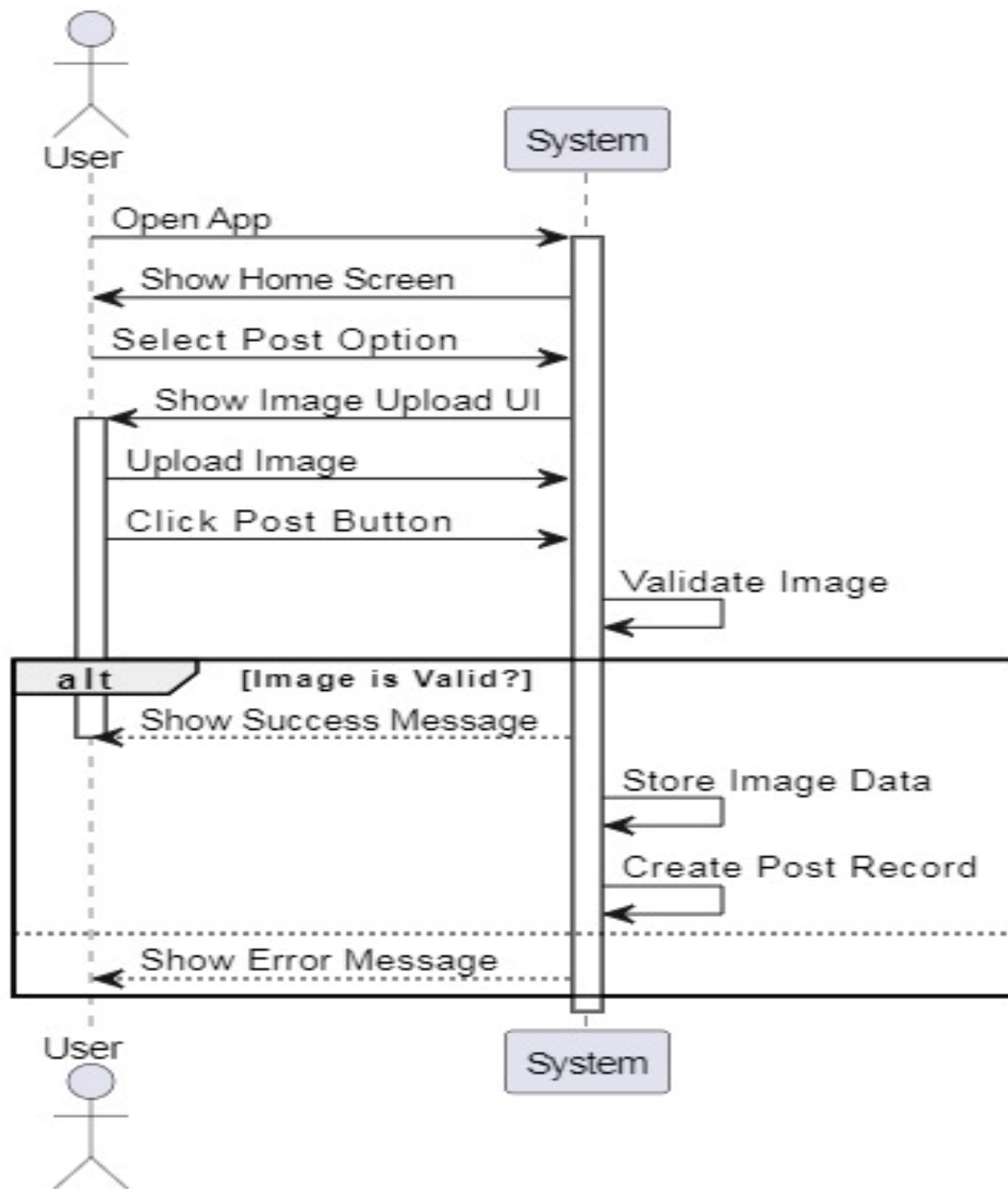
Sequence Diagram (Send Location)
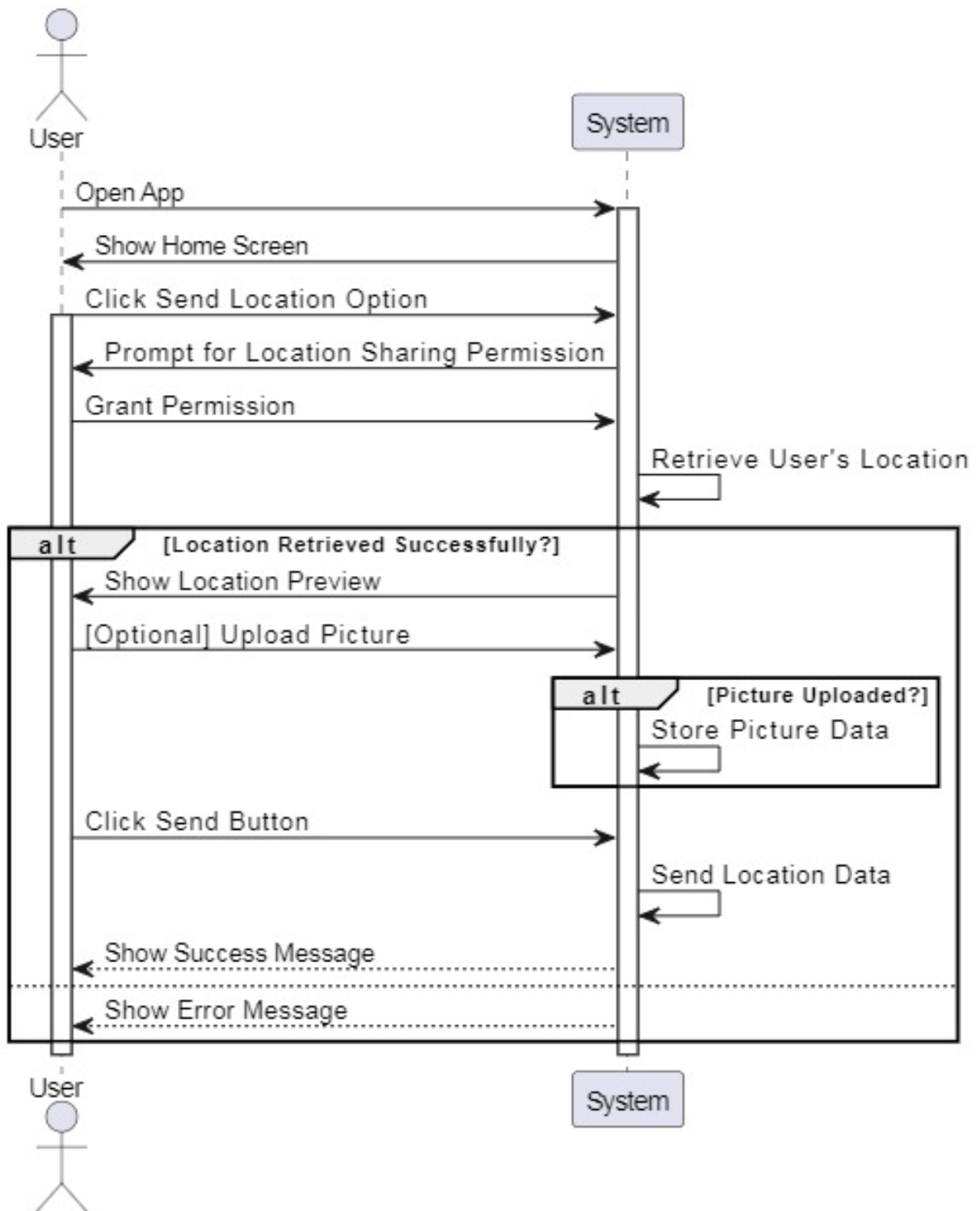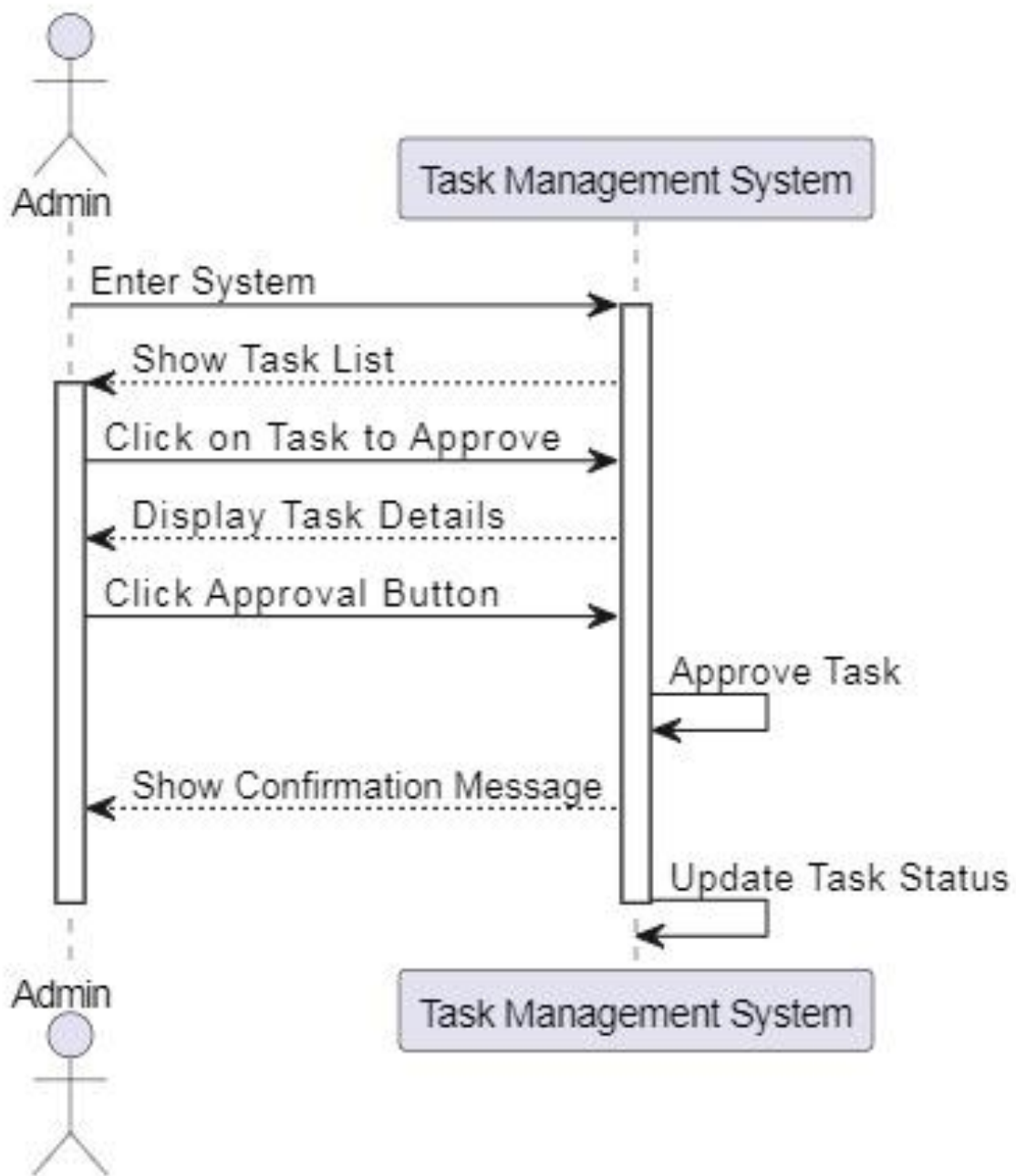
Sequence Diagram (Get Notifications)
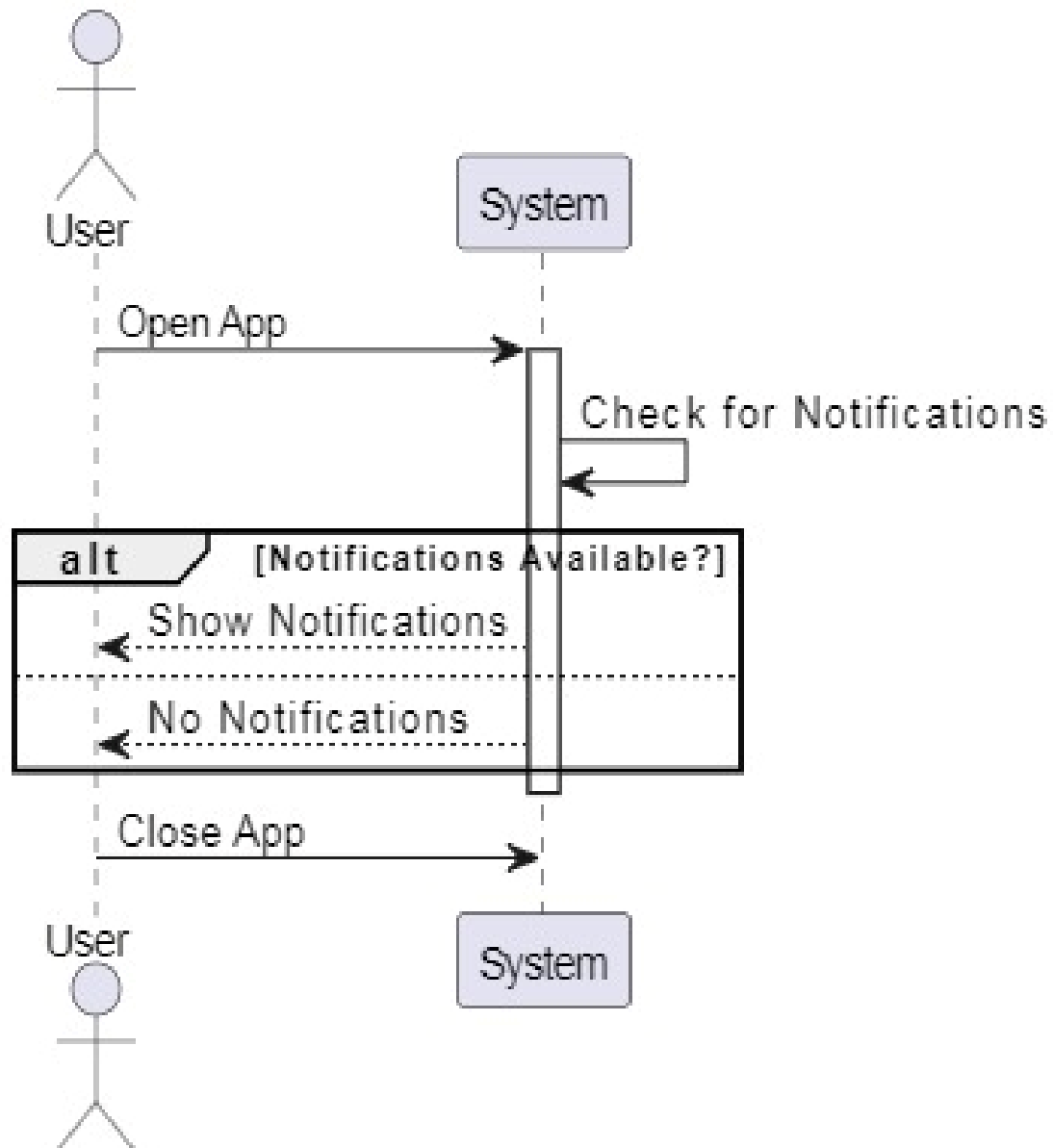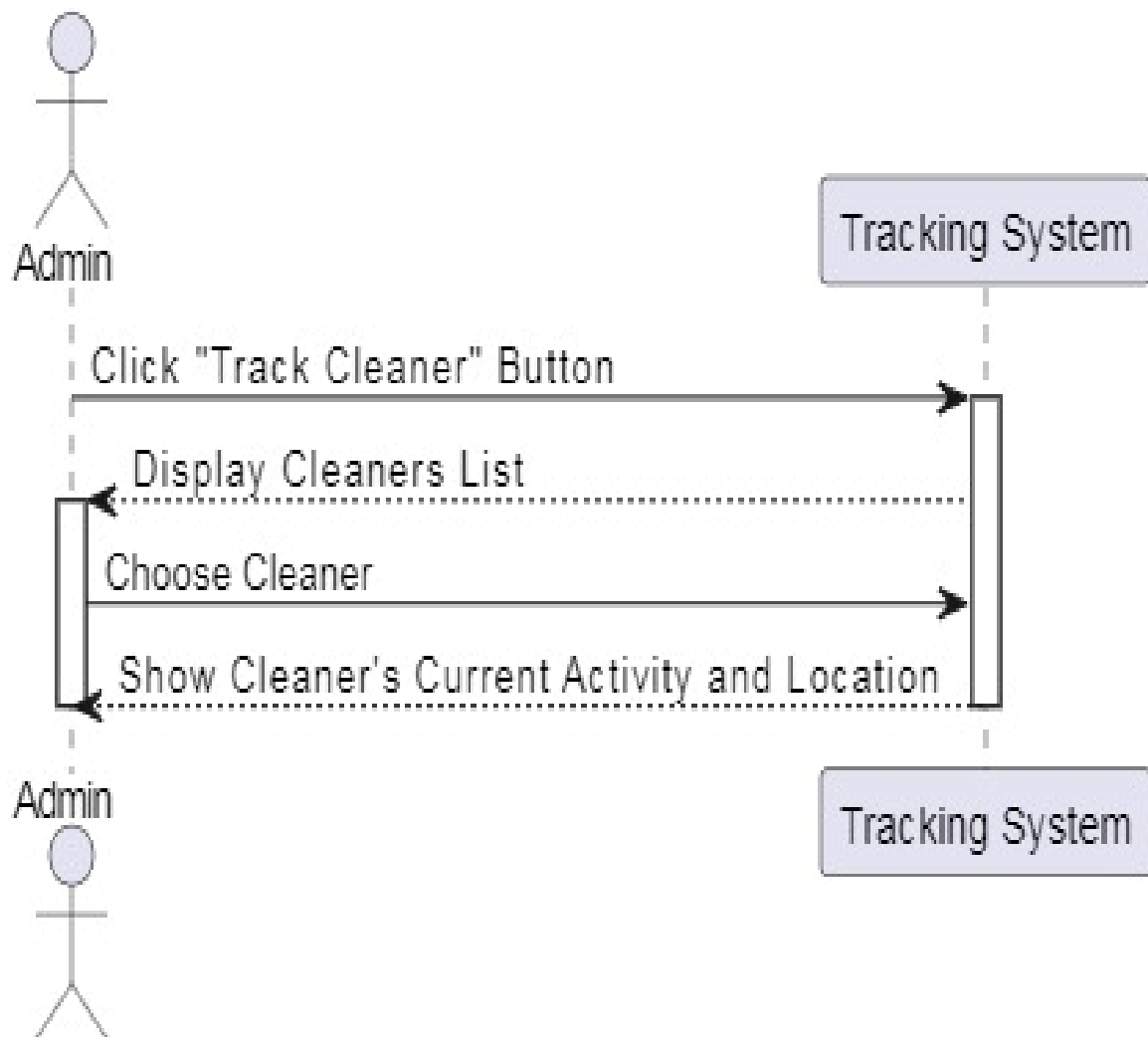
Sequence Diagram (Assign Cleaners)
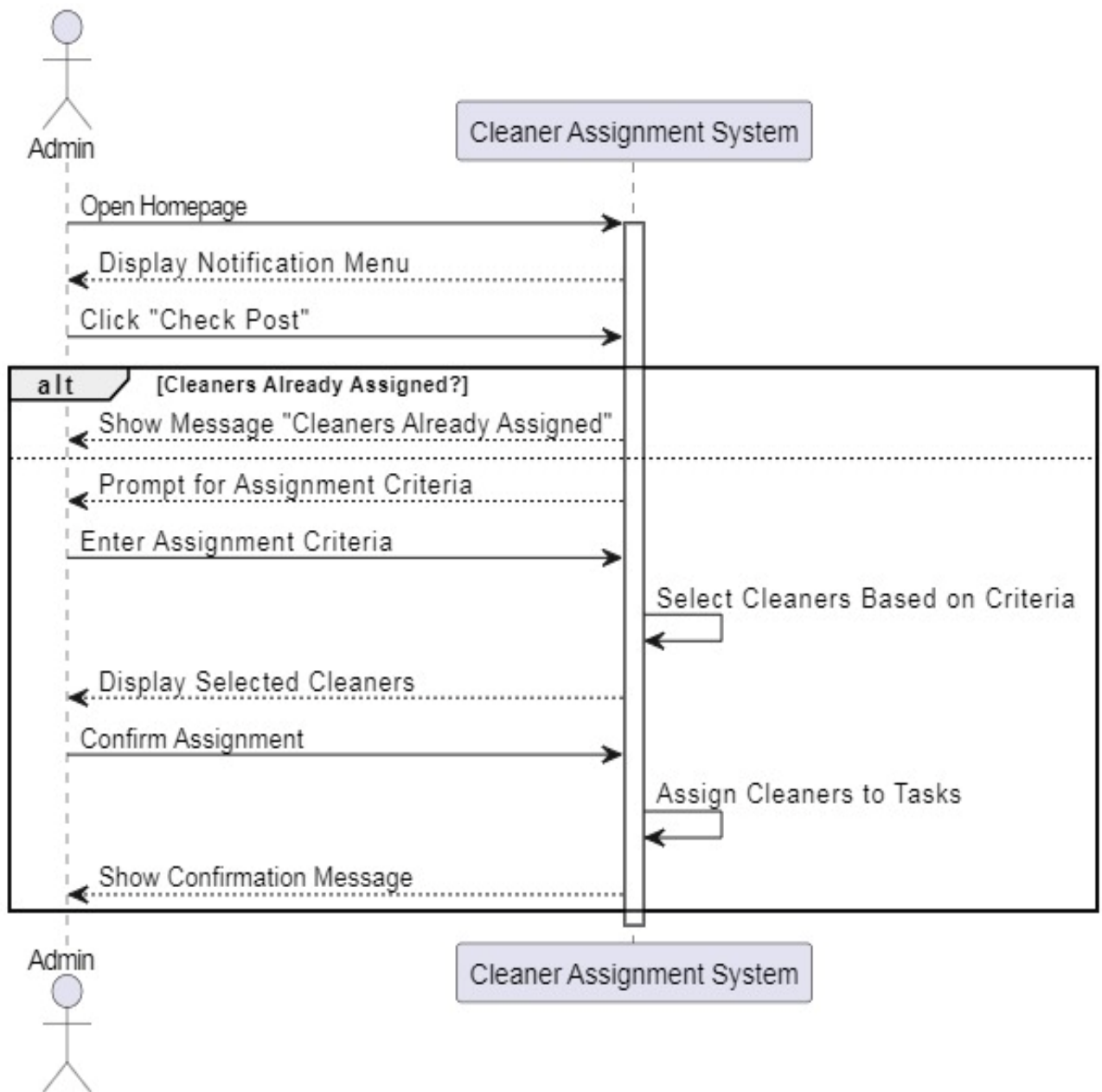
Sequence Diagram (Assign Task)
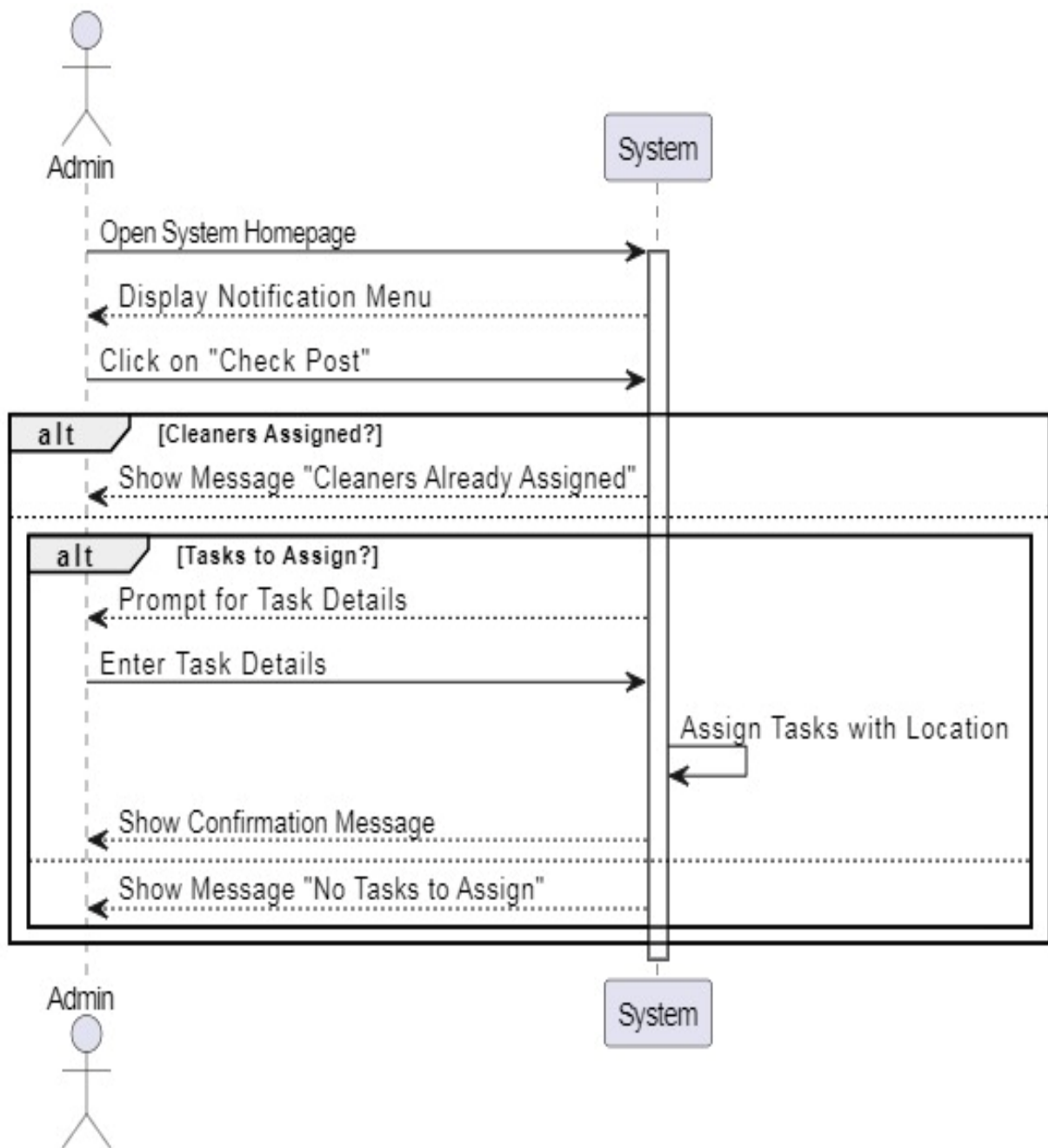
Sequence Diagram (Task Approval)

Sequence Diagram (Track Cleaner)

Sequence Diagram (Report on Completion)