SECJ2203: Software Engineering

System Documentation (SD)

MyGreen UTM

Version 2.0

Date:

8/12/2024

Faculty of MJIIT

Prepared by: <Chinese Student>

Revision Page

Overview

This system documentation contains a system introduction, use case diagram, class diagram and description of use cases.

Target Audience

Development Team

User ： student and UTM Staff

Project Team Members

Member Name	Role	Task	Status
Liu Ruoyang	Programmer	Develop and debug the system. Optimize system functionality.	Project Purpose Document completed. Currently working on system optimization and debugging.
Zhao Wei	System Analyst & Designer	Conduct system analysis. Design the system architecture and user interface.	System analysis and architecture design completed. Currently preparing detailed design documentation and supporting the development team.
Liu Wanpeng	Project Manager	Oversee project planning and execution. Allocate resources and monitor progress.	Project planning and team task allocation completed. Currently monitoring project progress and ensuring tasks align with timelines.
Bu Guoshun	Stakeholder Liaison & Validator	Communicate with stakeholders. Validate the system design.	Initial stakeholder communication completed. Currently validating design compliance with stakeholder requirements and providing feedback.

Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Liu Ruoyang	Project purpose	18/11/2024
2.0	Liu Ruoyang	System Documentation	8/12/2024

Note:
This System Documentation (SD) template is adapted from IEEE Recommended Practice for Software Requirements Specification (SRS) (IEEE Std. 830-1998), Software Design Descriptions (SDD) (IEEE Std. 1016-1998 1), and Software Test Documentation (IEEE Std. 829-2008) that are simplified and customized to meet the need of SECJ2203 course at Faculty of Computing, UTM. Examples of models are from Arlow and Neustadt (2002) and other sources stated accordingly.

Table of Contents

1.Introduction

The "My Green UTM" initiative is to raise environmental awareness and sustainable practices on university campuses. MYGreen UTM aims to use modern technology to promote more active participation of the campus community in environmental protection activities, increase students' environmental awareness and the sustainability of the campus.

2.6 Purpose

The purpose of the SD

1. Provide a detailed account of the system's objectives, and features.
2. Guide the development team and stakeholders in creating a user-centric, efficient, and innovative solution tailored to the needs of the campus community.
3. Ensure all stakeholders have a shared understanding of the system's design, functionalities, and implementation roadmap.

Audience for the SD.

1. UTM Staff: Staff will use the app to organize events, monitor participation, and check how green goals are being achieved.

2. Students: Students will use the app to learn about sustainability, join green events, and track their eco-friendly actions.

3. Development Team: To understand technical requirements, design specifications, and testing procedures for successful system implementation.

2.7 Scope

It is a mobile and stand-alone application designed to enhance environmental awareness and promote sustainability practices within the university campus.

UTM system will

1. Encourage participation: Through points and rankings, students and staff are promoted to participate in environmental activities.
2. Educate users: Provide interactive learning modules on sustainability topics.
3. Strengthening community participation: providing tools for organizing, participating and sharing activities related to sustainable development.

UTM system will NOT DO

Benefits：

1. To promote awareness and adoption of environmental protection measures among students and staff.
2. Enhanced participation in the sustainable development of the campus environment.

Objectives：

1. Create a more sustainable campus environment by fostering a green culture.
2. Provide a platform that integrates education and activity participation.

Goals：

1. Increase the participation of staff and students in environmental activities.
2. To provide the university community with the knowledge and tools to contribute to protecting the environment.
3. Support the long-term sustainability goals of the campus through user-centric and data-driven solutions.

2.8 **Definitions, Acronyms and Abbreviation**

Term	Definition
Emotion Chatbot Analyzer	The software product being developed to analyze the user input in a chatbot conversation and determine the user's emotion based on the language and sentiment used in the conversation.
SRS	System Requirements Specification - a document that outlines the requirements and objectives of the software product being developed.
SDD	System Design Document - a document that describes the system architecture, components, and interfaces in detail.
STD	System Testing Document - a document that outlines the testing process and procedures for the software product being developed.
API	Application Programming Interface - a set of protocols and tools for building software applications.
UI	User Interface - the graphical layout of an application or website, including buttons, menus, and other visual elements.
UX	User Experience - the overall experience a user has while interacting with a product, including ease of use, efficiency, and satisfaction.
NLP	Natural Language Processing - a branch of artificial intelligence that deals with the interaction between computers and human language.
ML	Machine Learning - a type of artificial intelligence that allows software applications to learn and improve from experience without being explicitly programmed.

2.9 **References**

2.10 **Overview**

Describe what the rest of the SD contains

1. System Requirements Specification (SRS):
   Define the functional and non-functional requirements of the system.
2. System Design Documentation (SDD)：
   Describes the system architecture in detail, including back-end and front-end design.
3. System Test Documentation (STD)：
   Describes the test strategies, tools, and methods used to validate the system.

Organization of the System Documentation

3 Specific Requirements

3.1 User characteristics

The MyGreen UTM app is designed for two main user groups: UTM students and UTM staff. This section explains who these users are and how they will use the app, helping the development team make the app user-friendly and suitable for everyone.

2.1.1 UTM Students

Technical Skills:

UTM students are expected to know how to use mobile apps and basic online tools.

Knowledge and Needs:

Students may have little experience with sustainability programs, so the app should be simple and easy to use.

They want to learn more about environmental practices and take part in activities like recycling or energy-saving projects.

How They Use the App:

Students will use the app to learn about sustainability, join green events, and track their eco-friendly actions.

They might also enjoy interactive features, like earning points or rewards for participating.

Accessibility:

The app should be easy to use for all students, including those with disabilities, so features like bigger text or screen reader compatibility might be needed.

2.1.2 UTM Staff

Technical Skills:

UTM staff, including lecturers and office workers, may have moderate experience with apps and online systems. If needed, they can be trained to use the app.

Knowledge and Needs:

Staff members are already involved in sustainability projects, so they will need tools to manage activities, track progress, and help students get involved.

How They Use the App:

Staff will use the app to organize events, monitor participation, and check how green goals are being achieved.

They might also use it to generate reports or share updates about ongoing projects.

Accessibility:

Like students, staff members with disabilities should also find the app easy to use.

Privacy and Security

To use the app, both students and staff will need to register their personal details, such as name and UTM email. The app will protect this information with strong security measures to ensure it stays private and safe.

3.2 System Features

The MyGreen UTM is a software system designed to operate on mobile devices, including both Android and iOS operating systems. The system encourages sustainable practices among UTM students and staff by providing an interactive platform to participate in green activities, log their contributions, and earn points for rewards. The system simplifies event participation and tracking through QR code scanning and real-time data updates, supported by a cloud-based database to ensure a seamless user experience.

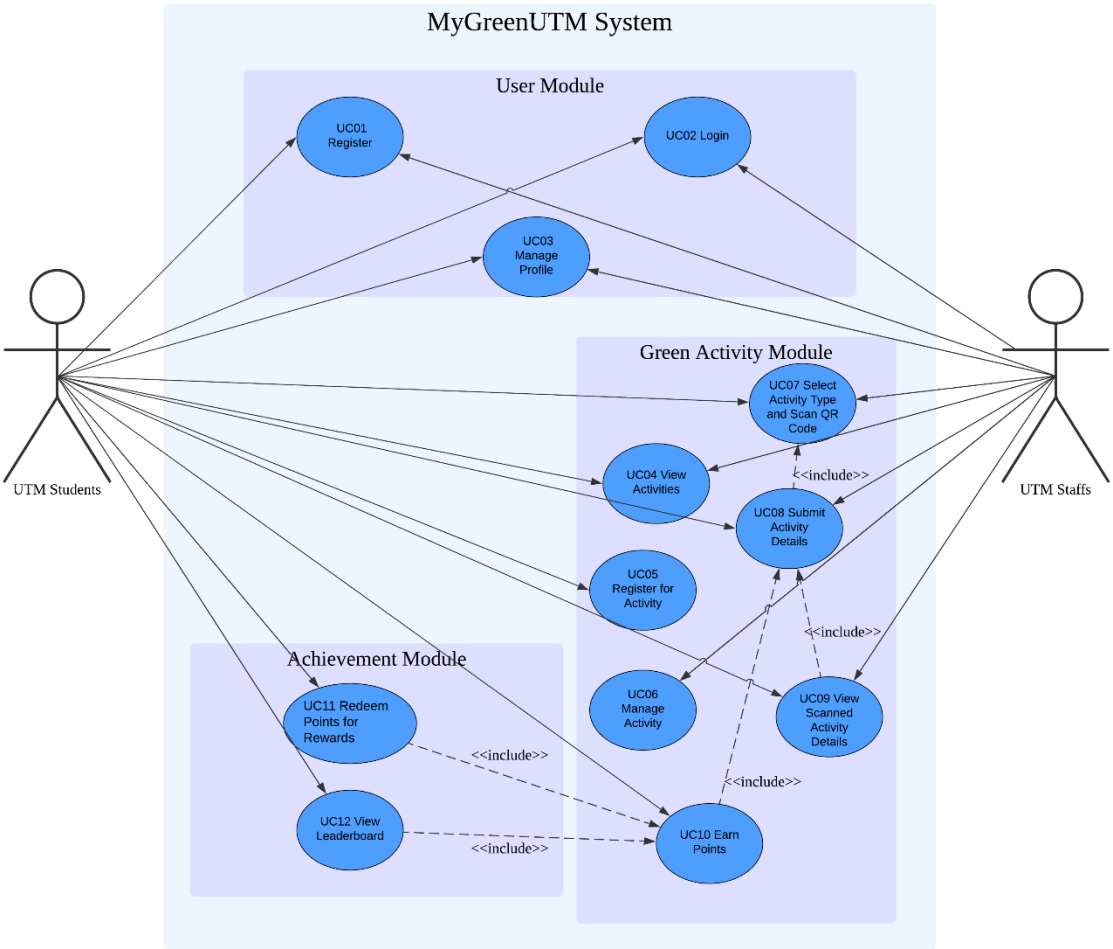


Figure 1: Use Case Diagram for <MyGreen UTM of the System>

Table 1: Description of Module and Functions for <MyGreen UTM of the System>

Module	Function	Description
User Module	UC01 Register	Users register an account using their UTM email address. They provide basic information such as name and email.
User Module	UC02 Login	Users log in using their registered email and password to access the app's features.
User Module	UC03 Manage Profile	Users can update their personal information, such as changing their email, password, or profile picture.
Green Activity Module	UC04 View Activities	Users can browse upcoming events and activities related to sustainability, such as recycling campaigns or workshops.
Green Activity Module	UC05 Register for Activity	Students can sign up for green events or programs directly through the app.
Green Activity Module	UC06 Manage Activity	Staff can create and manage green events, including updating event details and tracking participant lists.
Green Activity Module	UC07 Select Activity Type and Scan QR Code	Users can select an activity type (e.g., awareness, education, trash management), then open their camera to scan a QR code provided at the event.
Green Activity Module	UC08 Submit Activity Details	After scanning the QR code, users can input their user type (e.g., student, staff), choose the event type, and select the location. Once submitted, the app will confirm the submission with a success message.
Green Activity Module	UC09 View Scanned Activity Details	After submitting the details, users can view a summary screen showing the activity information, such as event name, type, location, date, and points earned. Users can also take a screenshot of the page for record-keeping.
Green Activity Module	UC10 Earn Points	Students earn points for every successful activity submission. Points are accumulated based on the type of activity and its contribution to sustainability goals (e.g., 300 points for attending a recycling event). Points contribute toward unlocking badges and rewards.
Achievement Module	UC11 Redeem Points for Rewards	Students can use their accumulated points to redeem various rewards offered within the app. Rewards can include items that align with sustainable practices, such as reusable products.
Achievement Module	UC12 View Leaderboard	Students can view a leaderboard ranking the most active participants in green activities, encouraging friendly competition.

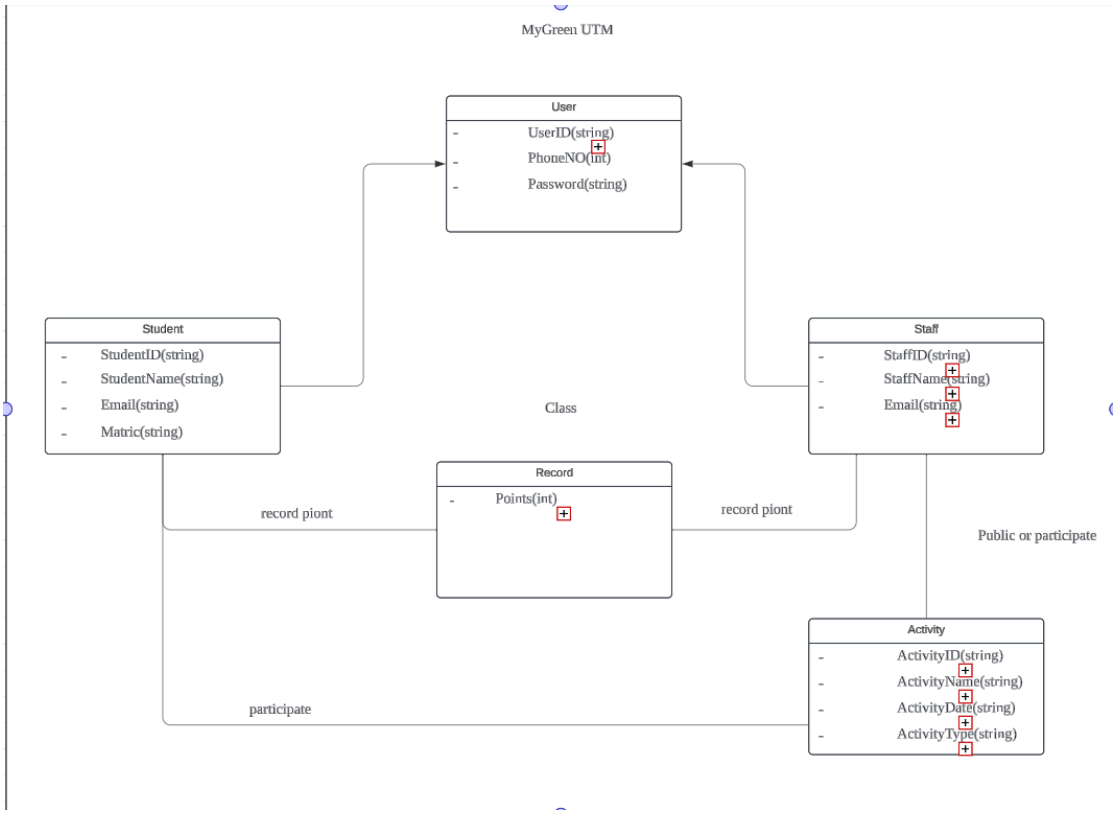


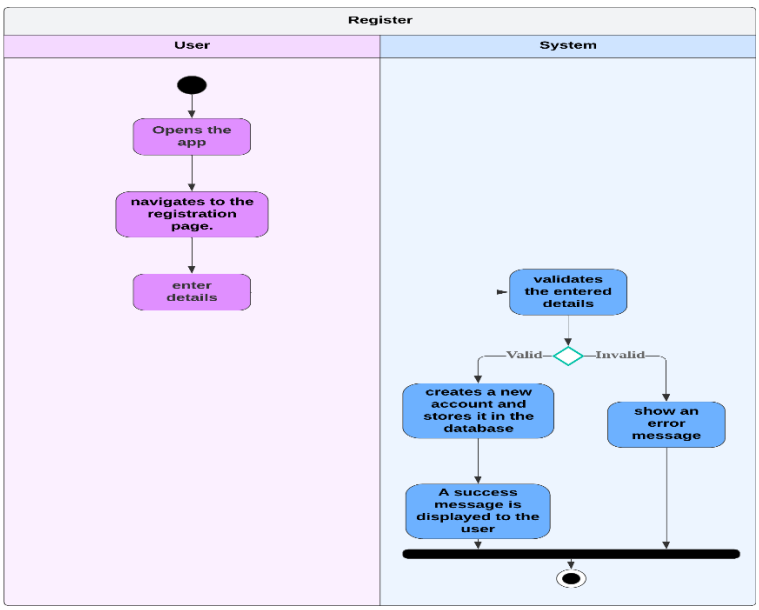
Figure2: Domain Model for <MyGreen UTM of the System>

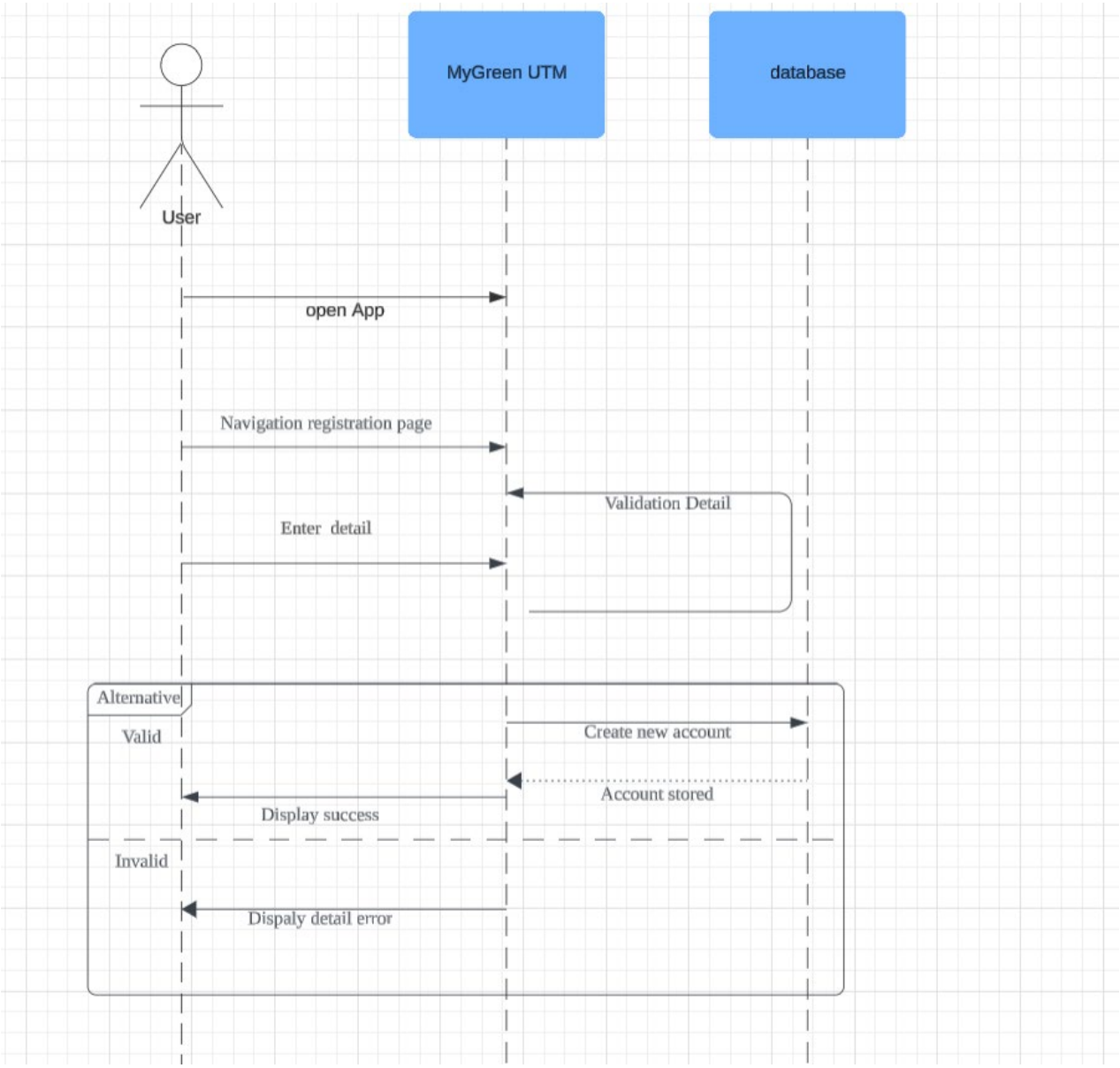
3.3 Use Case Detail

3.3.1 UC01: Use Case <Register>

2.3.1 UC01: Use Case <Register>

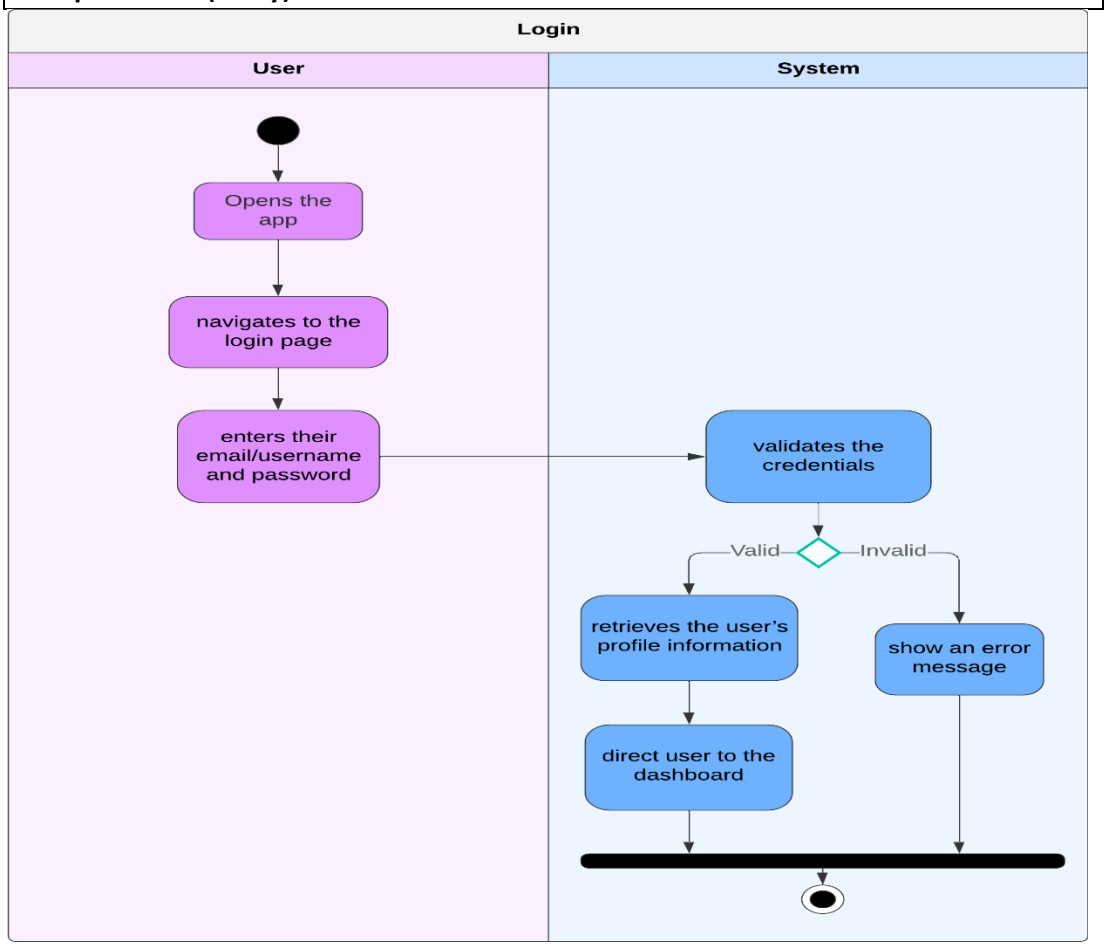
Use case: Register
ID: UC01
Actors: UTM Student, UTM Staff
Extends: <Register with UTM Student Account>
Preconditions:
Flow of events: 1. The user opens the app. 2. The user navigates to the registration page. 3. The user enters their details (e.g., name, email, password). 4. The system validates the entered details: 4.1 If valid, proceed to Step 6. 4.2 If invalid, show an error message. **(AF1)** 5. The system creates a new account and stores it in the database. 6. A success message is displayed to the user.
Postconditions: The user's account is successfully registered and saved in the system.
Alternative flow 1: If the entered details are invalid, the system displays an error message and prompts the user to correct them.

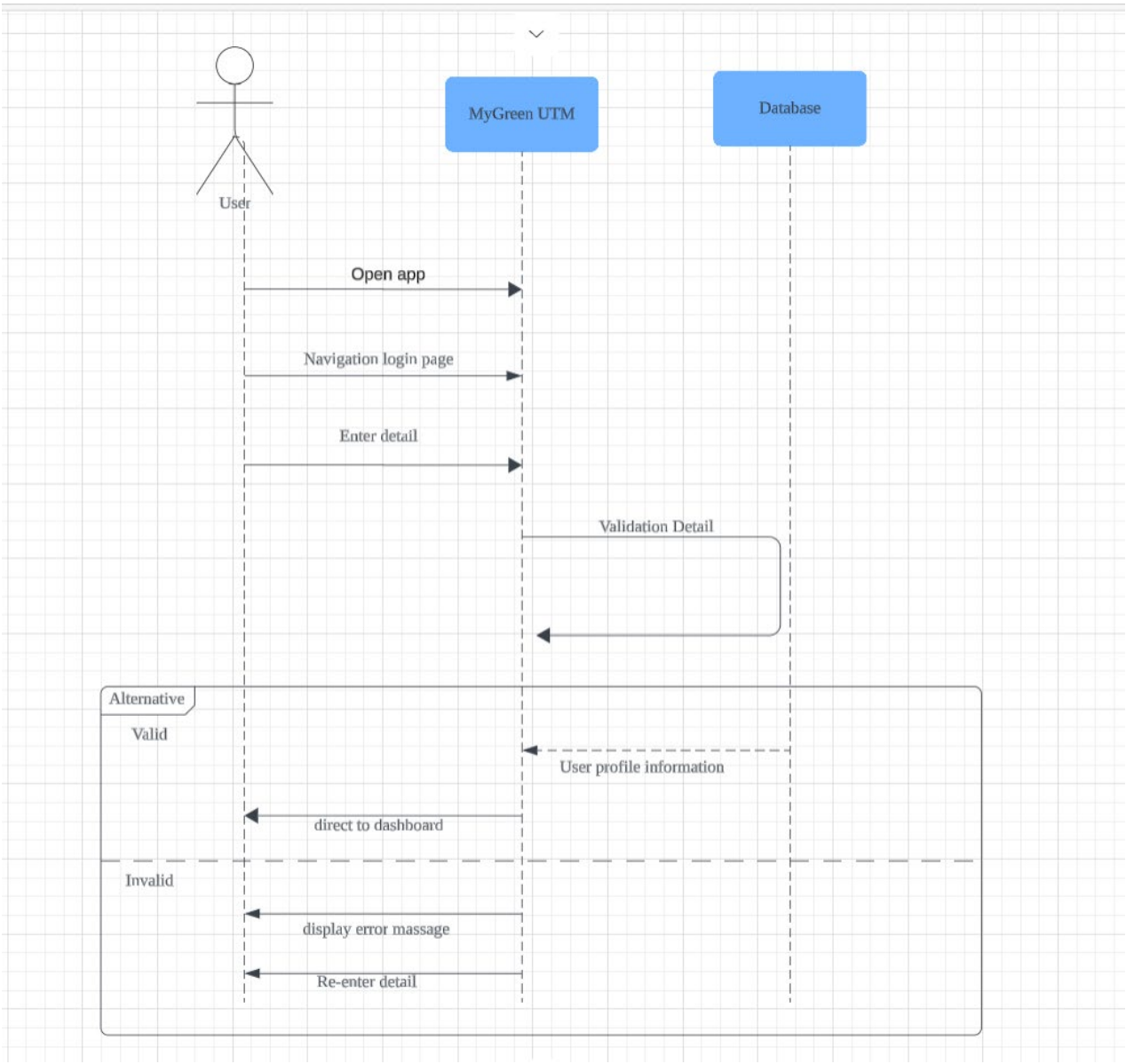



MyGreen UTM
database
User
open App
Navigation registration page
Validation Detail
Enter detail
Alternative
Valid
Create new account
Account stored
Display success
Invalid
Dispaly detail error

3.3.2 UC02: Use Case <Login>

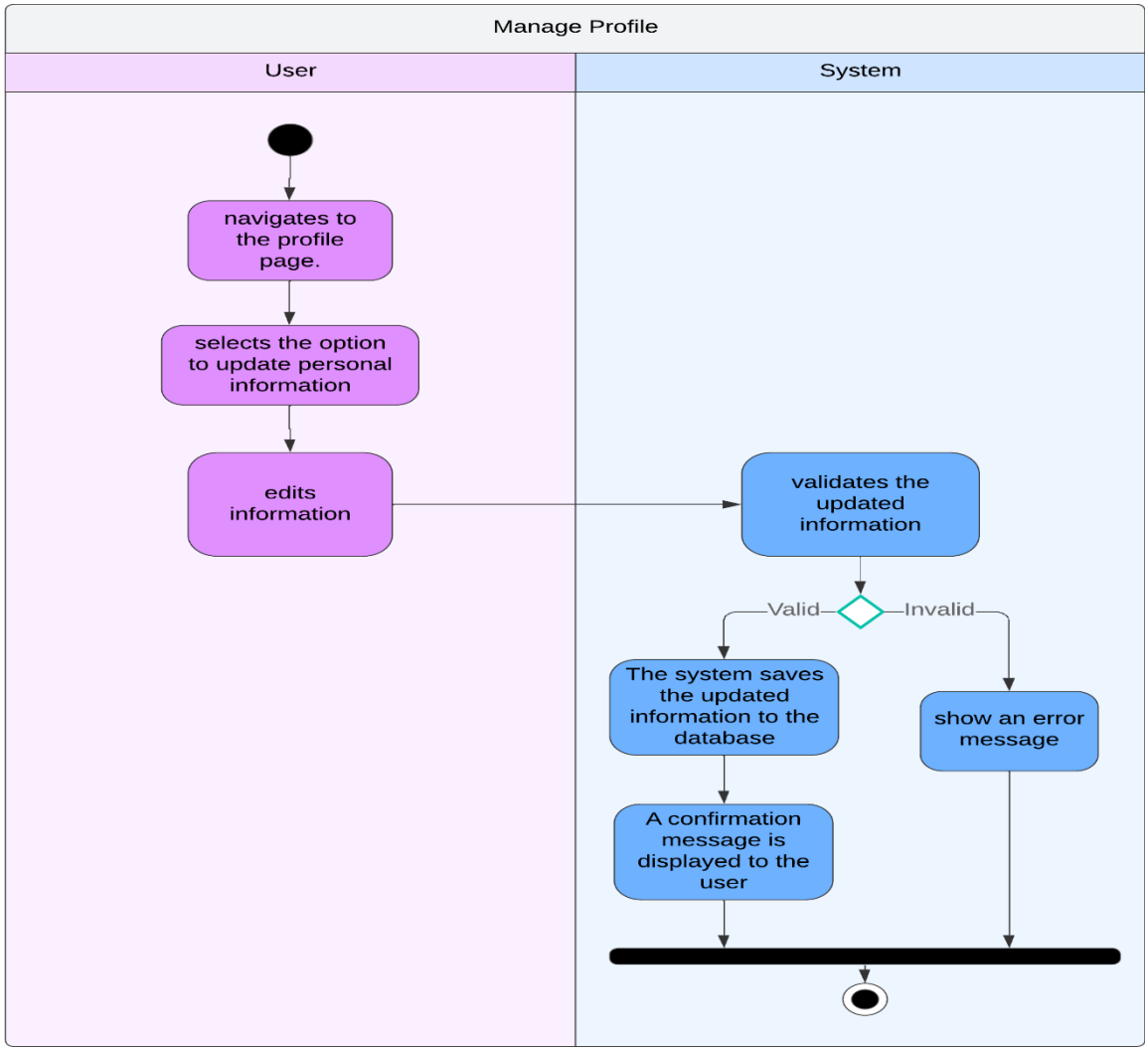
Use case: Login
ID: UC02
Actors: UTM Student, UTM Staff
Preconditions: The user must have already registered an account in the system.
Flow of events: 1. The user opens the app. 2. The user navigates to the login page. 3. The user enters their email/username and password. 4. The system validates the login credentials: 4.1 If the credentials are valid, proceed to Step 5. 4.2 If the credentials are invalid, show an error message. (AF1) 5. The system retrieves the user's profile information. 6. The user is directed to the dashboard.
Postconditions: The user is successfully logged into the system and can access the dashboard.
Alternative flow 1: If the login credentials are incorrect, the system displays an error message and prompts the user to re-enter them.
Exception flow (if any):

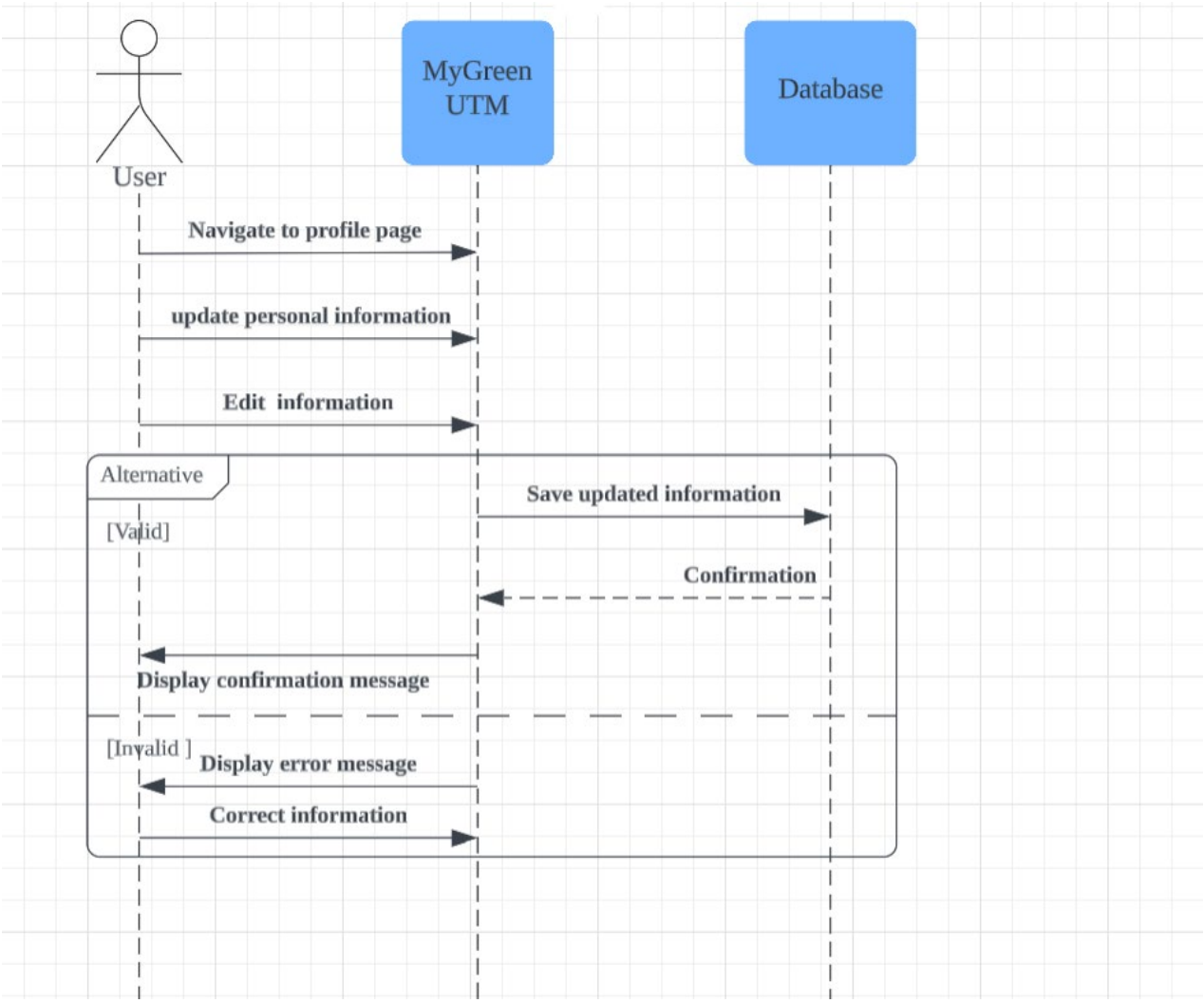



User
MyGreen UTM
Database
Open app
Navigation login page
Enter detail
Validation Detail
Alternative
Valid
User profile information
direct to dashboard
Invalid
display error massage
Re-enter detail

3.3.3 **UC03: Use Case <Manage Profile>**

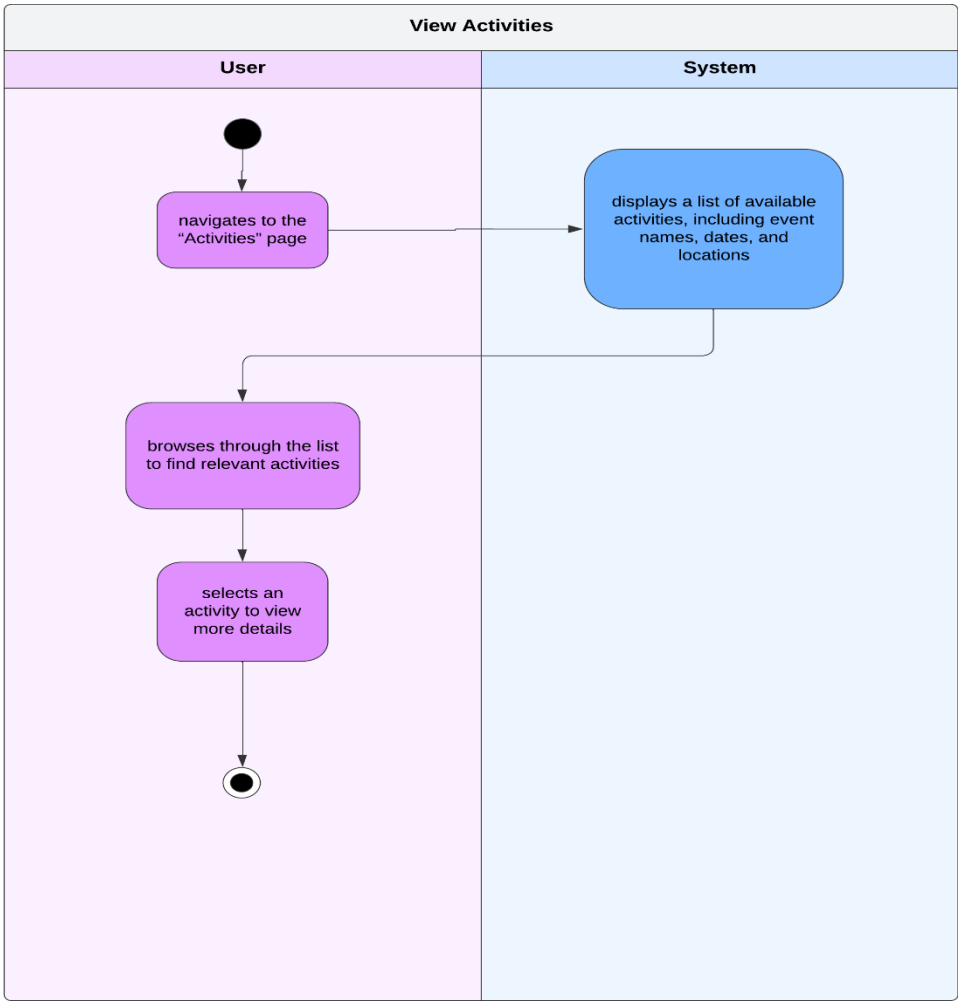
Use Case: Manage Profile
ID: UC03
Actors: UTM Students, UTM Staff
Preconditions: The user must be logged into the system.
Flow of Events: 1. The user navigates to the profile page. 2. The user selects the option to update their personal information. 3. The user edits their information (e.g., name, email, password). 4. The system validates the updated information: 4.1 If the information is valid, proceed to Step 5. 4.2 If the information is invalid, show an error message. (AF1) 5. The system saves the updated information to the database. 6. A confirmation message is displayed to the user.
Postconditions: The user's profile is successfully updated and saved in the system.
Alternative Flow 1: If the updated information is invalid, the system displays an error message and prompts the user to correct it.
Exception Flow:

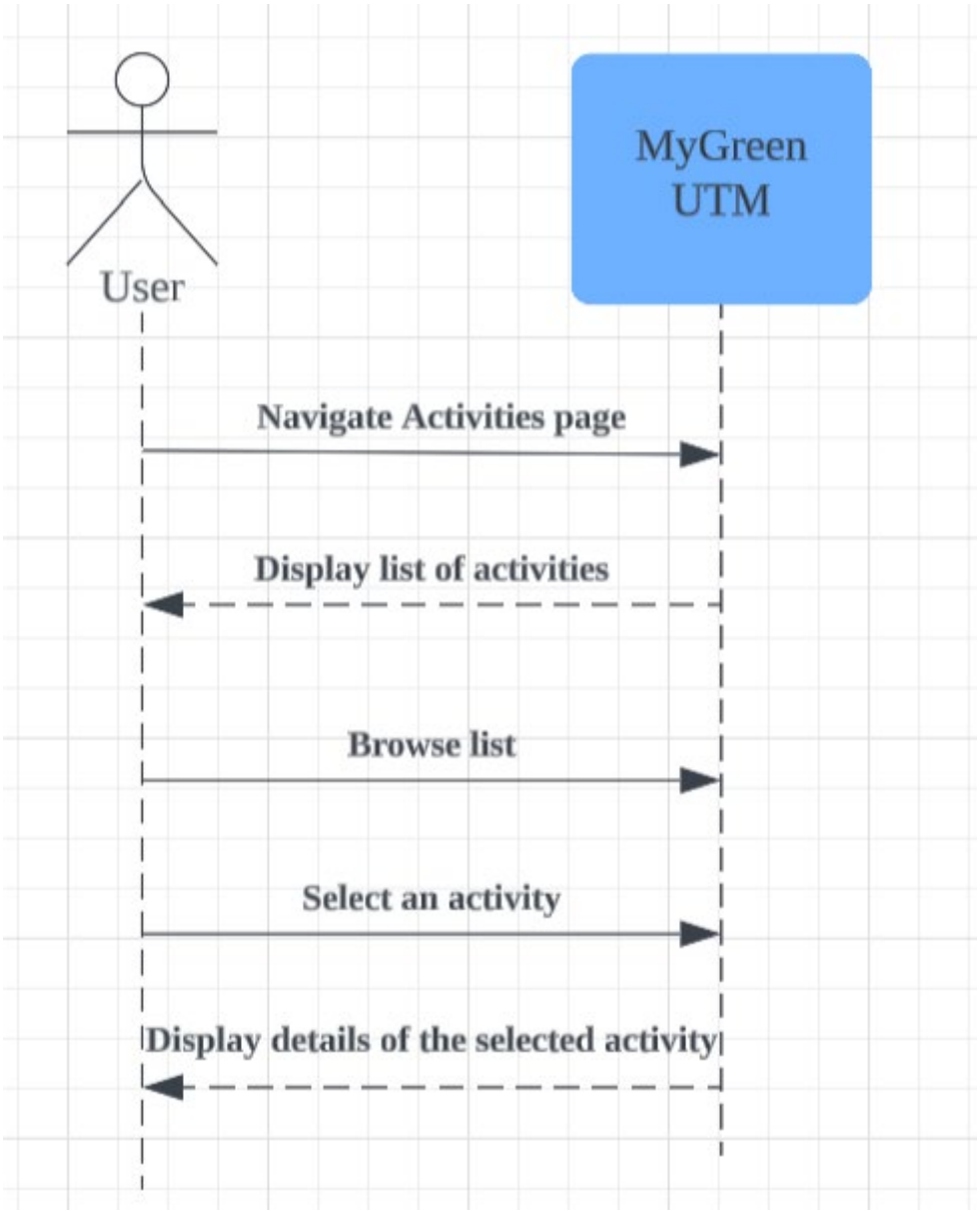

Manage Profile
User
System
navigates to the profile page.
selects the option to update personal information
edits information
validates the updated information
Valid
Invalid
The system saves the updated information to the database
show an error message
A confirmation message is displayed to the user


User
MyGreen UTM
Database
Navigate to profile page
update personal information
Edit information
Alternative
[Valid]
Save updated information
Confirmation
Display confirmation message
[Invalid]
Display error message
Correct information

3.3.4 UC04: Use Case <View Activities>

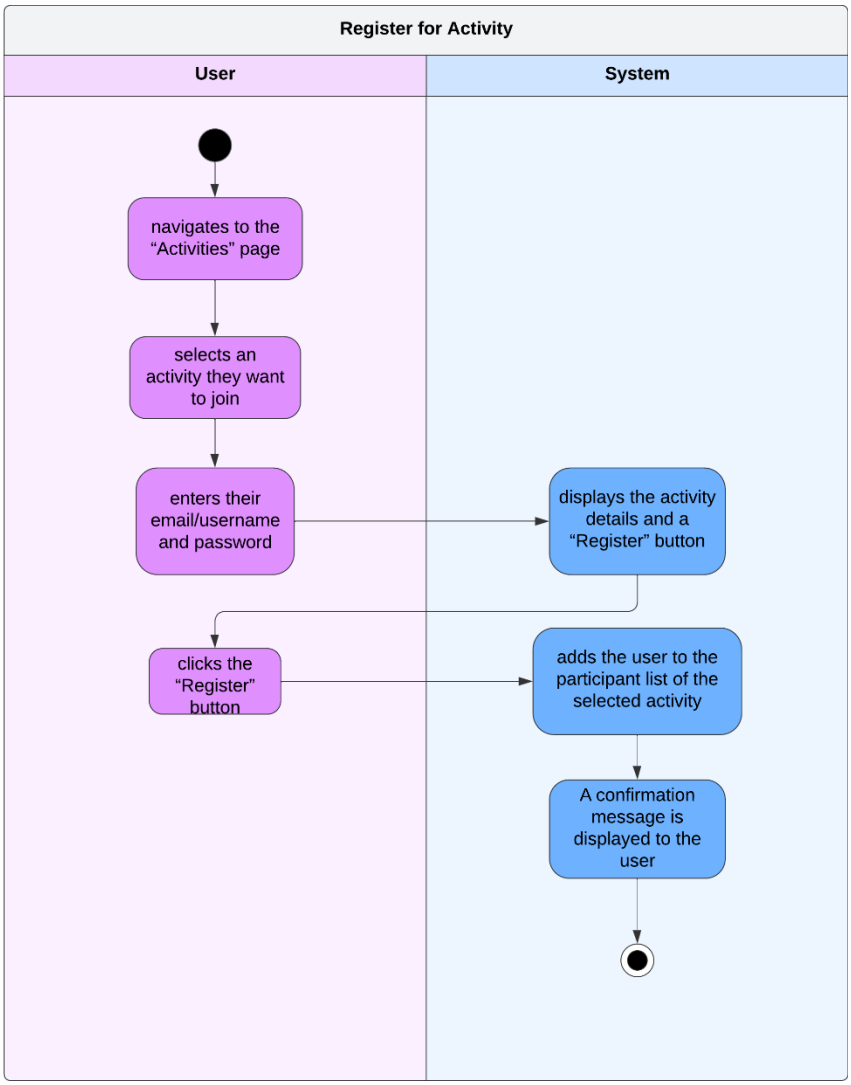
Use Case: View Activities
ID: UC04
Actors: UTM Students, UTM Staff
Preconditions: The user must be logged into the system.
Flow of Events: 1. The user navigates to the “Activities” page. 2. The system displays a list of available activities, including event names, dates, and locations. 3. The user browses through the list to find relevant activities. 4. The user selects an activity to view more details.
Postconditions: The user can successfully view details of the selected activity.
Alternative Flow:
Exception Flow:

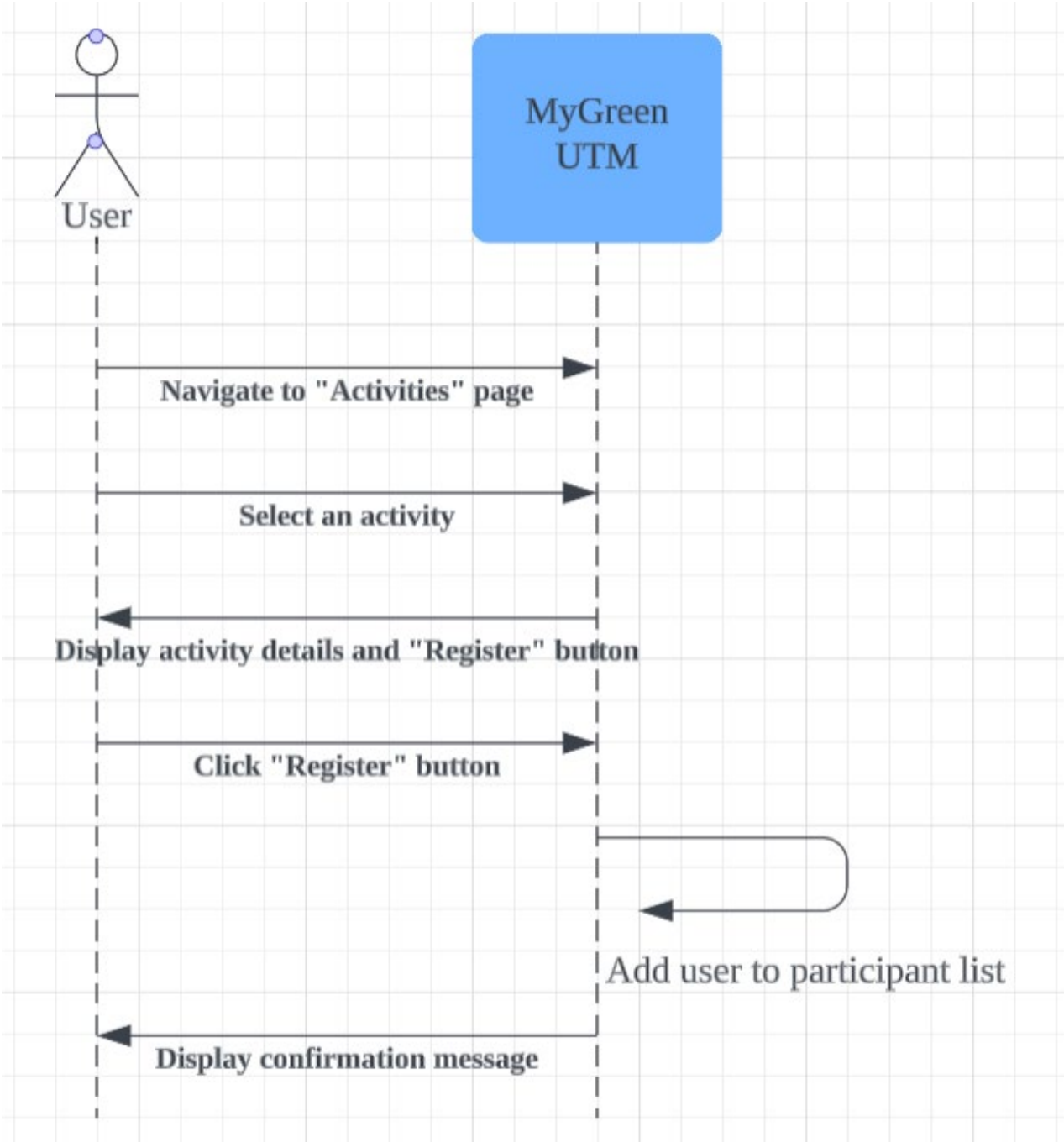



User
MyGreen
UTM
Navigate Activities page
Display list of activities
Browse list
Select an activity
Display details of the selected activity

3.3.5 UC05: Use Case <Register for Activity>

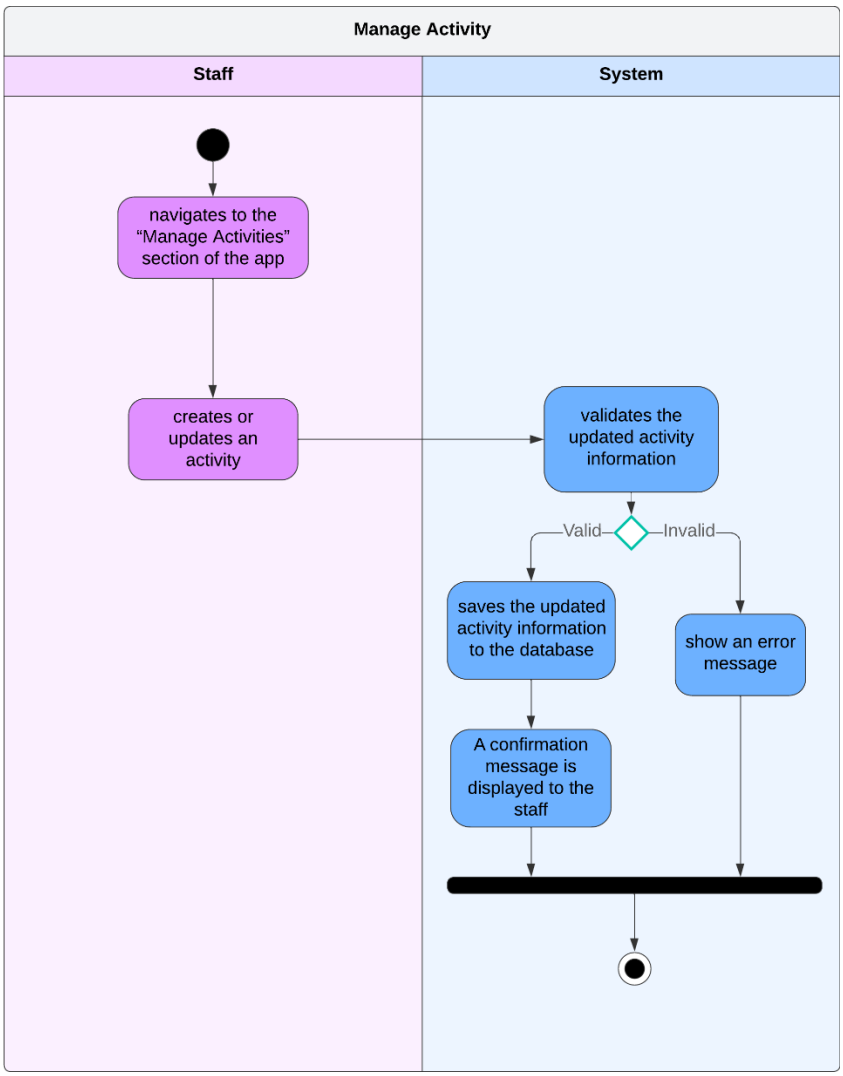
Use Case: Register for Activity
ID: UC05
Actors: UTM Students
Preconditions: The user must be logged into the system.
Flow of Events: 1. The user navigates to the “Activities” page. 2. The user selects an activity they want to join. 3. The system displays the activity details and a “Register” button. 4. The user clicks the “Register” button. 5. The system adds the user to the participant list of the selected activity. 6. A confirmation message is displayed to the user.
Postconditions: The user is successfully registered for the selected activity.
Alternative Flow:
Exception Flow:

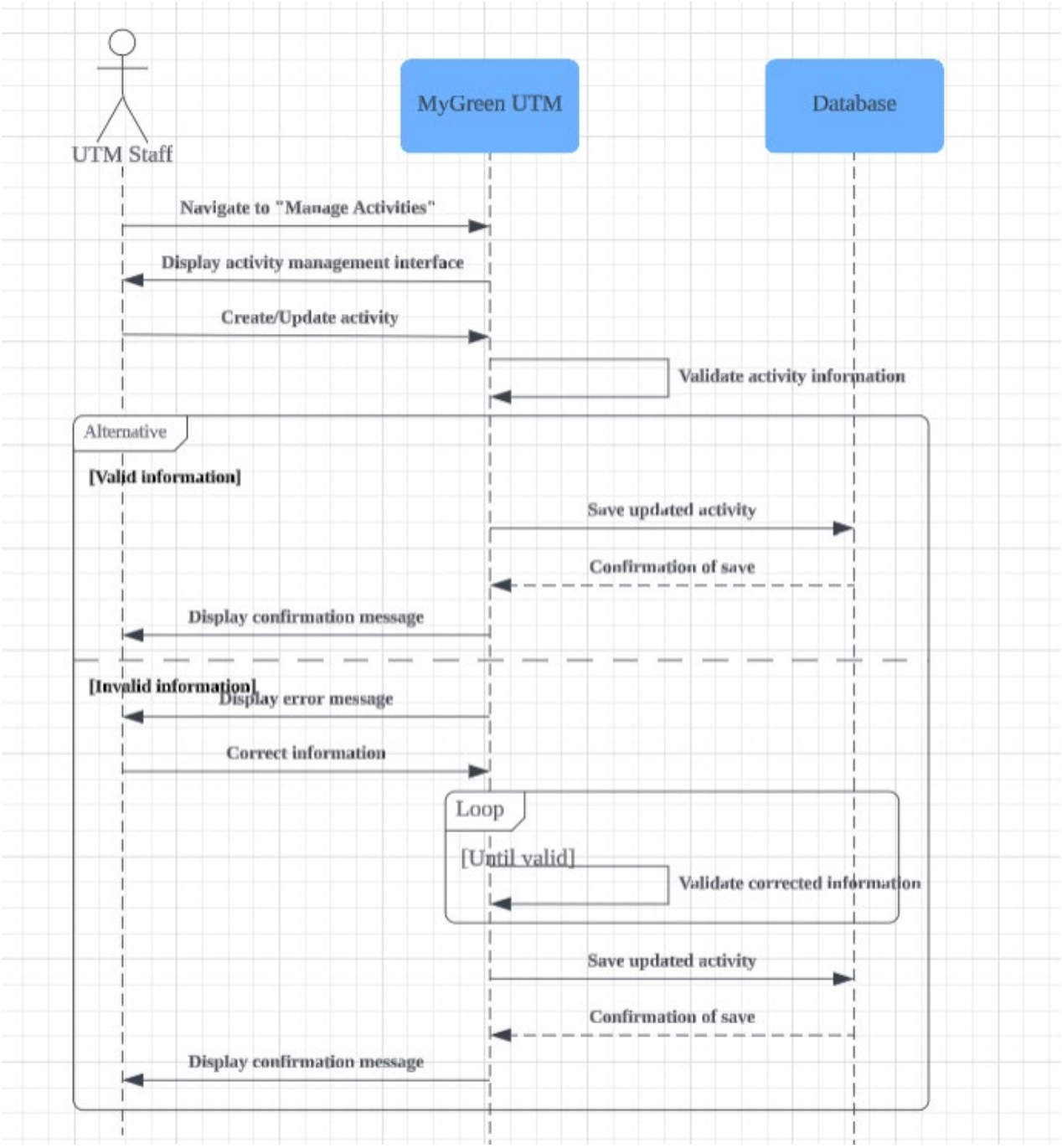



User
MyGreen UTM
Navigate to "Activities" page
Select an activity
Display activity details and "Register" button
Click "Register" button
Add user to participant list
Display confirmation message

3.3.6 UC06: Use Case <Manage Activity>

Use Case: Manage Activity
ID: UC06
Actors: UTM Staff
Preconditions: The user must be logged in with staff privileges.
Flow of Events: 1. The staff navigates to the “Manage Activities” section of the app. 2. The staff creates or updates an activity (e.g., event name, date, time, location). 3. The system validates the updated activity information: 3.1 If valid, proceed to Step 4. 3.2 If invalid, show an error message. (AF1) 4. The system saves the updated activity information to the database. 5. A confirmation message is displayed to the staff.
Postconditions: The activity information is successfully created or updated in the system.
Alternative Flow 1: If the input data is invalid, the system displays an error message and prompts the staff to correct it.
Exception Flow:

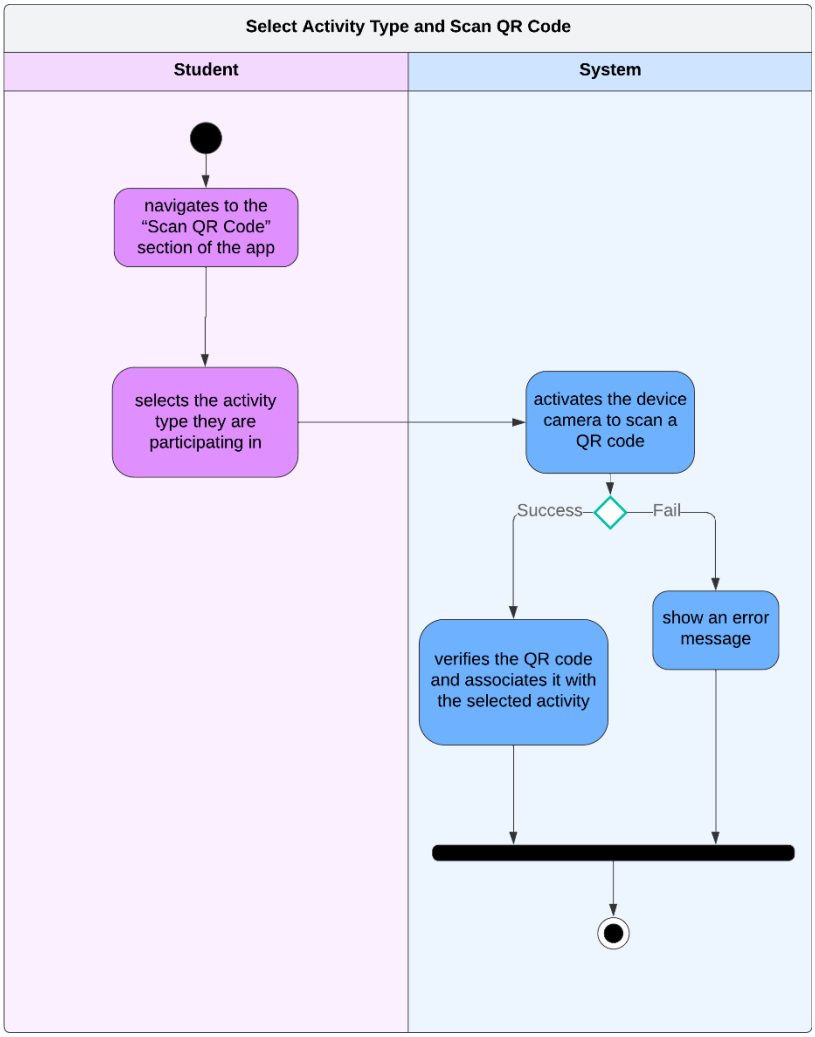


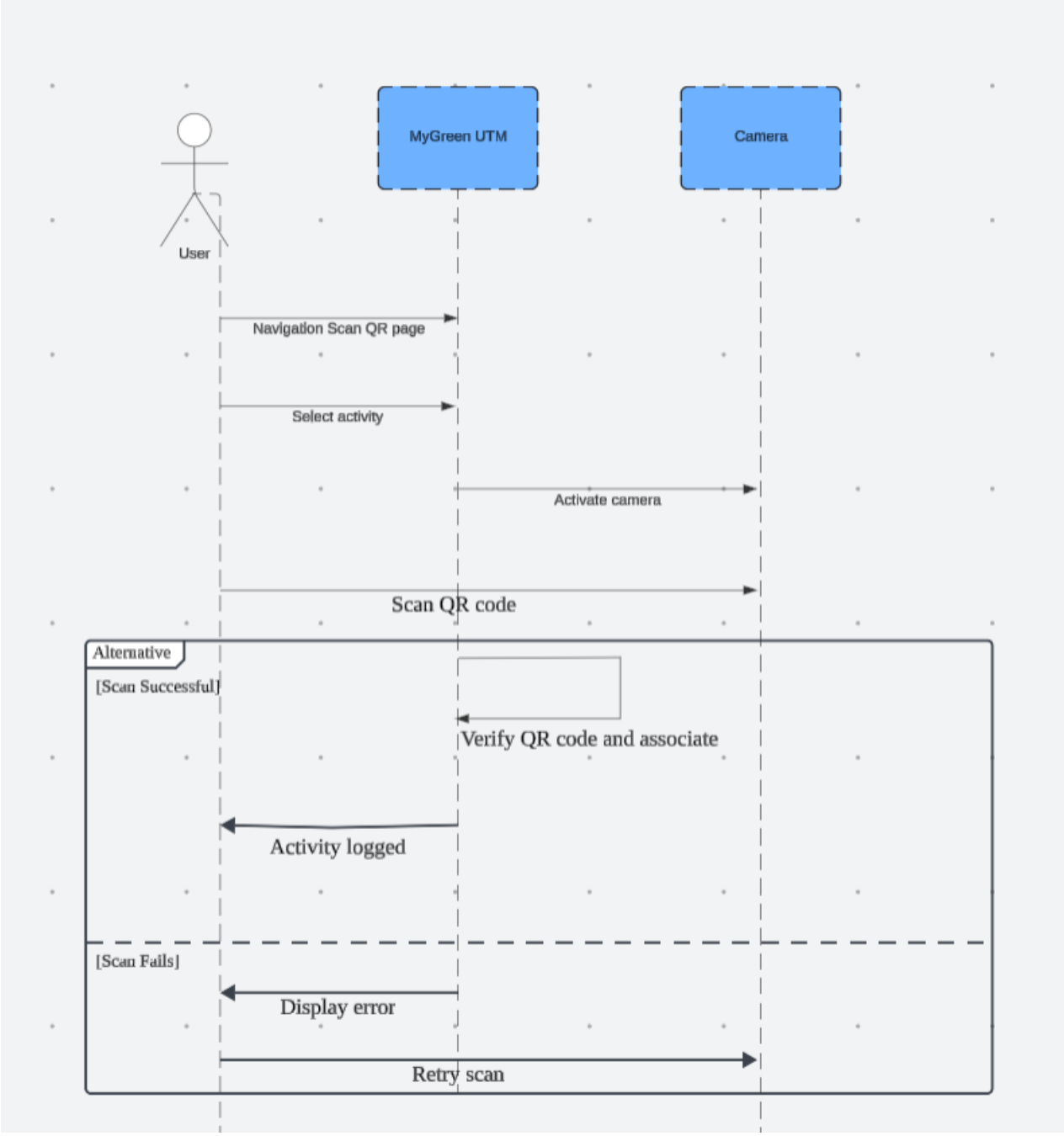

UTM Staff
MyGreen UTM
Database
Navigate to "Manage Activities"
Display activity management interface
Create/Update activity
Validate activity information
Alternative
[Valid information]
Save updated activity
Confirmation of save
Display confirmation message
[Invalid information]
Display error message
Correct information
Loop
[Until valid]
Validate corrected information
Save updated activity
Confirmation of save
Display confirmation message

ss

3.3.7 UC07: Use Case <Select Activity Type and Scan QR Code>

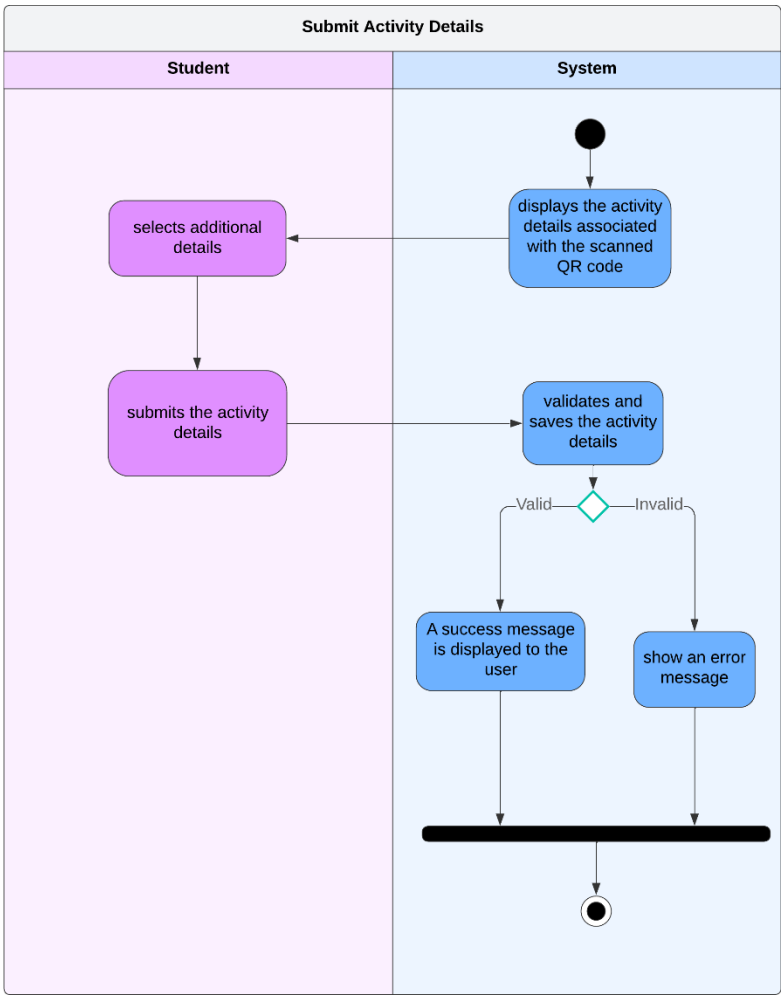
Use Case: Select Activity Type and Scan QR Code
ID: UC07
Actors: UTM Students, UTM Staff
Preconditions: The user must be logged in.
Flow of Events: 1. The user navigates to the “Scan QR Code” section of the app. 2. The user selects the activity type they are participating in (e.g., awareness, recycling). 3. The system activates the device camera to scan a QR code. 4. The user scans the QR code provided at the event: 4.1 If the scan is successful, proceed to Step 5. 4.2 If the scan fails, show an error message. (AF1) 5. The system verifies the QR code and associates it with the selected activity.
Postconditions: The QR code is successfully scanned, and the activity is logged in to the system.
Alternative Flow 1: If the QR code scan fails, the system displays an error message and prompts the user to retry.
Exception Flow:

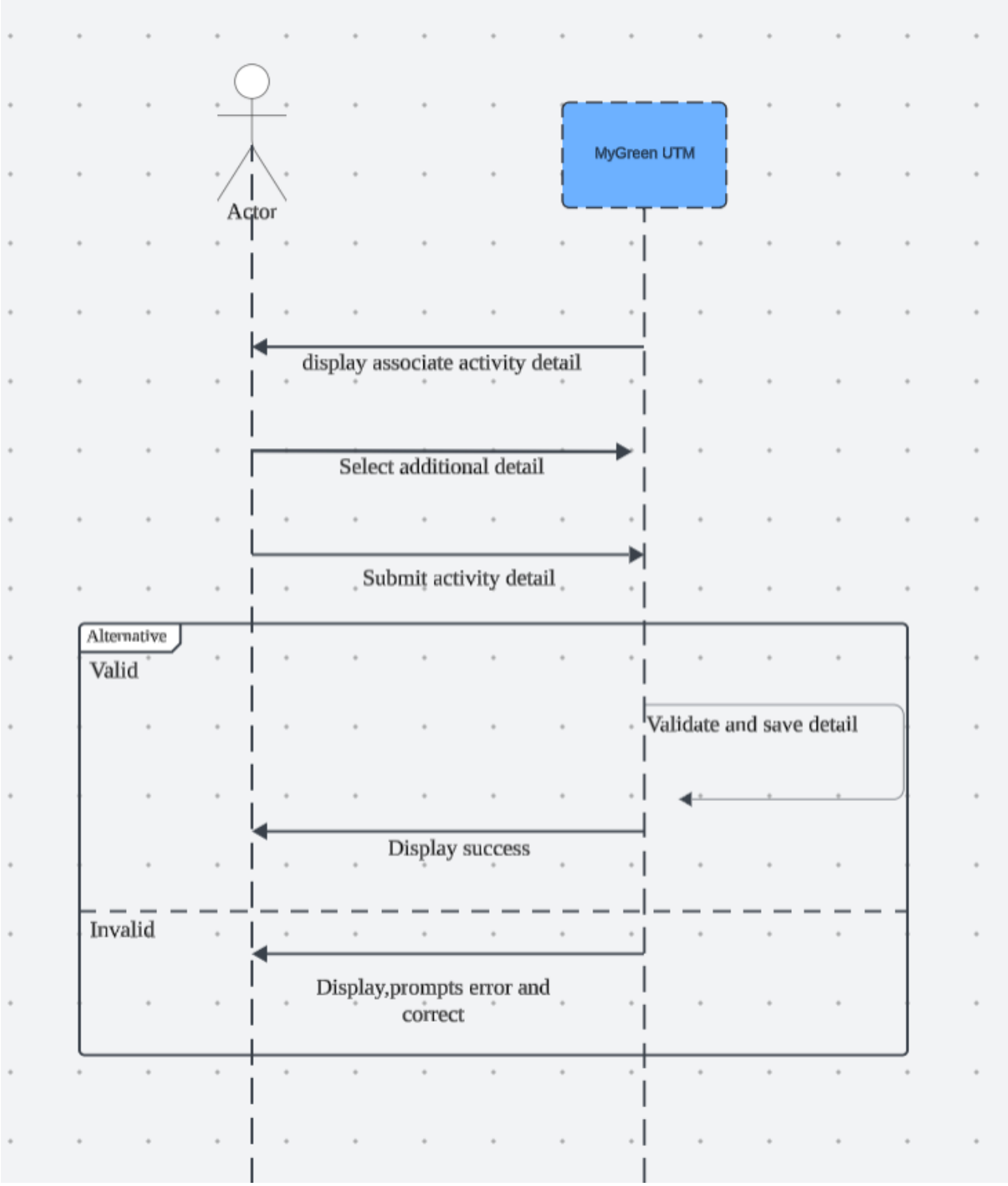

Select Activity Type and Scan QR Code
Student
System
navigates to the "Scan QR Code" section of the app
selects the activity type they are participating in
activates the device camera to scan a QR code
Success
Fail
verifies the QR code and associates it with the selected activity
show an error message


MyGreen UTM
Camera
User
Navigation Scan QR page
Select activity
Activate camera
Scan QR code
Alternative
[Scan Successful]
Verify QR code and associate
Activity logged
[Scan Fails]
Display error
Retry scan

3.3.8 UC08: Use Case <Submit Activity Details>

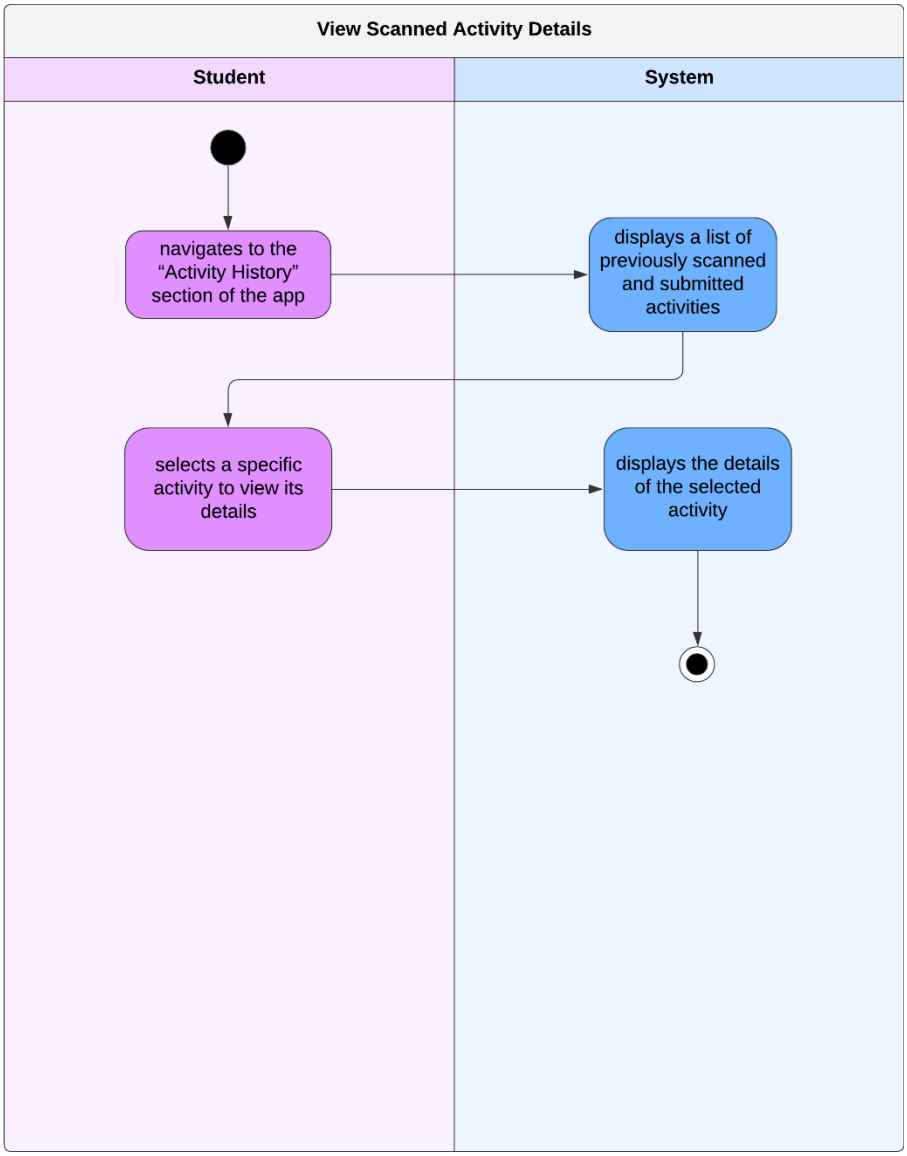
Use Case: Submit Activity Details
ID: UC08
Actors: UTM Students, UTM Staff
Includes: <UC07: Select Activity Type and Scan QR Code>
Preconditions: The user must have successfully scanned a QR code.
Flow of Events: 1. The system displays the activity details associated with the scanned QR code. 2. The user selects additional details (e.g., location, user type). 3. The user submits the activity details. 4. The system validates and saves the activity details: 4.1 If valid, proceed to Step 5. 4.2 If invalid, show an error message. (AF1) 5. A success message is displayed to the user.
Postconditions: The activity details are successfully submitted and saved in the system.
Alternative Flow 1: If the entered details are invalid, the system displays an error message and prompts the user to correct them.
Exception Flow:

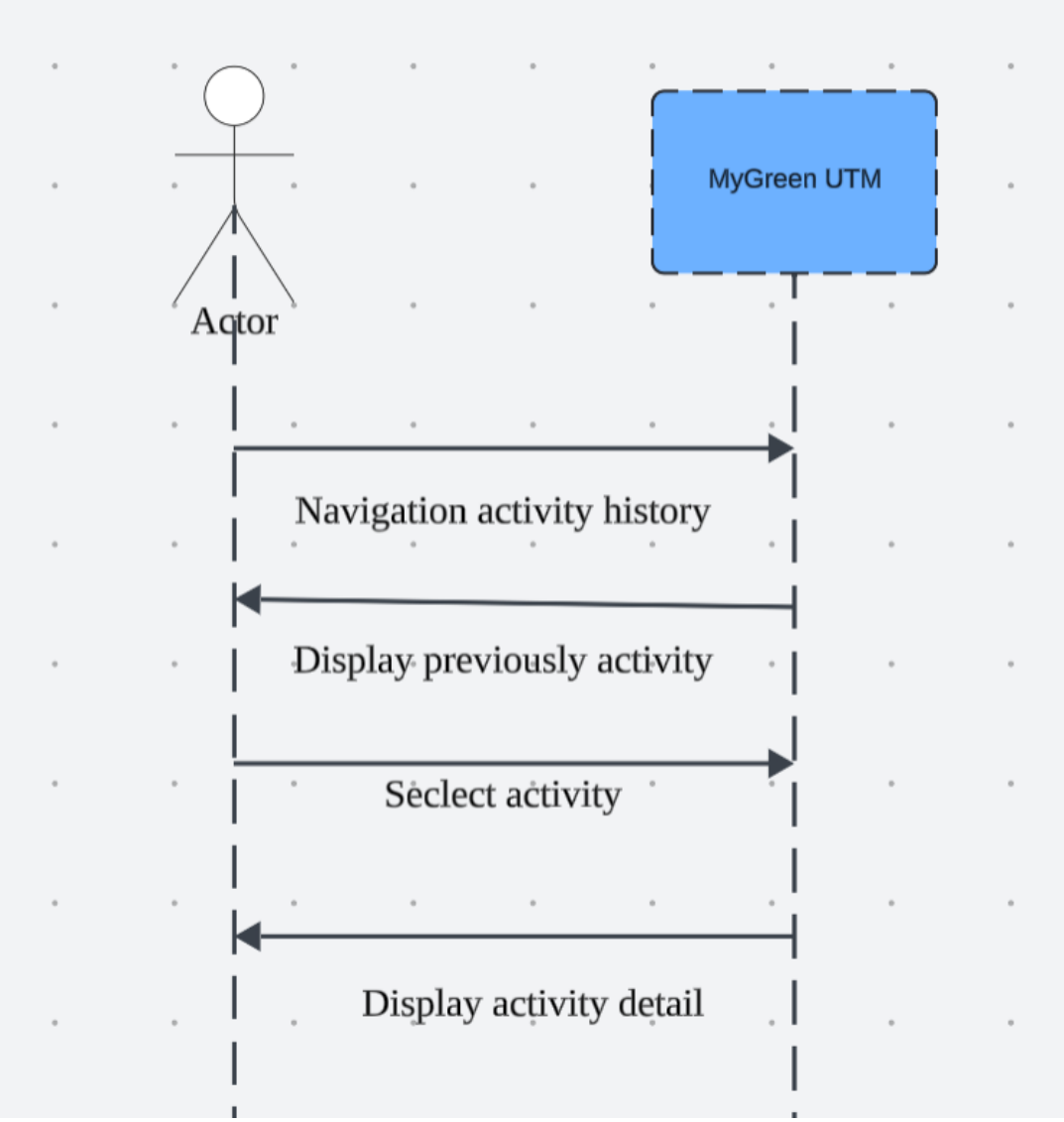



Actor
MyGreen UTM
display associate activity detail
Select additional detail
Submit activity detail
Alternative
Valid
Validate and save detail
Display success
Invalid
Display,prompts error and correct

3.3.9 **UC09: Use Case <View Scanned Activity Details>**

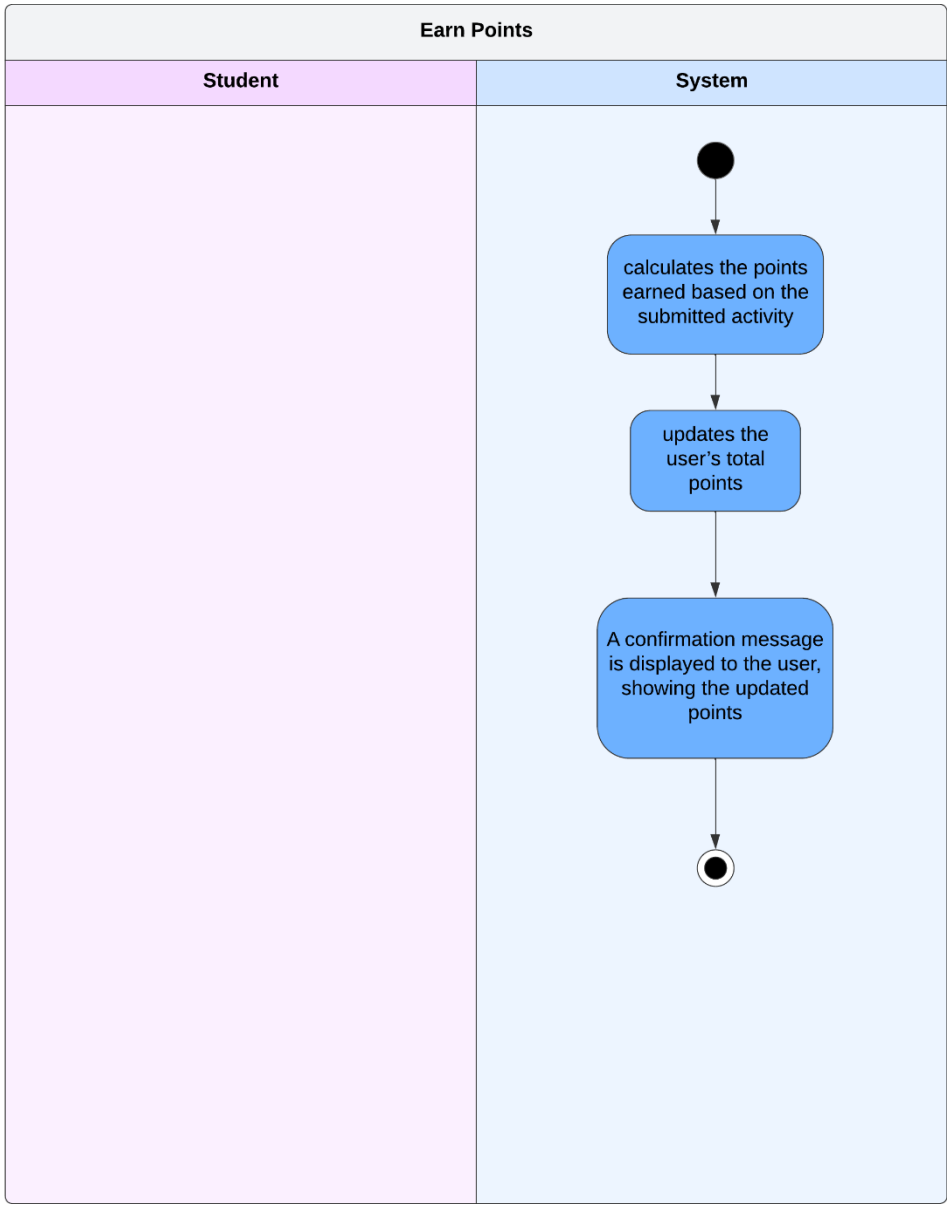
Use Case: View Scanned Activity Details
ID: UC09
Actors: UTM Students, UTM Staff
Includes: <UC08: Submit Activity Details>
Preconditions: The user must have successfully submitted activity details.
Flow of Events: 1. The user navigates to the “Activity History” section of the app. 2. The system displays a list of previously scanned and submitted activities. 3. The user selects a specific activity to view its details. 4. The system displays the details of the selected activity (e.g., activity type, location, timestamp).
Postconditions: The user successfully views the details of a previously scanned activity.
Alternative Flow:
Exception Flow:

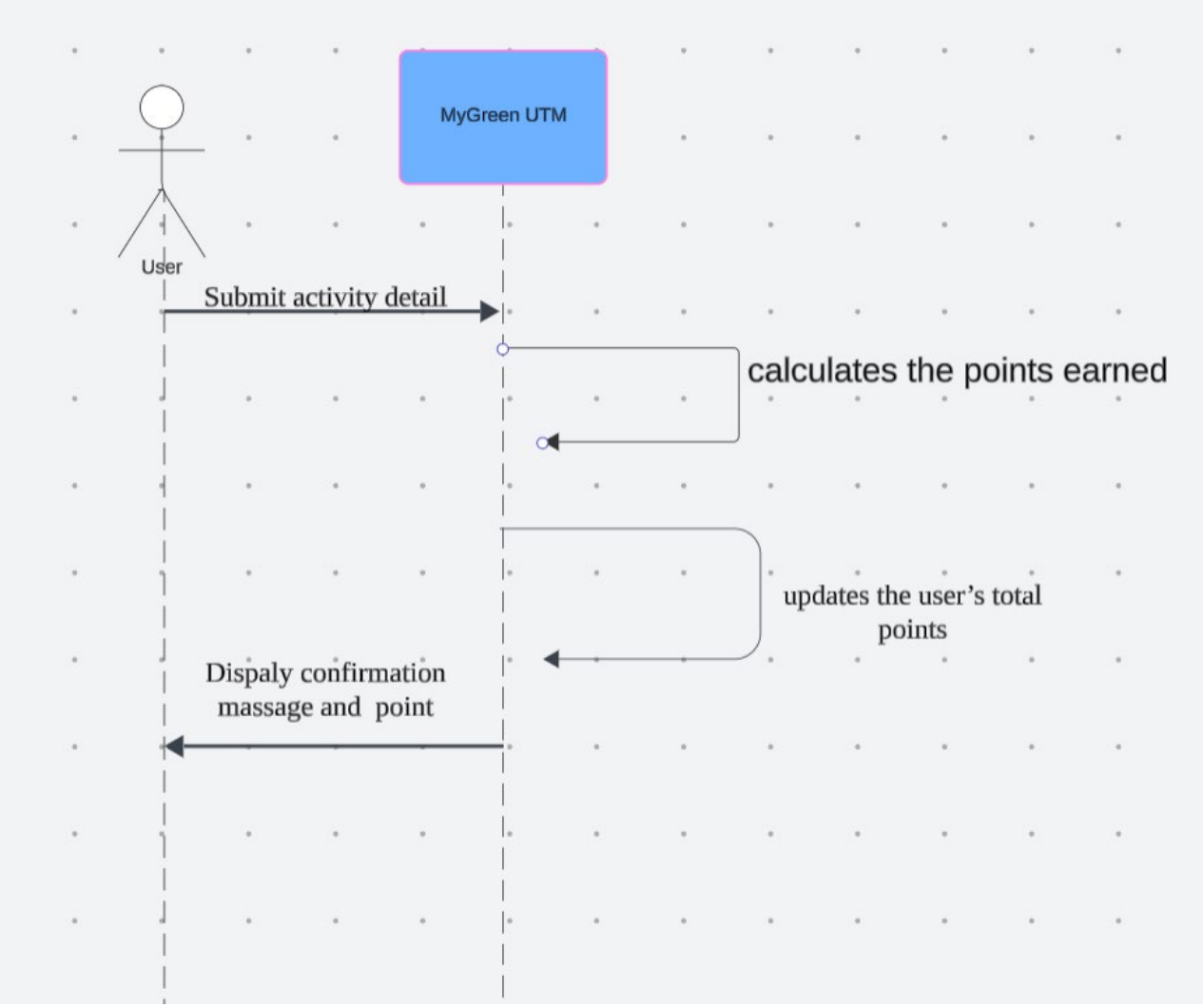

View Scanned Activity Details
Student
System
navigates to the “Activity History” section of the app
displays a list of previously scanned and submitted activities
selects a specific activity to view its details
displays the details of the selected activity


MyGreen UTM
Actor
Navigation activity history
Display previously activity
Seclect activity
Display activity detail

3.3.10 **UC10: Use Case <Earn Points>**

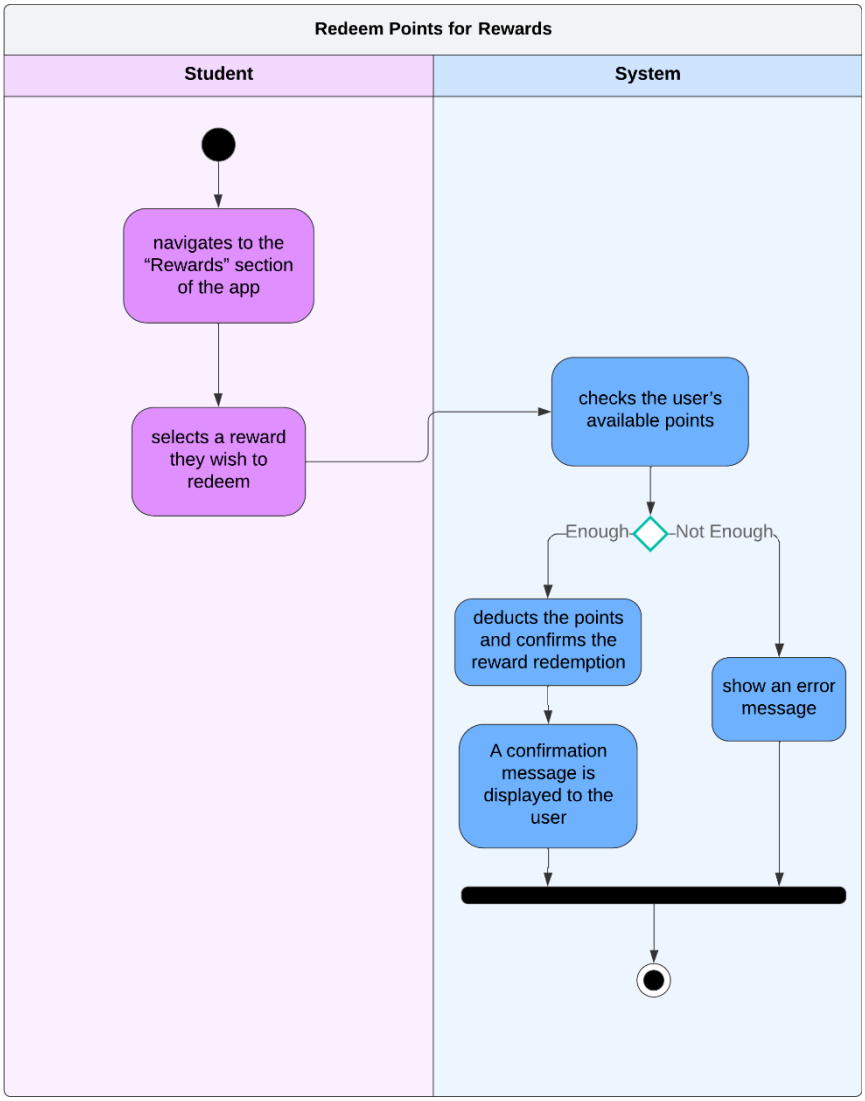
Use Case: Earn Points
ID: UC10
Actors: UTM Students
Includes: <UC08: Submit Activity Details>
Preconditions: The user must have successfully submitted activity details.
Flow of Events: 1. The system calculates the points earned based on the submitted activity. 2. The system updates the user's total points. 3. A confirmation message is displayed to the user, showing the updated points.
Postconditions: The user's total points are successfully updated in the system.
Alternative Flow:
Exception Flow:

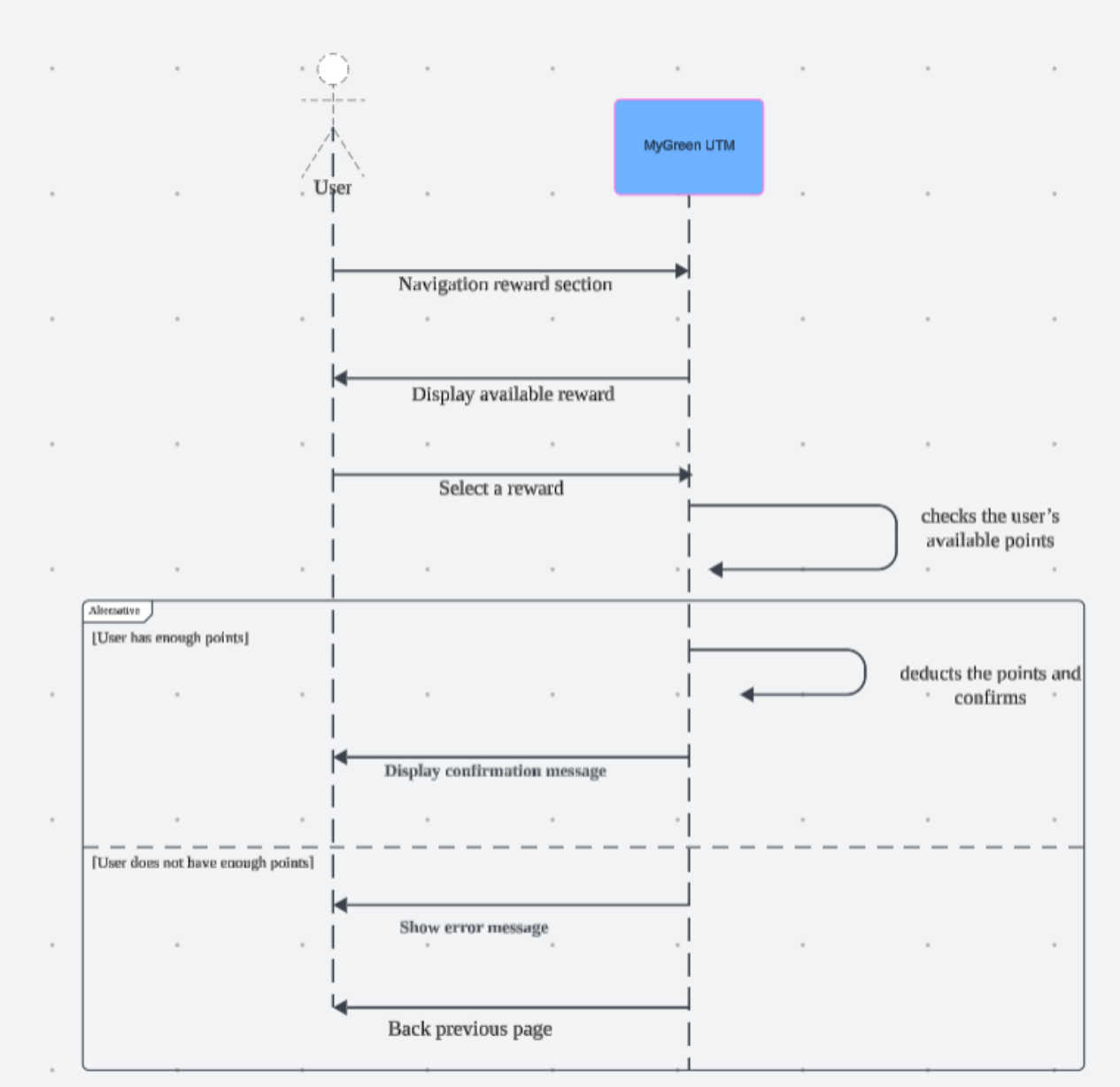

Earn Points
Student
System
calculates the points earned based on the submitted activity
updates the user's total points
A confirmation message is displayed to the user, showing the updated points


MyGreen UTM
User
Submit activity detail
calculates the points earned
updates the user's total points
Dispaly confirmation massage and point

3.3.11 **UC11: Use Case < Redeem Points for Rewards >**

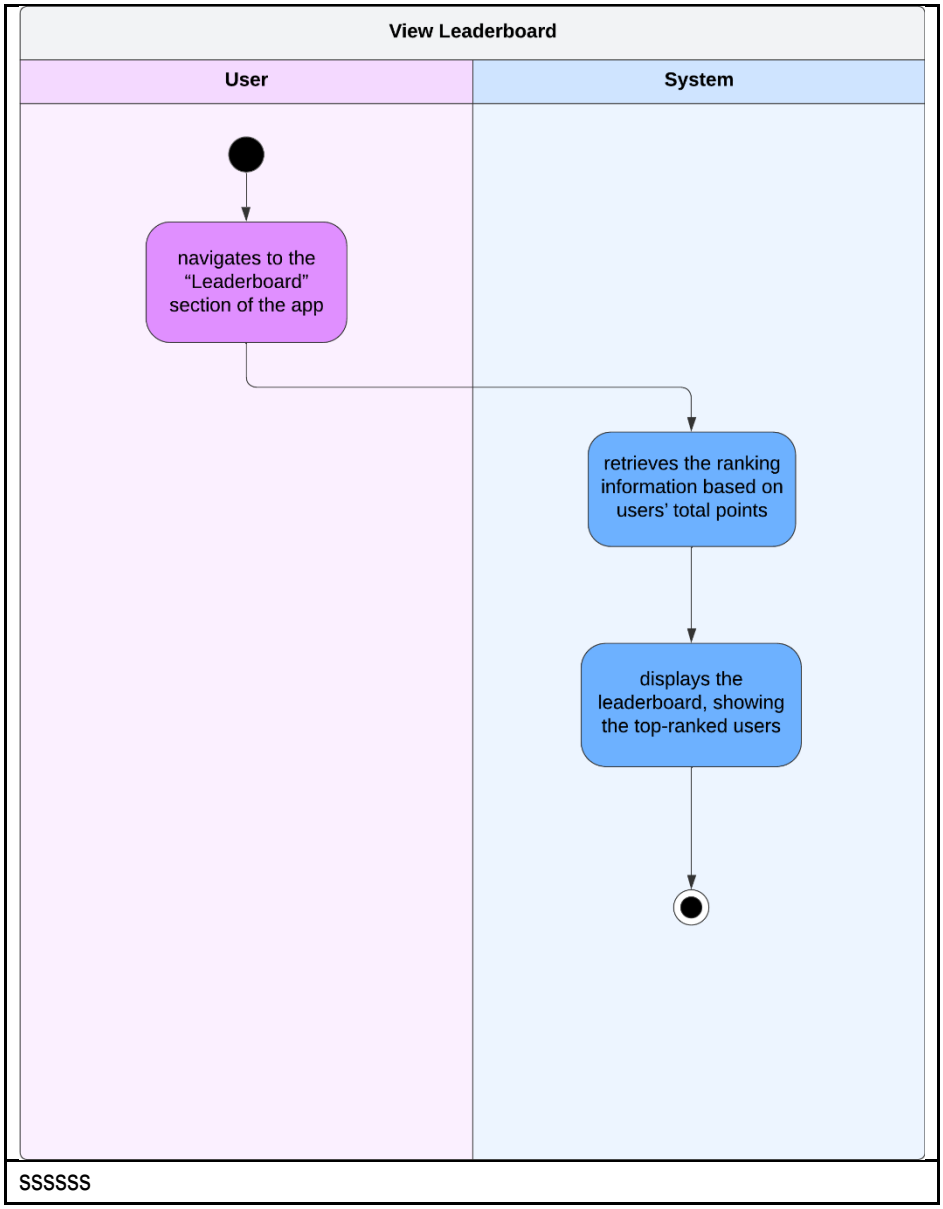
Use Case: Redeem Points for Rewards
ID: UC11
Actors: UTM Students
Includes: <UC10: Earn Points>
Preconditions: The user must have accumulated sufficient points.
Flow of Events: 1. The user navigates to the “Rewards” section of the app. 2. The user selects a reward they wish to redeem. 3. The system checks the user’s available points: 3.1 If the user has enough points, proceed to Step 4. 3.2 If the user does not have enough points, show an error message. (AF1) 4. The system deducts the points and confirms the reward redemption. 5. A confirmation message is displayed to the user.
Postconditions: The user successfully redeems the reward, and their points are updated.
Alternative Flow 1: If the user does not have enough points, the system displays an error message and goes back to the previous page.
Exception Flow:

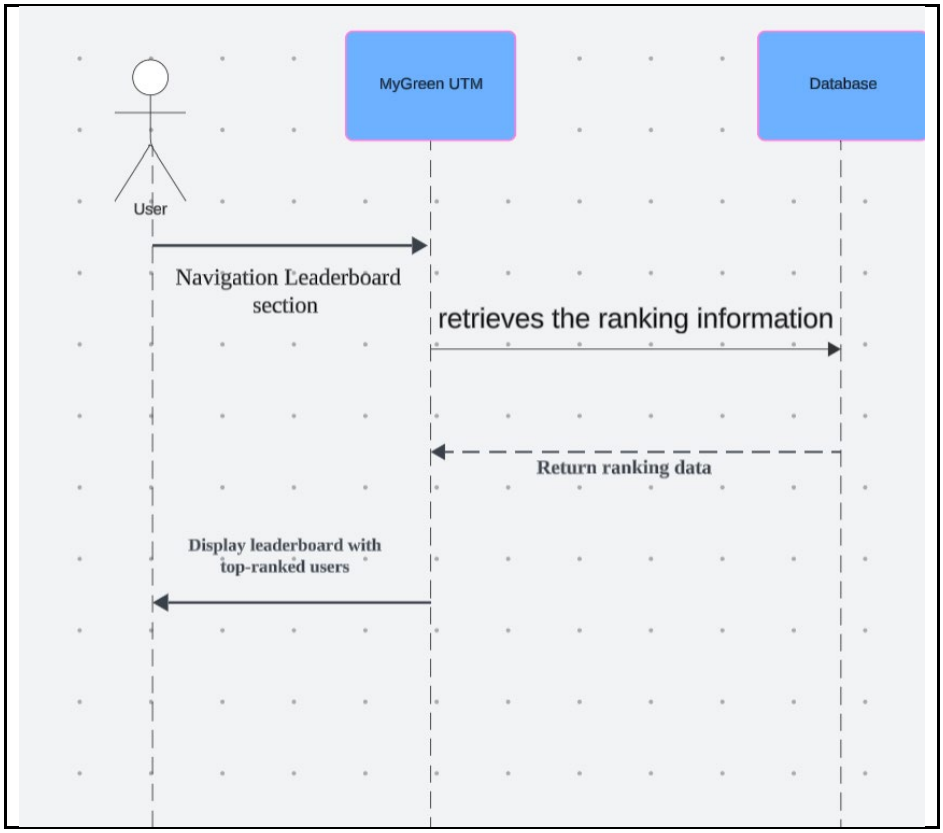

Redeem Points for Rewards
Student
System
navigates to the “Rewards” section of the app
selects a reward they wish to redeem
checks the user’s available points
Enough
Not Enough
deducts the points and confirms the reward redemption
show an error message
A confirmation message is displayed to the user


User
MyGreen UTM
Navigation reward section
Display available reward
Select a reward
checks the user's available points
Alternative
[User has enough points]
deducts the points and confirms
Display confirmation message
[User does not have enough points]
Show error message
Back previous page

3.3.12 UC12: Use Case <View Leaderboard>

Use Case: View Leaderboard
ID: UC12
Actors: UTM Students
Includes: <UC10: Earn Points>
Preconditions: The user must be logged in.
Flow of Events: 1. The user navigates to the “Leaderboard” section of the app. 2. The system retrieves the ranking information based on users’ total points. 3. The system displays the leaderboard, showing the top-ranked users.
Postconditions: The user successfully views the leaderboard and ranking details.
Alternative Flow:
Exception Flow:


View Leaderboard
User
System
navigates to the “Leaderboard” section of the app
retrieves the ranking information based on users' total points
displays the leaderboard, showing the top-ranked users
SSSSSS


MyGreen UTM
Database
User
Navigation Leaderboard section
retrieves the ranking information
Return ranking data
Display leaderboard with top-ranked users

3.4 Performance and Other Requirements

2.4.1 Performance Requirements

Response Time: The application must respond to user inputs within 1 second under normal conditions and within 2 seconds during peak usage.
Throughput: The system should handle up to 1,000 concurrent users during peak events like sustainability campaigns.
Capacity: Support for 20,000 registered users with the ability to scale for future increases.
Availability: Achieve 99.95% uptime to ensure users can access features reliably during green events.

2.4.2 Software System Attributes:

Usability: Intuitive UI for users with varying levels of technical expertise, including accessibility options like screen readers and adjustable font sizes.
Reliability: The app must log all user actions accurately and recover from any crashes within 5 minutes.
Maintainability: Modular code structure to allow for quick bug fixes and feature enhancements.
Portability: Compatible with both Android and iOS platforms, supporting their latest versions.
Compatibility: Integration with UTM's existing IT infrastructure and support for external systems such as QR code readers.

2.4.3 Security Requirements:

Data Protection: Use end-to-end encryption (AES-256) for sensitive information, such as user registration details and activity logs.
Authentication: Implement multi-factor authentication for staff accounts to secure sensitive data.

2.4.4 Environmental Impact

Low Power Usage: Optimize system algorithms to reduce power consumption on mobile devices, aligning with UTM's green initiatives.
Cloud Efficiency: Leverage energy-efficient cloud solutions to minimize carbon footprint.

3.5 Design Constraints

Environmental Constraints

The system must operate effectively in various environments, including areas with intermittent or low network connectivity. Offline caching should enable users to log activities even when internet access is unavailable.
The system should minimize energy consumption on mobile devices to align with UTM's green initiatives. Algorithms must be optimized for battery efficiency, especially during activities like QR code scanning.

All cloud operations should utilize energy-efficient and eco-friendly server solutions, contributing to reduced carbon emissions from backend operations.

Hardware Constraints

The application must run on mobile devices with at least:
- 2GB of RAM.
- Dual-core processors for standard processing requirements.

Compatibility should include Android 9.0/iOS 13 or later versions, ensuring seamless operation across both operating systems.
Devices with limited resources (e.g., older models) should still experience acceptable performance by utilizing adaptive features like reducing graphical intensity.

Security Constraints

All user data, including personal information and activity logs, must be encrypted using AES-256 encryption both during transmission and storage.
Multi-factor authentication (MFA) should be implemented for UTM staff to prevent unauthorized access to sensitive data, particularly during green event management.
The system should regularly conduct security audits to identify and resolve potential vulnerabilities, maintaining compliance with UTM's IT policies and industry best practices.

Legal and Regulatory Constraints

The application must comply with the Malaysian Personal Data Protection Act 2010, ensuring that user information is securely collected, stored, and processed.
Any data collection must be limited to what is necessary for the system's functionality, with explicit user consent for activities like leaderboard participation or public activity sharing.
Environmental data collected (e.g., carbon footprint or waste reduction statistics) must adhere to Malaysian environmental reporting standards for sustainability initiatives.

Performance Constraints

The system should:- Support up to 1,000 active users logging activities simultaneously without performance degradation, especially during high-demand periods like green events.
- Maintain a response time of under 3 seconds for all major functions, including event registration and activity submission, even under peak loads.
- Scale efficiently to handle a 20% increase in concurrent users during special campaigns, leveraging cloud infrastructure for dynamic resource allocation.

Ensure smooth operation for real-time functionalities like QR code scanning, with error recovery mechanisms in place to handle failed scans.

Organizational Standards

The design and implementation must follow IEEE 830-1998 standards for Software Requirements Specifications to ensure clarity and consistency.
Documentation and system architecture must be reviewed and approved by UTM's designated project governance body to ensure alignment with institutional goals and standards.
All system components should integrate seamlessly with UTM's existing IT infrastructure, including student and staff databases.

4 System Architectural Design

4.1 Architecture Style and Rationale Chosen Style: Layered Architecture

The Layered Architecture divides the system into multiple layers. Each layer has its own specific function and is connected to the other layers in an organized way. In this case, the system is split into three subsystems: Green Activity, Accounting, and Achievement.

Reasons:

Easy to Understand: Each part of the system does one main job. For example, Green Activity manages activities, Accounting handles points and accounts, and Achievement tracks rewards and leaderboards. This makes it easier to see what each part is responsible for.
Flexible for Changes: If we need to change something in one part (like adding a new rule for earning points), it won't break the other parts.
Easier to Fix Bugs: If there's a problem in one layer, we can focus on fixing that layer without worrying about the others.
Reusing Features: Some parts of the system can be reused in other places. For example, the points system could also be used in other university apps.

4.2 Component Model

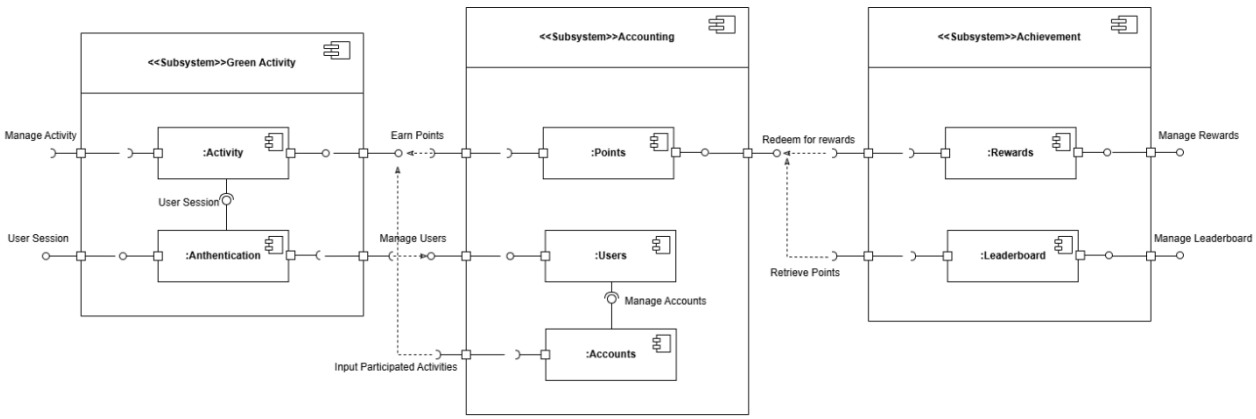


Figure 3.1: Component Diagram of <MyGreen UTM of the System>

4.Detailed Description of Components

4.3 Complete Package Diagram

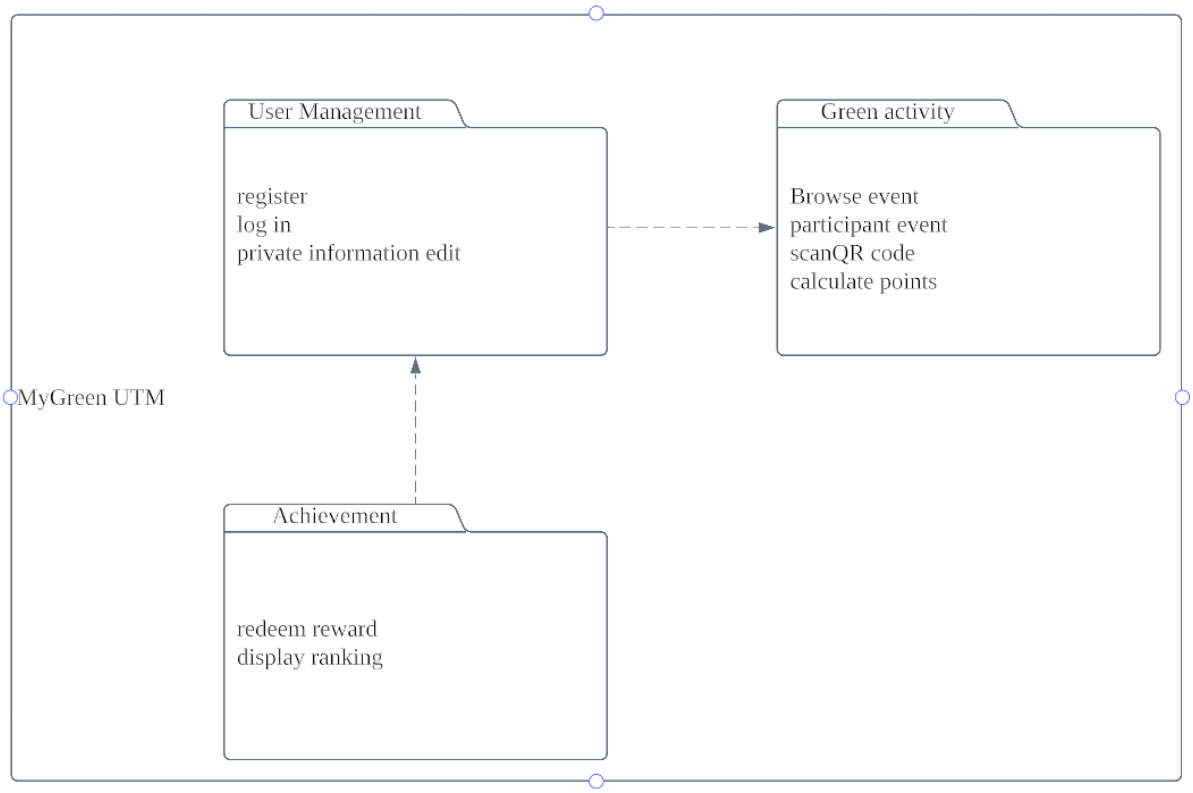


Figure 4.1: Package Diagram for <MyGreen UTM of the System>

4.4 Detailed Description

User Module:
Covers user-related functionalities such as registration, login, and profile management.

Green Activity Module:
includes activity-related functionalities like viewing activities, scanning QR codes, and submitting details.

Achievement Module:
Focuses on achievements, points, and leaderboards

4.4.1 .P001: <User Management > Subsystem

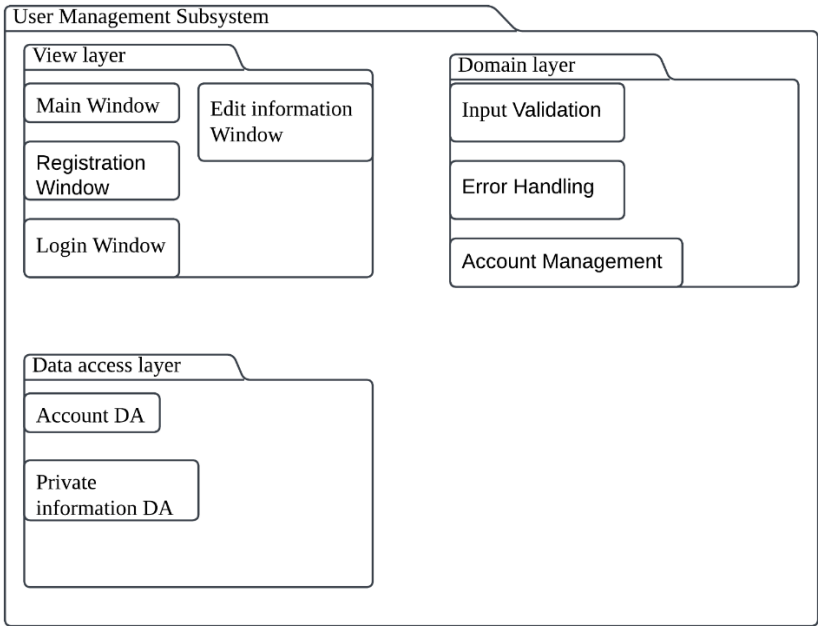
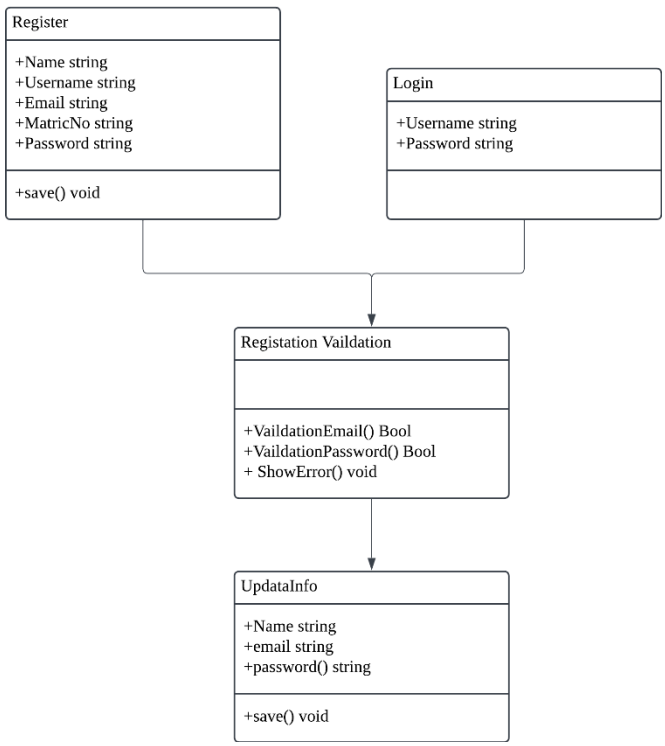


Figure 4.2: Package Diagram for < User Management > Subsystem

Class Diagram

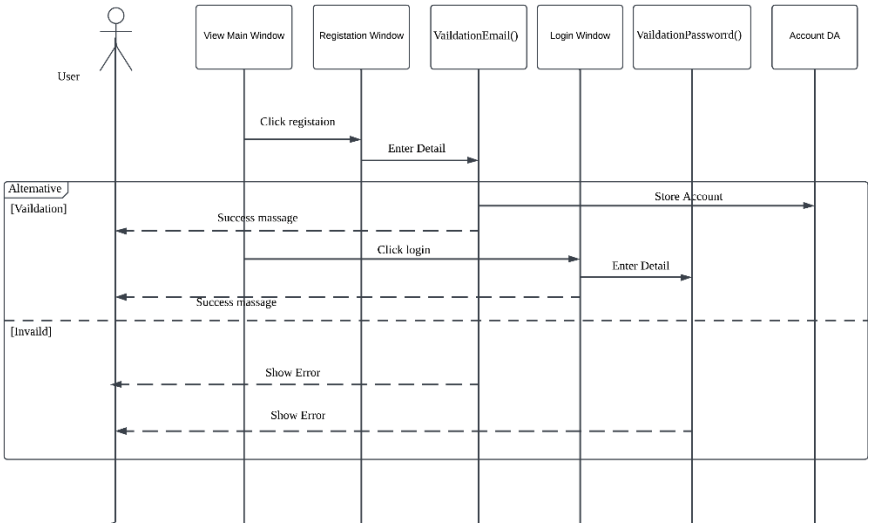


Entity Name	User Management
Method Name	Registration, Login
Input	Name, Username, Password, Email, MatricNo
Output	Registration/Login Success ,Failure Message
Algorithm	1. Start 2. Name, Username, Password, Email, MatricNo 3. Call VaildationPassword(),if password invalid, show error and try again. 4. Validate Email using validationEmail(). 5. If invalid, show error and end. 6. Display success message. 7. End

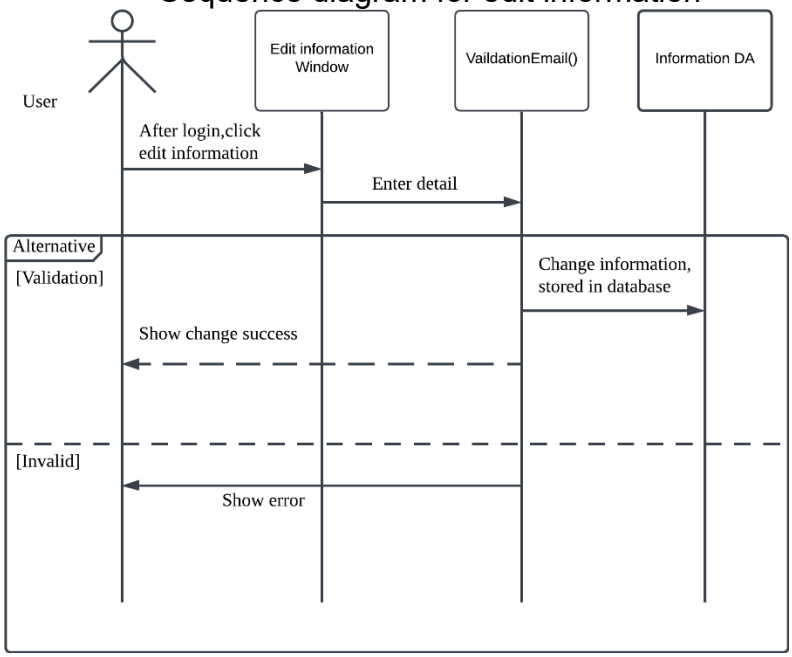
Entity Name	User Management
Method Name	Edit information
Input	Name, Password, Email, MatricNo
Output	Success ,Failure Message
Algorithm	1. Start 2. Name, Password, Email, MatricNo 3. Validate Email using validationEmail(). 4. If invalid, show error and end. 5. Display success message. 6. End

4.2.3.2 Sequence Diagram

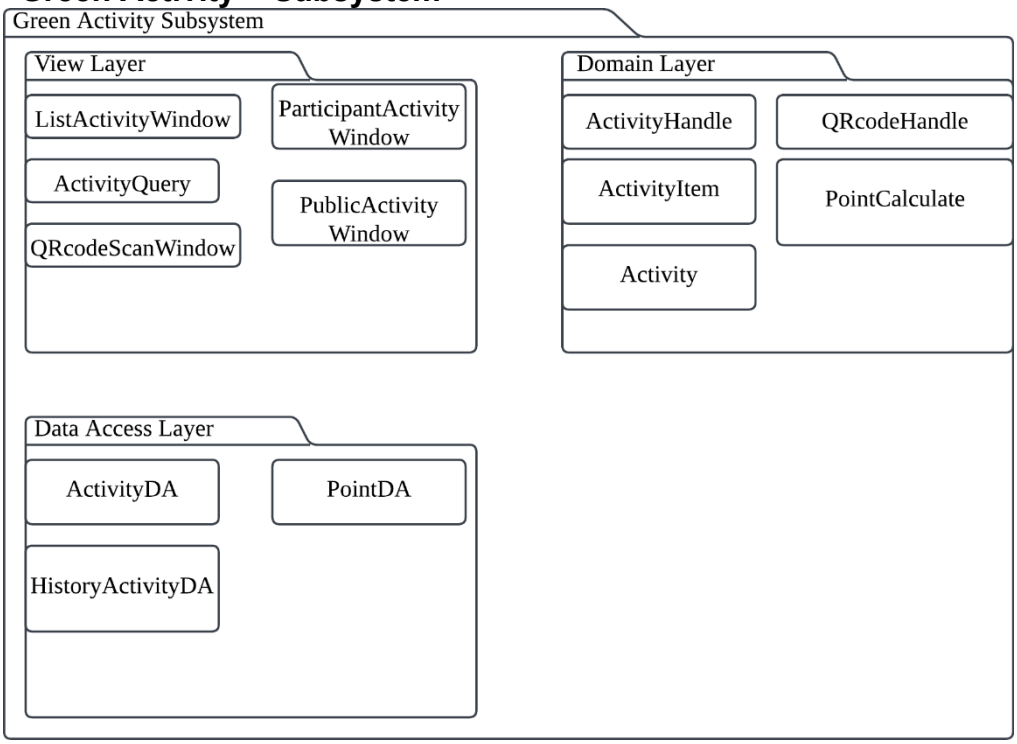
SD001: Sequence diagram for Log in and Registration



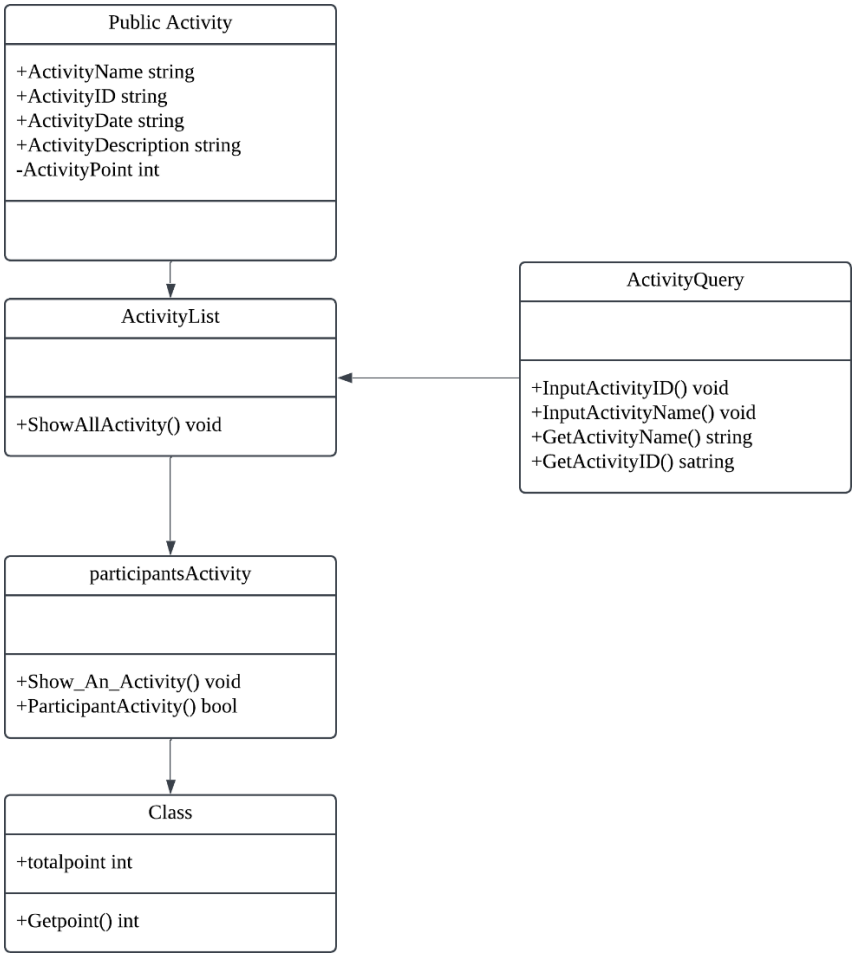
Sequence diagram for edit information



P002: < Green Activity > Subsystem



< Green Activity > Class Diagram

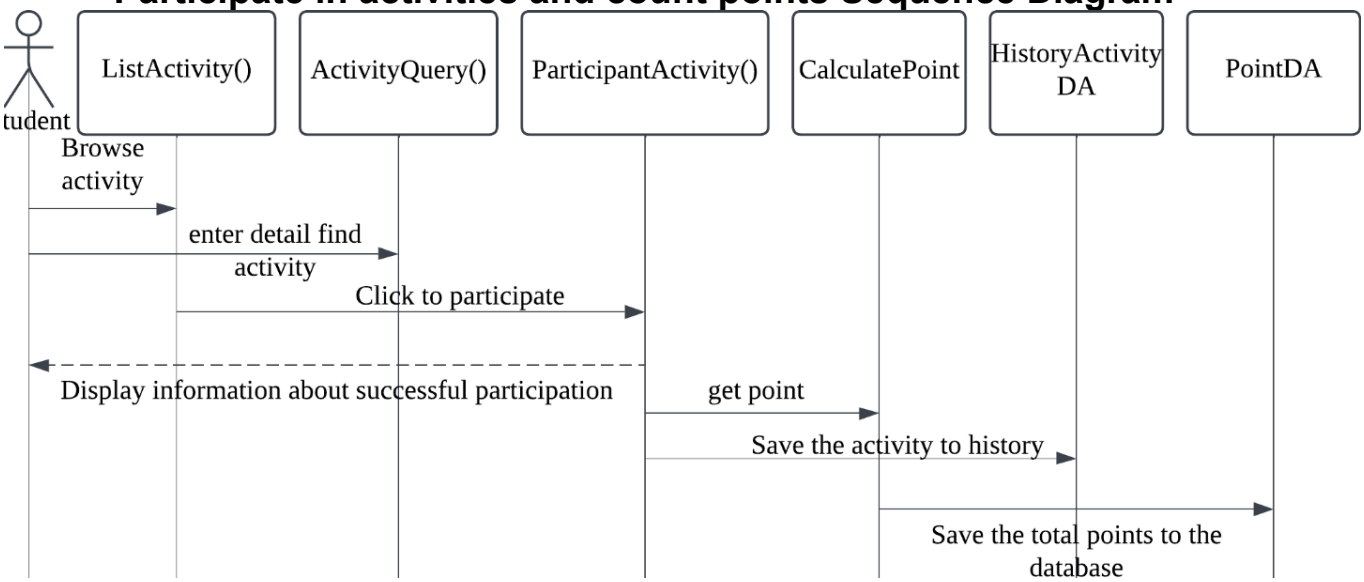


Entity Name	Green Activity
Method Name	Participate in activities and count points
Input	Choose whether to participate in the activity or by scanning QRcode
Output	Activity participation success, this activity point
Algorithm	1. Start 2. Browse directly to the activity list by calling ShowAllActivity(). 3. Input activity of name or id to query activity. 4. Whether to participate in activities by calling ParticipantActivity(). 5. Calculate the points for this activity and add them to the points you currently have. 6. End

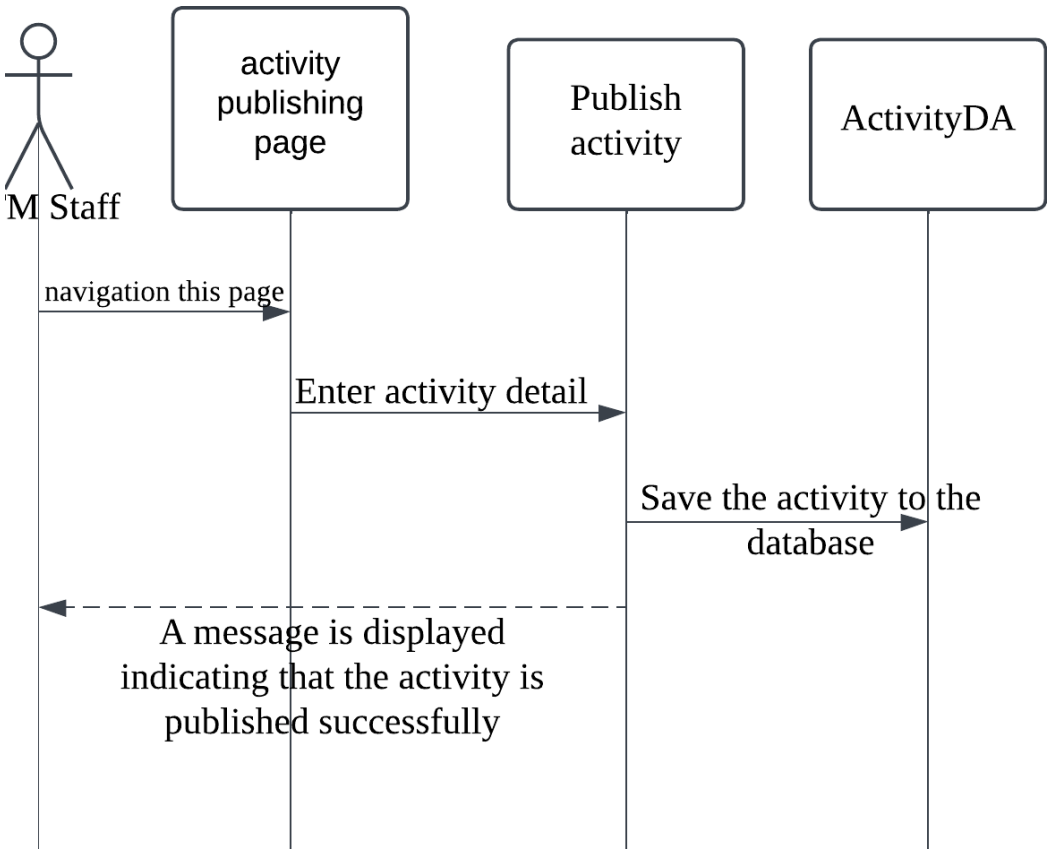
Entity Name	Green Activity
Method Name	UTM staff public activity
Input	Activity name, date, description, point
Output	Successful release

Algorithm	1. Start 2. The activity publishing page is displayed 3. Input activity detail 4. A message is displayed indicating that the activity is published successfully 5. End

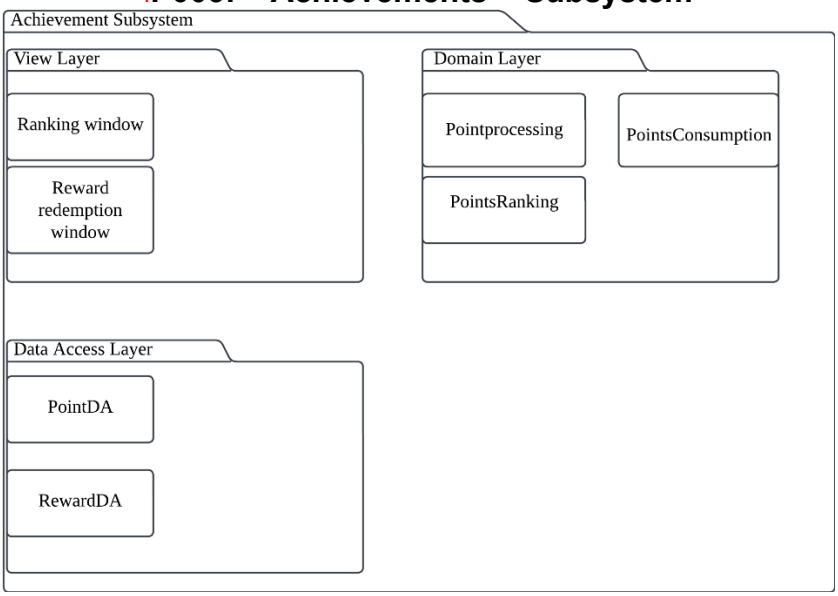
Participate in activities and count points Sequence Diagram



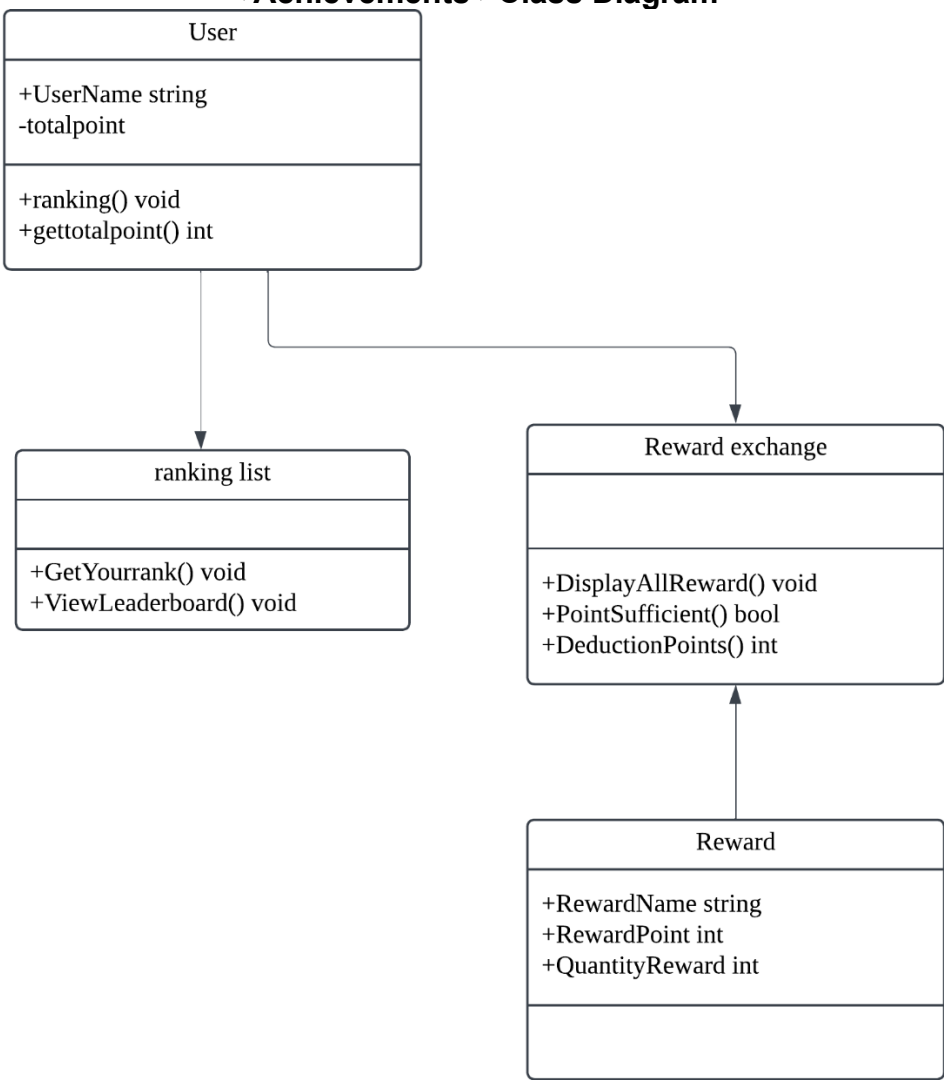
public activity Sequence Diagram


activity publishing page
Publish activity
ActivityDA
M Staff
navigation this page
Enter activity detail
Save the activity to the database
A message is displayed indicating that the activity is published successfully

P003: < Achievements > Subsystem



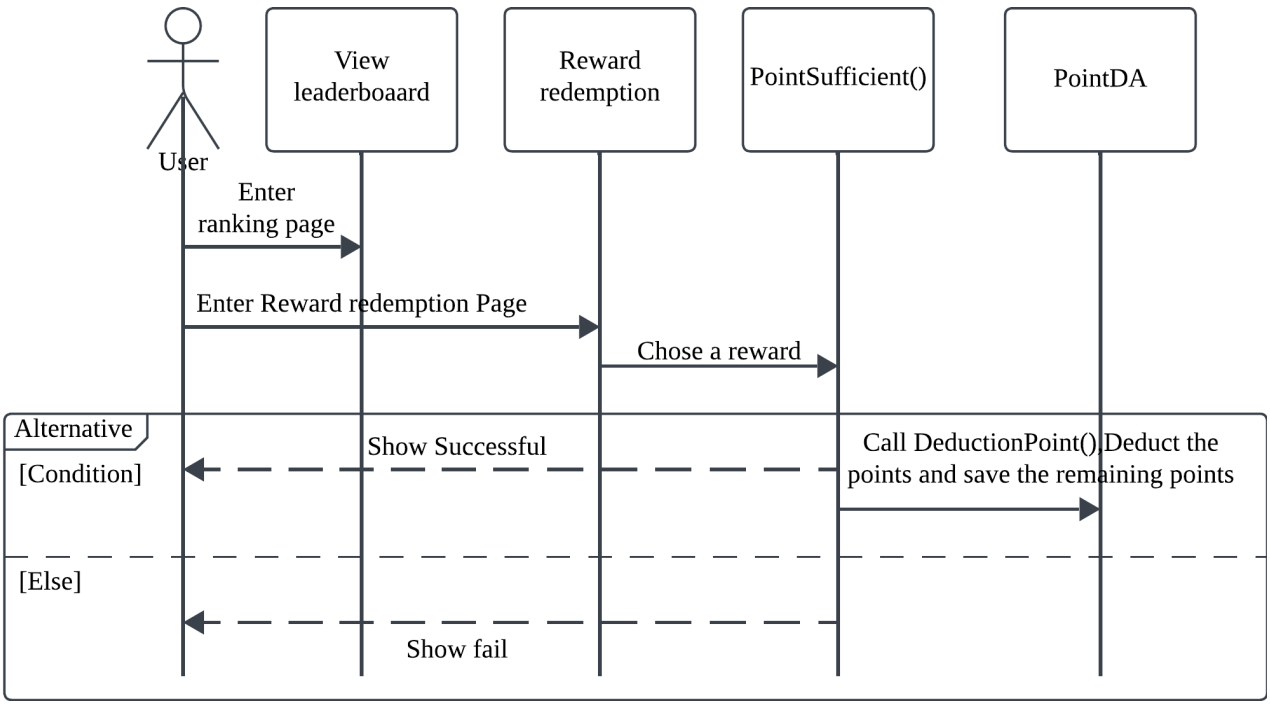
< Achievements >Class Diagram



Entity Name	Achievements
Method Name	View leaderboard
Input	
Output	Whole leaderboard
Algorithm	1. Start 2. Enter the ranking page 3. View the entire leaderboard to show your rank 4. End

Entity Name	Achievements
Method Name	Redemption reward
Input	Choose a reward
Output	The exchange succeeds or fails
Algorithm	1. Start 2. Enter the redemption reward page 3. Choose a reward 4. The system checks whether the point is sufficient 4.1 If satisfied return Successful 4.2 Eles return Point dissatisfy 5. End

Sequence Diagram



5.Data Design

1.1 **Data Description**

The data design of the "MyGreen UTM" system aims to support a user-friendly interface and efficient backend operations. The data domain is structured and managed within a relational database named GreenUTM_DB. The database design follows these principles:

1. **Entity Conceptualization**: All system functionalities are abstracted into independent data entities, such as users, activities, and participation records, to align with the business logic.
2. **Data Normalization**: The database design adheres to the Third Normal Form (3NF) to eliminate redundancy and ensure data consistency.
3. **Scalability**: Flexible table structures are designed to accommodate future functional extensions, such as reward redemption and leaderboard rankings.
4. **Data Processing Logic**: All data operations (insertion, updating, deletion) follow strict transaction management principles to ensure system reliability.

The major data or systems entities are stored into a relational database named as…, processed and organized into *n* entities as listed in Table 5.1.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1	User	Stores user information, including detailed data about registered students and staff.
2	Activity	Describes detailed information about green activities, such as event theme, time, location, and type.
3	Participation	Records users' participation history, including activity IDs, user IDs, timestamps, and points earned.
4	Reward	Manages reward information, including reward name, points required for redemption, and descriptions.
5	Leaderboard	Displays users' accumulated points and rankings to incentivize participation in green activities.

These entities not only meet the core functional requirements of the system but also reflect the key aspects of data management in a green campus initiative.

1.2 **Data Dictionary**

This section provides a detailed description of each entity's structure, attributes, data types, and constraints to ensure the integrity and reasonableness of the system's data model.

1.2.1 Entity: < User >

Background:

The User entity is central to the system, storing all essential user data. Both students and staff are authenticated and authorized through this table. Many system features (e.g., activity registration, points tracking) rely on user information.

Purpose:

- Used for identity authentication (login functionality).
- Stores user roles to assign appropriate permissions (e.g., activity creation is limited to staff).
- Tracks user behavior and participation records.

Table Definition:

Attribute Name	Type	Constraints	Description
UserID	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each user.
Name	VARCHAR	NOT NULL, MAX LENGTH 100	Full name of the user, displayed for personalization.
Email	VARCHAR	UNIQUE, NOT NULL, MAX LENGTH 100	User email, used for login and communication.
Password	VARCHAR	ENCRYPTED, NOT NULL	Encrypted password to ensure security.
Role	ENUM	VALUES ('Student', 'Staff'), NOT NULL	Role type (student or staff), determining system permissions.
Created At	DATETIME	DEFAULT CURRENT_TIMESTAMP	Timestamp of account creation, for logging and statistics.
Updated At	DATETIM	DEFAULT CURRENT_TIMESTAMP ON	Timestamp of the most recent update to the user record.

	E	UPDATE	

1.2.2 Entity: < Activity >

Background:

The Activity entity serves as the primary storage unit for green activity-related data. It includes details about each event, such as theme, time, and location. The classification of activity types facilitates user filtering and statistical analysis.

Purpose:

- Supports activity registration functionality.
- Enables the creation and management of green activities.
- Provides foundational data for points and participation tracking.

Table Definition:

Attribute Name	Type	Constraints	Description
ActivityID	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each activity.
Name	VARCHAR	NOT NULL, MAX LENGTH 200	Activity name describing the theme.
Date	DATE	NOT NULL	Date of the activity.
Location	VARCHAR	NOT NULL, MAX LENGTH 200	Location of the activity.
Type	ENUM	VALUES ('Awareness', 'Recycling', 'Education'), NOT NULL	Type of activity for easier classification.
MaxParticipants	INT	DEFAULT 0	Maximum number of participants allowed.
CreatedAt	DATETIME	DEFAULT CURRENT_TIMESTAMP	Timestamp when the activity was created.

1.2.3 Entity: < Participation >

Background:

The Participation entity records users' participation history, including activity details and points earned. It is essential for both the points system and leaderboard functionality.

Purpose:

- Tracks users' activity participation history.
- Provides data for leaderboard points accumulation.
- Supports reward redemption by tracking earned points.

Table Definition:

Attribute Name	Type	Constraints	Description
ParticipationID	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each participation record.
UserID	INT	FOREIGN KEY REFERENCES User(UserID)	Foreign key linking to the user ID.
ActivityID	INT	FOREIGN KEY REFERENCES Activity(ActivityID)	Foreign key linking to the activity ID.
Timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP	Timestamp of user participation.
PointsEarned	INT	NOT NULL	Points earned by the user for the activity.

1.2.4 Entity: < Reward >

Background:

The Reward entity stores all rewards offered by the system to encourage users' continuous participation in green activities. Rewards may include virtual badges or physical items.

Purpose:

- Provides a list of rewards redeemable by users.
- Specifies the points threshold required for each reward.

Table Definition:

Attribute Name|	Type|	Constraints|	Description
RewardID|	INT|	PRIMARY KEY AUTO_INCREMENT|	Unique identifier for each reward.|
Name|	VARCHAR|	NOT NULL MAX LENGTH 200|	Name of the reward.|
Description|	TEXT|	NULLABLE|	Detailed description of the reward.|

PointsRequired|	INT|	NOT NULL|	Points required to redeem the reward.|

1.2.5 Entity: < Leaderboard >

Background:

The Leaderboard entity provides data for ranking users based on accumulated points. This table dynamically generates rankings to incentivize participation.

Purpose:

- Encourages participation through competitive rankings.
- Supplies data for leaderboard displays.

Table Definition:

Attribute Name	Type	Constraints	Description
LeaderboardID	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for leaderboard records.
UserID	INT	FOREIGN KEY REFERENCES User(UserID)	Foreign key linking to the user ID.
TotalPoints	INT	NOT NULL	Total points accumulated by the user.
Rank	INT	GENERATED BASED ON TotalPoints DESC	User ranking dynamically generated based on total points.

1.2.6Relationships Between Entities

The relationships between entities are implemented using foreign keys:

1. **User and Participation**: A user can participate in multiple activities.
2. **Activity and Participation**: An activity can have multiple participation records.
3. **User and Leaderboard**: A user's points are aggregated for leaderboard rankings.

6.User Interface Design

1.3 **Overview of User Interface**
The interface includes…

The **MyGreen UTM** system features a user-friendly interface designed for simplicity, functionality, and accessibility:

Login Screen: Users log in with their UTM email and password. Options for password recovery and account registration are available. Feedback is provided for invalid credentials.

Activity List: Displays upcoming activities with details (title, date, location) and filtering options. Users can view and register for events easily.

Leaderboard: Shows top-ranked users by points, encouraging participation. Rankings update dynamically based on activity data.

Rewards Section: Lists rewards with required points for redemption. Users are notified of successful redemptions or insufficient points.

Navigation: A bottom navigation bar provides quick access to key sections (Home, Activities, Leaderboard, Profile). The design is responsive for mobile devices.

Feedback and Accessibility: Real-time feedback ensures clarity, while accessible design features (e.g., large buttons, high contrast) support all users, including those with disabilities.

1.4 **Screen Images**

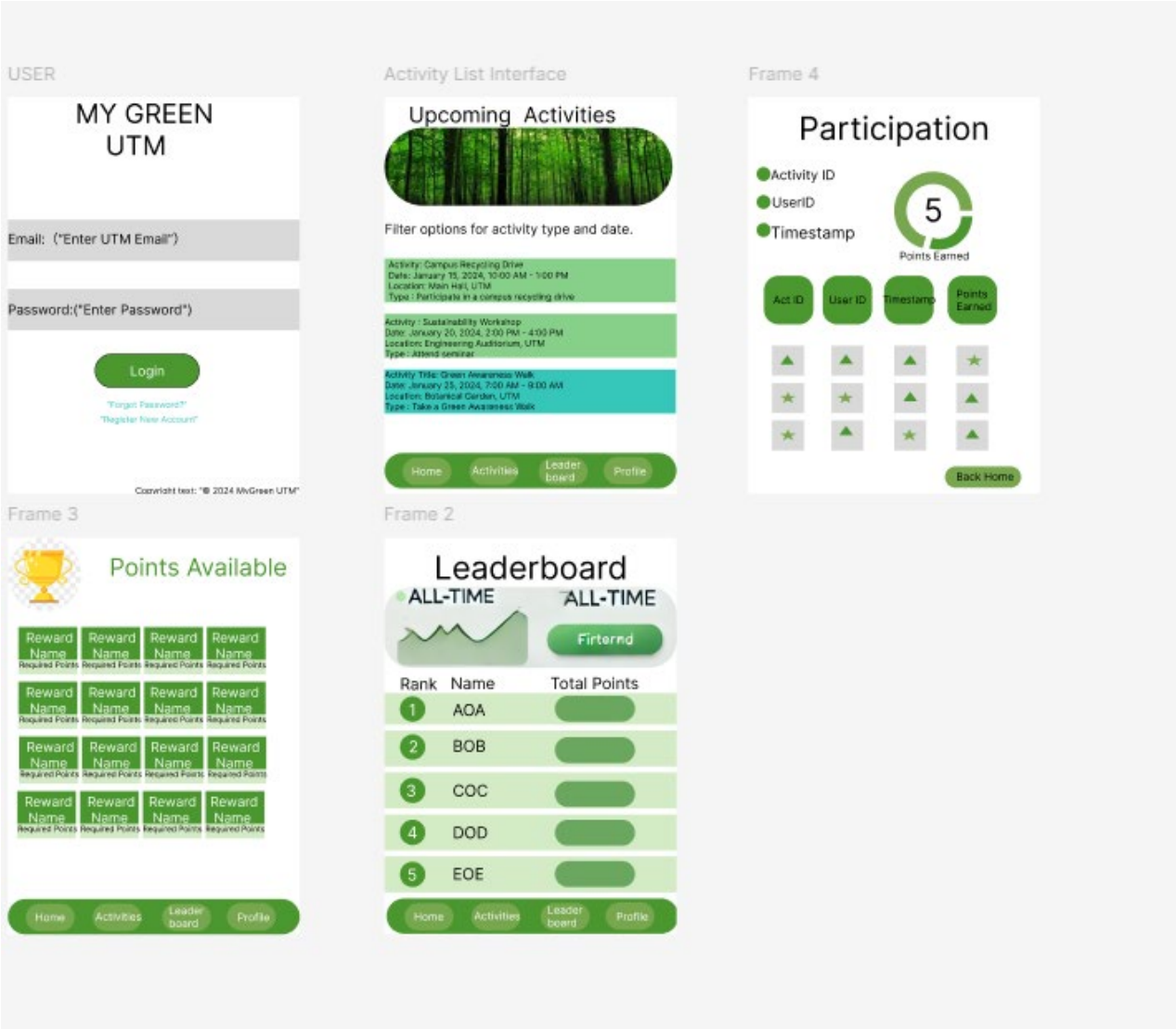


Figure 6.1:

Interface for <State the Feature/Function>