SC2006
Lab Assignment 1

System Name: Report Quest

Target users of application: Singapore community (based on residential areas)

Project Mission Statement:

This project will create an application that enables residents to easily report neighborhood issues, ensuring local authorities can promptly address them. By facilitating efficient communication between residents and authorities, the application will enhance community involvement and improve the overall quality of life. The outcome will be a well-maintained, safer, and more connected neighborhood.

- Problem: A platform that eases the communication between the authorities and the residents, reduces the time delay between reporting problems in the neighborhood and addressing the issue, provides a common platform for the community to voice their concerns.
- Stakeholders: community residents, community leaders, contractors
- Outcomes and benefits: problems can be addressed promptly and reward residents

I) Functional Requirements

1. **The system shall allow user registration and authentication.**

    1.1 The system shall allow users to create a new user's account.

        1.1.1 The system shall allow users to register with a unique username.

        1.1.2 The system shall ensure that the username is unique by checking it against existing usernames in the database.

        1.1.3 The system shall allow users to register with a contact number in a valid format.

        1.1.4 The system shall allow users to register with a unique email address.

        1.1.5 The system shall validate the email address format during registration.

        1.1.6 The system shall ensure the email address is unique by checking it against=existing emails in the database.

        1.1.7 The system shall allow users to set a password during registration.

        1.1.8 The system shall enforce password strength criteria (e.g., minimum length, inclusion of numbers and special characters).

1.1.9 The system shall send a confirmation email to the user upon successful registration with a verification link.

1.2 Users must be able to log in and log out using their credentials.

1.2.1 The system shall allow users to log in using their email or username and passwords.

1.2.2 The system shall verify the entered credentials against the stored user data in the database.

1.2.3 The system shall grant access to the user if the credentials are correct.

1.2.4 The system shall notify the user of incorrect credentials if the login fails.

1.2.5 The system shall allow users to log out from their session.

1.2.6 The system shall invalidate the user session upon logout.

1.3 Users must be able to reset their password if forgotten or when they want to change a new password.

1.3.1 The system shall provide users with an option to reset their password if forgotten.

1.3.2 The system shall require users to enter their registered email address to initiate the password reset process.

1.3.3 The system shall send a password reset link to the user's registered email address.

1.3.4 The system shall allow users to set a new password after clicking the reset link.

1.3.5 The system shall enforce password strength criteria when setting a new password.

1.3.6 The system shall allow users to change their password from within their account settings.

1.3.7 The system shall require the current password to be entered before setting a new password.

1.3.8 The system shall send an email notification to the user upon a successful password change.

2. The system shall allow profile management

2.1 Users should be able to view their profile details, including points balance and reward history.

2.1.1 The system shall allow users to view their profile details, including their username and email address.

2.1.2 The system shall display the user's current points balance.

2.1.3 The system shall provide users with access to their reward history, including details of earned and redeemed rewards.

2.1.4 The system shall ensure that profile details are displayed in a secure manner, only accessible to the authenticated user.

2.2 Users must be able to update their profile information, such as email address, contact number and profile picture.

2.2.1 The system shall allow users to update their email address from the profile settings.

2.2.2 The system shall validate the new email address format during the update process.

2.2.3 The system shall check the uniqueness of the new email address against existing records in the database.

2.2.4 The system shall allow users to update their profile picture by uploading a new image file.

2.2.5 The system shall ensure that the updated profile information is saved securely in the database.

2.2.6 The system shall notify the user of successful profile updates via email or an on-screen notification.

3. The system should allow registered users to make reports.

3.1 Users must be able to upload photos from their device and provide a description.

3.1.1 The system must allow users to navigate to the "Report" feature from the main menu or dashboard.

3.1.2 The system must display a “Make new reports” option on the “Report” page.

3.1.3 The system must restrict access to the "Make new reports" feature to registered users only.

3.1.4 The system must provide an option for users to upload a photo from their device's gallery.

3.1.5 The system must provide an option for users to take a new photo using the device's camera.

3.1.6 The system must display an input box for the user to describe the issue/fault after selecting or taking a photo.

3.1.7 The system must display an input field for users to enter the location where the issue was observed.

3.1.8 The system must provide an option for users to input the date and time the issue was observed.

3.2 The system should validate the photo and details provided by the user before submission.

3.2.1 The system must validate the file format of the uploaded photo to ensure it is in an acceptable format (eg. PNG, JPEG).

3.2.2 The system must check that the photo does not exceed the maximum allowed size (eg. 5MB).

3.2.3 The system must check that the description does not exceed the maximum character limit.

3.2.3.1 The system must display an error message if the description exceeds the word/character limit.

3.2.4 The system must inform the user of the outcome of the validation checks.

3.2.4.1 The system must prompt the user to correct any issues if validation checks fail.

3.2.4.2 The system must allow the user to proceed to metadata capture or final submission if all validation checks pass.

3.3 Users should be able to add metadata (eg. location) to the photo if applicable.

3.3.1 The system must prompt the user to add or confirm metadata related to the uploaded photo, such as location.

3.3.2 The system must allow the user to skip metadata entry and jump to final submission if it is optional.

3.3.3 The system must provide input fields for the user to manually enter or make changes to metadata details, such as location or time.

3.3.4 The system must check if the entered details are complete and valid.

3.3.4.1 If the details are incomplete and invalid, the system must prompt the user to correct it.

3.3.5 The system must store the metadata so that it can be retrieved for processing.

4. **Points System should award points to users after submission of report.**

4.1 Points Awarding

- 4.1.1 The system must automatically calculate and award points based on predefined criteria once a report is successfully submitted.
- 4.1.2 The points calculation must take into account factors such as:
  - 4.1.2.1 The severity of the issue was determined by the OLLAMA LLM (1-10 severity scale).
  - 4.1.2.2 The relevance of the photo to the description (relevance scale 1-10).
  - 4.1.2.3 The urgency of the issue (urgency scale 1-10).
- 4.1.3 The system must adjust the awarded points based on moderation outcomes or manual reviews.
- 4.1.4 The system must not award points for submissions that fail moderation or violate submission rules.
- 4.1.5 The system should cap the maximum points that can be awarded for a single submission.

4.2 Points Tracking

- 4.2.1 The system must maintain a record of awarded points for each user and display the current points balance in the user dashboard.
- 4.2.2 The system must allow users to view their transaction history, detailing:
  - 4.2.2.1 The date and time of each submission.
  - 4.2.2.2 The points awarded for each submission.
  - 4.2.2.3 The status of each submission (e.g., pending review, approved, rejected).

- 4.2.3 The system must categorize transaction history entries into distinct sections.
    - 4.2.3.1 Points earned from reports.
    - 4.2.3.2 Points deducted from rejected reports.
    - 4.2.3.3 Points deducted from claimed rewards (if applicable).
- 4.2.4 The system must allow users to filter or sort the transaction history by date, points, or submission status.
- 4.2.5 Users must be notified of any point adjustments (e.g., deduction after moderation) with a clear explanation.

5. User shall redeem gifts from the system.

5.1 The user should be able to access the reward catalog.

5.1.1. The user can browse the reward catalog, the system must display available gift cards and vouchers to the user.

- 5.1.1.1 The system must display the name, description, value, and points required for redeeming for each gift card or voucher.

5.1.2. The system must allow the user to filter the reward catalog.

- 5.1.2.1 The user must be able to filter the catalog by category, value and availability.

5.2. The user shall redeem the reward.

5.2.1. The user can select a gift card or voucher for redemption, the system must verify the user's points balance.

- 5.2.1.1 The system must check if the user's points balance is equal to or greater than the points required for redemption.
- 5.2.1.2 If the user's points balance is insufficient, the system must display an error message.

5.2.2. If the gift card or voucher is successfully redeemed, the system must deduct the points from the user's balance.

- 5.2.2.1 The system must deduct the exact number of points required for the selected gift card or voucher.
- 5.2.2.2 The system must update the user's points balance after the deduction.

5.2.3. When a gift card or voucher is successfully redeemed, the system must provide the user with the redeemed item.

- 5.2.3.1 The system must deliver the redeemed gift card or voucher to the user via email.
- 5.2.3.2 The system must make the redeemed gift card or voucher available within the user's account.

5.3. The system should give confirmation.

5.3.1. If the redemption is successful, the system should confirm the transaction.

- 5.3.1.1 The system must display a confirmation message to the user.
- 5.3.1.2 The system must log the redemption transaction.
- 5.3.1.3 The system must update the user’s transaction history with the redemption details.

5.3.2. When a redemption is unsuccessful, the system must inform the user.

- 5.3.2.1 The system must display an error message indicating the reason for the failure (e.g., insufficient points, item out of stock).

6. Severity Ranking and Reporting

6.1 Severity Assessment

- 6.1.1 The system must assess and rank photos based on their severity or impact, considering categories such as:
    - 6.1.1.1 Safety hazards.
    - 6.1.1.2 Environmental issues.
    - 6.1.1.3 Public health risks.
- 6.1.2 The severity ranking must be determined using the OLLAMA LLM, which will rate the severity on a scale of 1-10.
- 6.1.3 The system must display the severity rating to the user upon submission.
- 6.1.4 The system must provide an explanation of the severity ranking, including key factors affecting the rating (e.g., high urgency due to proximity to a school zone).
- 6.1.5 If the severity is below a predefined threshold (e.g., severity rating below 3), the system must inform the user that their submission may not require immediate action.
- 6.1.6 Users must have an option to manually report an issue as "urgent" for consideration by moderators, overriding the automatic severity ranking.

6.2 Automated Reporting

- 6.2.1 The system must automatically inform relevant authorities or agencies based on the severity ranking.
- 6.2.2 The system must define a severity threshold (e.g., a rating of 7 or above) to trigger automated reporting.

- 6.2.3 The system must notify the user if their report has been escalated and sent to authorities.
- 6.2.4 The system must send an alert to the appropriate agency with relevant details, including:
    - 6.2.4.1 The location of the issue, which the user provides or confirms.
    - 6.2.4.2 The date and time of the report.
    - 6.2.4.3 The severity ranking and description of the issue provided by the user.
    - 6.2.4.4 The photo submitted by the user, ensuring it meets quality standards (e.g., clear image of the issue).
- 6.2.5 The system must allow the user to receive updates on the status of their submission once it has been forwarded to authorities.

6.3 Manual Review

- 6.3.1 Admins or moderators must be able to review the severity ranking and adjust it if necessary.
- 6.3.2 The system must provide admins with a detailed report of the submission, including:
    - 6.3.2.1 The original severity ranking provided by the OLLAMA LLM.
    - 6.3.2.2 The photo, description, and metadata provided by the user.
    - 6.3.2.3 The relevance score of the photo to the description, which helps admins in decision-making.
- 6.3.3 Admins must have the ability to override the automated severity ranking and update the report status (e.g., escalate to a higher priority or downgrade if needed).
- 6.3.4 The system must allow admins to leave comments or notes explaining any adjustments made to the severity ranking.
- 6.3.5 The system must log any changes made by admins or moderators, including:
    - 6.3.5.1 The reason for the adjustment (e.g., "misclassified as a low severity issue").
    - 6.3.5.2 The updated severity ranking and new classification.
    - 6.3.5.3 The admin or moderator responsible for the adjustment.
- 6.3.6 The system must notify users if their report’s severity ranking has been adjusted after manual review.
- 6.3.7 The system must allow users to appeal or dispute the ranking adjustment, providing further details or evidence if necessary.

7. **The admin and authority will have additional actions.**

7.1 Admin has separate actions which the system provides.

7.1.1. The admin will be able to access the admin dashboard.

7.1.1.1. When accessing the admin dashboard, the system must allow admins to manage user accounts.

- 7.1.1.1.1. The system must allow admins to view a list of all user accounts.
- 7.1.1.1.2. The system must allow admins to deactivate or activate user accounts.
- 7.1.1.1.3. The system must allow admins to reset passwords for user accounts.

7.1.1.2. When accessing the admin dashboard, the system must allow admins to monitor user submissions.

- 7.1.1.2.1. The system must display a list of all user submissions.
- 7.1.1.2.2. The system must allow admins to filter submissions by date, status, or severity.
- 7.1.1.2.3. The system must allow admins to view details of each submission.

7.1.1.3. When accessing the admin dashboard, the system must allow admins to oversee points and rewards.

- 7.1.1.3.1. The system must display a summary of points distribution across users.
- 7.1.1.3.2. The system must allow admins to adjust points for individual users.
- 7.1.1.3.3. The system must allow admins to view the history of points transactions for each user.

7.1.2. The admin must be able to manage the rewards.

7.1.2.1. When managing the reward catalog, the system must allow admins to add new gift cards and vouchers.

- 7.1.2.1.1. The system must allow admins to enter the name, description, and value of the new gift card or voucher.
- 7.1.2.1.2. The system must allow admins to specify the points required for redeeming the new gift card or voucher.
- 7.1.2.1.3. The system must save the new gift card or voucher in the reward catalog upon submission.

7.1.2.2. When managing the reward catalog, the system must allow admins to update existing gift cards and vouchers.

- 7.1.2.2.1 The system must allow admins to select an existing gift card or voucher for editing.
- 7.1.2.2.2 The system must allow admins to update the name, description, value, or points required for the selected item.
- 7.1.2.2.3 The system must save the updated details to the reward catalog upon submission.

7.1.2.3. When managing the reward catalog, the system must allow admins to remove gift cards and vouchers.

- 7.1.2.3.1. The system must allow admins to select an existing gift card or voucher for removal.
- 7.1.2.3.2. The system must confirm the removal action before permanently deleting the item from the reward catalog.
- 7.1.2.3.3. The system must update the reward catalog to reflect the removal.

7.2. Authorities should receive notifications from the system and reply.

7.2.1. When a severe or high-impact submission is made, the system must notify relevant authorities.

- 7.2.1.1. The system must classify submissions by severity and impact.
- 7.2.1.2. The system must send notifications to relevant authorities for submissions classified as severe or high-impact.
- 7.2.1.3. The system must include the submission details in the notification sent to authorities.

7.2.2. When authorities receive a notification, they must be able to update the progress of solving the issue.

- 7.2.2.1. The system must allow authorities to access the submission linked to the notification.
- 7.2.2.2. The system must allow authorities to update the status of the submission (e.g., in progress, resolved).
- 7.2.2.3. The system must record the progress updates and make them visible to relevant users and admins.

8. The system should allow for sending notifications.

8.1 Users should receive notifications for key events such as successful submissions, points earned, and reward redemptions. Notifications can be in the form of emails or in-app alerts.

8.1.1 The system must provide users with an option to enable or disable notifications.

8.1.2 The system must allow users to select their preferred notification method (eg. email or in-app alerts).

8.1.3 The system must default to in-app notifications if the user has not specified a preference.

8.1.4 The system must send notifications via the user's selected method.

8.1.5 The system must allow users to change their notification preferences at any time.

8.1.6 The system must trigger notifications when a user action results in a successful report submission.

8.1.7 The system must trigger notifications when a user earns points.

8.1.8 The system must trigger notifications when a user redeems rewards.

8.2 Authorities should receive alerts for high-severity issues or critical submissions.

8.2.1 The system must classify a submitted fault report as high-severity based on a predefined criterion.

8.2.2 The system must generate an alert message that includes details of the high-severity issue and its location.

8.2.3 The system must identify the appropriate authorities based on the location and type of issue.

8.2.4 The system must send the alert to the relevant authorities via email, SMS, or any other suitable communication channel.

8.2.5 The system must confirm that authorities have received and acknowledged the alert within the system.

8.2.6 The system must prioritize sending alerts for high-severity issues to ensure timely responses.

8.2.7 The system must send alerts immediately upon classifying an issue as high-severity.

9. Search and Filtering

9.1 - The System must store all user submission into a Database.

9.1.1 - A title of the submission should be self-generated in the format of “<Issue> at <Location>, <Time and Date>”.

9.1.2 - A user submission record in the Database must consist of submission title, images of the issue, the description that was provided, location of issue, time and date of report, rank of severity given and the response by authorities.

9.2 - System’s Report page should display the User’s active reports and past reports.

9.2.1 - The System must be able to display all reports of a User by sorting based on categories, severity or time/date of report made.

9.2.2 - The User should be able to sort reports based on the options in 9.2.1 and reports should be displayed accordingly.

9.2.2 - When opening the Report page on startup, the reports should be displayed in order based on date, from most recent to oldest by default.

9.3 - System’s Report page should include a search bar at the top.

9.4 - Users should be able to enter keywords in the search bar to search for their active/past reports.

9.5 - The System should query the database based on User ID, given keywords and retrieve all relevant reports.

9.5.1 - After retrieval completes, only relevant reports should then be displayed to the User.

10. User Reports

10.1 Reporting and Analytics must not be accessible to users (Access Restriction)

10.2 Reporting and Analytics must be accessible to Admins

10.3 System must allow admins to generate reports

10.3.1 System must be able to search for user through their email

10.3.2 System must be able to search for user through their username

10.3.3 System must allow admins to view reports on user’s submissions

10.3.3.1 User submissions must include timestamps

10.3.3.2 User submission must include submission incident history details

10.3.3.3 User submission must include user information

10.3.2 Systems must allow admins to view reports on user’s point transactions

10.3.2.1 Point transactions must include the total number of points user currently have

10.3.2.2 Point transactions must include the total number of points user have accumulated so far

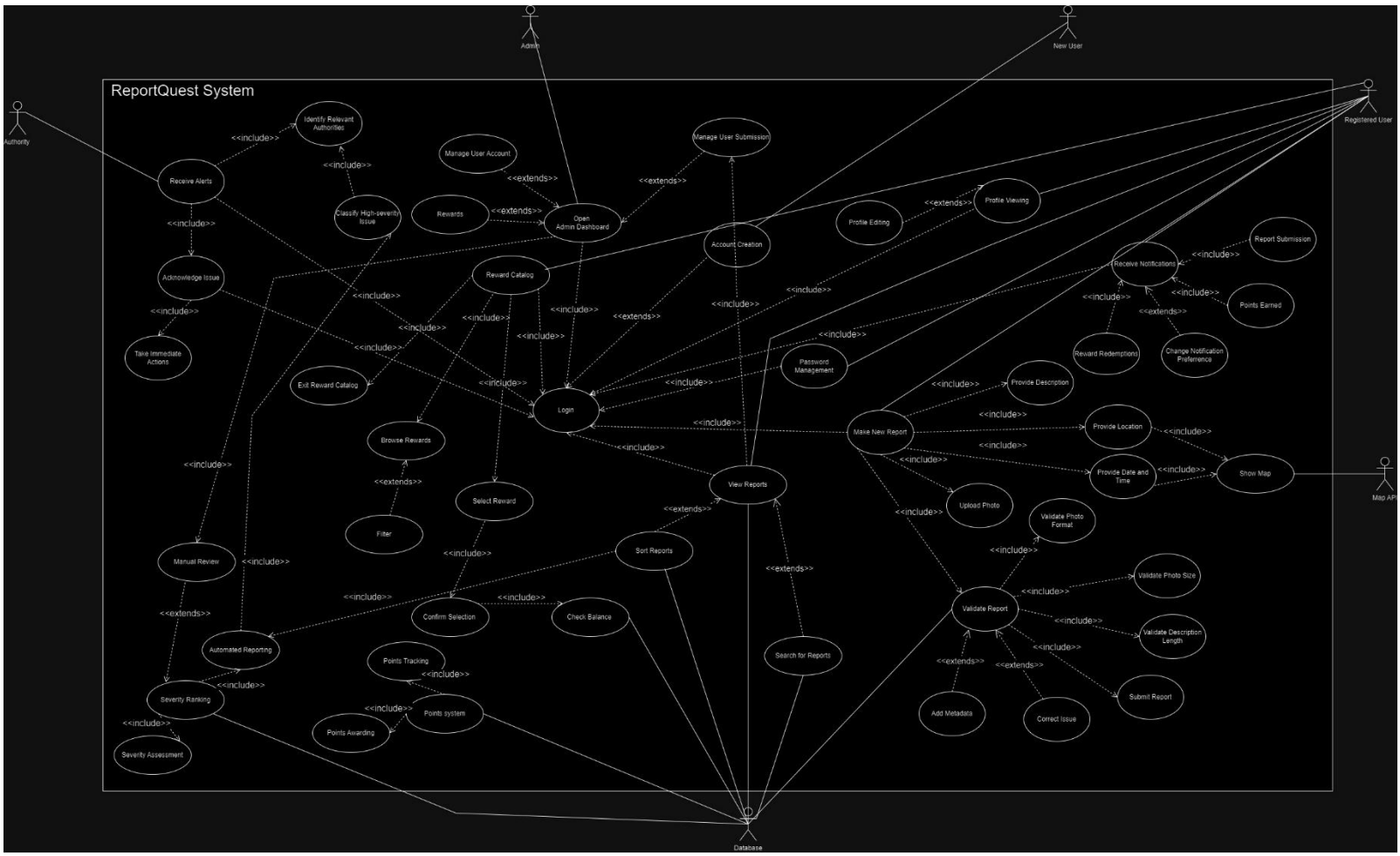
10.3.2.3 Point transactions must include detailed records of past reward redemptions

10.3.2.3.1 Reward redemptions must include the reward type

10.3.2.3.2 Reward redemptions must include the redemption dates

10.3.2.3.3 Reward redemptions must include the reward costs

II) USE CASE DIAGRAM:



III) USE CASE DESCRIPTION:

Use Case ID:	1.1		
Use Case Name:	Account creation		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	System, New user
Description:	The "Account Creation" use case allows a new user to register for an account on the platform by providing necessary personal details.
Preconditions:	1. The user has access to the website. 2. The user does not have an existing account in the system.
Postconditions:	1. A new user account is created and activated, allowing the user to log in and access the platform's services. 2. The user receives a welcome message and any additional instructions for using the platform.
Priority:	
Frequency of Use:	
Flow of Events:	1. The new user navigates to the "Sign Up" or "Create Account" page. 2. The user is prompted to enter the following details: ● Unique username ● Password ● Confirm password ● Email address 3. The system validates the input: ● Ensures the username is unique and meets any specified criteria (e.g., length, allowed characters). ● Validates the email format and checks if the email is already registered. ● Checks the password strength and confirms it matches the confirmation field. 4. Upon successful validation, the system creates a new user account. 5. The system sends a verification email to the provided contact information. 6. The user confirms their account by following the verification instructions. 7. The system activates the account, allowing the user to log in and access their profile.

Alternative Flows:	1. Invalid Input: If any of the user inputs are invalid (e.g., username taken, weak password), the system prompts the user to correct the errors. 2. Verification Failure: If the user does not verify their account within a specified time, the account remains inactive until verified.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	1.2		
Use Case Name:	Login		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	System, registered user
Description:	The "Login and Logout" use case enables registered users to securely access and exit their account on the platform.
Preconditions:	1. The user has access to the website. 2. The user has already registered an account in the system.
Postconditions:	1. The user is logged into their account and has access to personalized features. 2. The system creates a session for the user to track their activity during this login.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user navigates to the "Login" page. 2. The user is prompted to enter their login credentials: ● Username or email address ● Password 3. The system validates the credentials: ● Checks if the username/email exists. ● Verifies if the password matches the stored password for that account. 4. Upon successful validation, the system logs the user in and grants access to their account dashboard. 5. The system will display the home dashboard.

Alternative Flows:	1. Invalid Credentials: If the username/email or password is incorrect, the system displays an error message and prompts the user to retry. 2. Forgot Password: If the user has forgotten their password, they can select a "Forgot Password?" option, initiating the password recovery process.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	1.3		
Use Case Name:	Password Management		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	System, registered user
Description:	The "Password Management" use case allows users to securely manage their account passwords. This includes resetting a forgotten password, updating the current password,
Preconditions:	1. The user has access to the website login page. 2. The user has an active account but has forgotten their password OR the user intends to change his password
Postconditions:	The user’s password is successfully reset, and they can log in with the new credentials.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user selects the "Forgot Password?" option on the login page OR the user selects “Reset Password” at the profile page 2. The system prompts the user to enter their registered email address. 3. The system verifies the provided information and sends a password reset link (via email). 4. The user receives the reset link and clicks the link, which redirects them to a password reset page. 5. On the password reset page, the user enters and confirms a new password. 6. The system validates the new password based on predefined security criteria (e.g., length, complexity).

	7. Upon successful validation, the system updates the user's password and confirms the change. 8. The user is notified that the password has been successfully reset and can now log in using the new password.
Alternative Flows:	1. Invalid Email/Contact Number: If the email or contact number provided by the user is not recognized, the system prompts the user to retry or contact support. 2. Expired Reset Link: If the reset link or code has expired, the system prompts the user to request a new one.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	2.1		
Use Case Name:	Profile Viewing		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	System, registered user
Description:	The "Profile Viewing" use case allows a registered user to view their personal profile details, including points balance and reward history, ensuring that their information is accurate and up-to-date.
Preconditions:	1. The user has an existing account in the system. 2. The user must be logged into their account.
Postconditions:	The user successfully views their profile details.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user navigates to the “Rewards” page. 2. The system retrieves the user’s profile data and displays the information including user’s point balance, reward history, and the rewards that can be redeemed. 3. The user then navigates to the “Profile” page and selects “Personal Information”. 4. The system retrieves the user’s profile data and displays the information including username, email and address.
Alternative Flows:	
Exceptions:	

Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	2.2		
Use Case Name:	Profile Editing		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	System, registered user
Description:	The "Profile Editing" use case allows a registered user to update their profile information, such as their email address,username, profile picture and address.
Preconditions:	1. The user has an existing account in the system. 2. The user must be logged into their account.
Postconditions:	1. The user's profile information is updated successfully. 2. The user receives confirmation of the changes made.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user navigates to the "Profile" page. 2. The user selects "Update personal information". 3. The user updates the desired fields (e.g., email address, username, profile picture). 4. The system validates the new input (e.g., ensuring email format is correct, unused username). 5. The user submits the changes. 6. The system updates the profile with the new information and logs the changes. 7. The user receives a confirmation message that the profile has been successfully updated.
Alternative Flows:	Invalid Input - If the system detects invalid input (e.g., incorrect email format, invalid phone number), it displays an error message. The user is prompted to correct the input and resubmit the changes.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	3.1		
Use Case Name:	Upload functionality		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	1. Registered user (resident of the neighborhood) 2. System (reporting application)
Description:	Upload functionality use case describes the process by which a registered user uploads a photo of the issue in the neighborhood as well as a description of the issue.
Preconditions:	1. The user has signed up for an account in the app. 2. The user has logged into his account. 3. The user has identified an issue in the neighborhood that he wants to report. 4. The user has a device that can take photos and upload it.
Postconditions:	1. The photo and description are uploaded and stored temporarily for validation. 2. The system notifies the user that the upload is successful and waits for the validation and processing.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user navigates to the Report page and selects Make new reports. 2. The user selects a photo from his device or takes a new photo. 3. The user keys in details of the issue, the length of the description should not exceed the word/character limit. 4. The user selects the location where he saw the issue. 5. The user selects the date and time when he saw the issue. 6. The system notifies the user that the upload was successful.
Alternative Flows:	If the user cancels halfway during the reporting process, the system aborts the upload process.
Exceptions:	If the system fails to store the photo or description, an error message will be shown and ask the user to resubmit.
Includes:	

Special Requirements:	
Assumptions:	The user knows how to navigate to the report feature in the app.
Notes and Issues:	

Use Case ID:	3.2		
Use Case Name:	Validation		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	1. Registered user (resident of the neighborhood) 2. System (reporting application)
Description:	In the validation use case, the system checks the file type, size, and description length to ensure they meet the requirements.
Preconditions:	1. A photo and description have been uploaded by the user. 2. The system has stored the uploaded content temporarily.
Postconditions:	1. The photo and description are either approved for submission or rejected with an error message. 2. The user is informed of the validation outcome.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system checks that the uploaded photo is in an acceptable format (eg. JPEG, PNG). 2. The system checks that the photo does not exceed the maximum allowed size (eg. 5MB). 3. The system checks that the description does not exceed the maximum length limit. 4. If all checks pass, the system moves to the metadata capture or final submission. 5. The system informs the user of the validation outcome.
Alternative Flows:	1. If the file type or size is invalid, the system prompts the user to upload a valid file. 2. If the description is too short or too long, the system prompts the user to adjust the text.
Exceptions:	If the system encounters an error during validation, the user is notified, and the validation process is restarted.
Includes:	

Special Requirements:	
Assumptions:	The system can process and validate various photo formats and description lengths efficiently.
Notes and Issues:	

Use Case ID:	3.3		
Use Case Name:	Metadata capture		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	1. Registered user (resident of the neighborhood) 2. System (reporting application)
Description:	In the metadata capture use case, the user is prompted to add details related to the uploaded photo or location before final submission.
Preconditions:	1. The photo and description have passed validation 2. Metadata capture is optional for the user to add on more details if he wishes.
Postconditions:	1. Metadata is captured and associated with the photo and description. 2. The system is ready for final submission or further processing.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system prompts the user to add or confirm metadata, such as location. 2. The user manually enters or adjusts metadata as needed. 3. The system checks if the metadata is complete. 4. The system associates the metadata with the uploaded photo and description. 5. The system confirms that metadata capture is complete and notifies the user.
Alternative Flows:	If the user skips metadata entry, the system uses default or previously captured data.
Exceptions:	If the metadata is incomplete, the system prompts the user to correct it before proceeding.

Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4		
Use Case Name:	Points System		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	**Registered User**: A resident of the neighborhood who has signed up and logged into the system. **System**: The gamified reporting system that awards and tracks points. **Moderator**: An individual or automated system responsible for reviewing submissions before awarding points.
Description:	The process through which a registered user is awarded points for submitting reports (photos and descriptions), where each submission is moderated before points are awarded. The user can also track their points balance and transaction history.
Preconditions:	1. The user has signed up for an account in the app. 2. The user has logged into their account. 3. The user has submitted a report, including both a photo and description, which is pending moderation. 4. The user has access to a device capable of taking photos and uploading them to the system.
Postconditions:	1. Points are awarded to the user only after their submission has been reviewed and approved by a moderator. 2. The system updates the user's points balance and records the transaction history. 3. The user is notified whether their submission was accepted or rejected. 4. The user can view their current points balance and transaction history.

Priority:	High
Frequency of Use:	1. Points awarding occurs after a submission is approved through moderation. 2. Users can view their points balance and transaction history at any time.
Flow of Events:	**Points Awarding:** 1. The user submits a report with a photo and description. 2. The system stores the submission temporarily and sends it to the moderation queue. 3. The moderator reviews the submission and either approves or rejects it based on predefined criteria. 4. If the submission is approved, the system awards points to the user based on the quality and relevance of the submission. 5. The system will then notify the appropriate authorities to resolve the issue. 6. The system updates the user's points balance and transaction history. 7. The system notifies the user that points have been awarded. 8. If the submission is rejected, the system notifies the user of the rejection without awarding points. **Points Tracking:** 1. The user navigates to their profile or points page. 2. The system displays the user's current points balance. 3. The user selects an option to view their points transaction history. 4. The system shows a list of past transactions, including the date, time, and description of each points award or deduction.
Alternative Flows:	**Rejected:** If the submission does not meet the criteria after moderation, the system rejects the submission and notifies the user that no points have been awarded. **Resubmission:**

	After a rejection, the user may edit and resubmit the report for further review.
Exceptions:	1. If the submission fails to upload (e.g., due to technical issues), the system displays an error message and asks the user to try again.
Includes:	
Special Requirements:	
Assumptions:	1. Users understand that all submissions are moderated and that points are only awarded after approval. 2. Predefined criteria for moderation and points awarding are in place and consistently applied.
Notes and Issues:	1. The system should ensure minimal delay in moderating reports to maintain user engagement. 2. Moderation should strike a balance between speed and quality control.

Use Case ID:	5		
Use Case Name:	Redeem Gifts from the System		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	Registered User, System
Description:	This use case allows a registered user to redeem gift cards and vouchers from the system using their accumulated points. The user can browse the reward catalog, select an item for redemption, and receive the item upon successful transaction completion.
Preconditions:	1. The user must have an active account in the system. 2. The user must be logged into their account. 3. The user must have a sufficient points balance for redemption.
Postconditions:	1. The selected gift card or voucher is successfully redeemed. 2. The user's points balance is updated. 3. The user receives confirmation of the redemption, and the gift card or voucher is delivered to their email and made available in their account.

Priority:	
Frequency of Use:	Varies depending on user engagement and reward offerings.
Flow of Events:	1. The user navigates to the “Rewards” section, then the system displays the reward catalog to the user. 2. The user can browse the reward catalog and the system displays available gift cards and vouchers, including their names, descriptions, values, and points required for redemption. 3. The user can filter the reward catalog (Optional), then the user can filter items by category, value, and availability. 4. The user can select a gift card or voucher for redemption.. 5. Then the system checks the user's points balance. If the user's points balance is equal to or greater than the points required, the system deducts the exact number of points from the user's balance. 6. The system makes the redeemed gift card or voucher available in the user's account. 7. The system displays a confirmation message to the user and logs the redemption transaction. 8. The user can choose to exit and he will be sent to the rewards main page.
Alternative Flows:	AF-S7: insufficient points: 1. If the user attempts to redeem with insufficient points the system displays an error message indicating the issue. 2. The user can either earn more points or select a different item. AF-S8: Item Out of Stock: 1. If the selected gift card or voucher is no longer available, the system informs the user that the item is out of stock. 2. The user can choose another item from the catalog
Exceptions:	EX-1: System Down During Redemption: 1. If the system experiences a failure during the redemption process, the system should ensure no points are deducted. 2. The system informs the user of the issue and logs the incident for review. EX-2: Network Error During Transaction:

	1. If a network error occurs after the points have been deducted but before the item is delivered, the system should retry the delivery. 2. If retry fails, the system must refund the points and notify the user.
Includes:	
Special Requirements:	
Assumptions:	1. The user has sufficient points for at least one reward in the catalog. 2. The system catalog is regularly updated to reflect the current availability of rewards
Notes and Issues:	

Use Case ID:	6		
Use Case Name:	Severity Ranking and Reporting		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	Registered User, System, Moderator/Admin, Relevant Authorities
Description:	The severity assessment functionality describes how the system automatically evaluates submitted reports (photos and descriptions) for severity and urgency using OLLAMA LLM. Points are awarded and authorities are informed based on these assessments. Manual review is triggered for cases with low photo relevance.
Preconditions:	1. The user has signed up for an account and logged in. 2. The user has submitted a report containing both a photo and a description. 3. OLLAMA LLM is properly integrated with the system to perform the severity, urgency, and relevance assessments.
Postconditions:	1. The system rates the severity and urgency of the case on a scale from 1 to 10 using OLLAMA LLM.

	2. If the photo's relevance to the description is rated above 5, points are awarded to the user and relevant authorities are notified. 3. If the relevance rating is 5 or below, the report is sent to a manual review queue for further assessment.
Priority:	High
Frequency of Use:	The severity assessment is used each time a new report is submitted.
Flow of Events:	Severity Assessment and Automated Reporting 1. The user submits a report with a photo and description. 2. The system processes the report and sends the description and photo to OLLAMA LLM for analysis. 3. OLLAMA LLM evaluates the description based on: ● Severity (rated 1-10) ● Urgency (rated 1-10) 4. OLLAMA LLM checks the relevance of the photo to the description (rated 1-10). 5. If the relevance rating is greater than 5: ● Points are awarded to the user based on a predefined equation. ● The system automatically notifies relevant authorities based on the severity rating. The system updates the user's points balance and logs the transaction. Manual Review 1. If the relevance rating is 5 or below, the system sends the case to the manual review queue. 2. A moderator reviews the case and manually adjusts the severity or urgency rankings if needed. 3. After manual review, points may be awarded, and authorities may be notified if applicable.
Alternative Flows:	Manual Adjustment of Severity/Urgency 1. If a moderator feels the severity or urgency rating given by OLLAMA LLM is incorrect, they can adjust it manually. 2. The system updates the report based on the moderator's adjustments, and authorities are notified accordingly. Resubmission

	1. If the report is flagged for low relevance, the user may choose to edit the description or upload a new photo and resubmit for reassessment.
Exceptions:	If OLLAMA LLM fails to process the report, the system displays an error message and automatically sends the case for manual review. If the system fails to notify authorities, an error message is logged, and the report is flagged for admin follow-up.
Includes:	1. OLLAMA LLM integration for assessing severity, urgency, and relevance. 2. Notification system to inform authorities based on severity. 3. Points calculation system based on the relevance rating.
Special Requirements:	
Assumptions:	1. The user submits a report with a clear photo and description. 2. OLLAMA LLM has been pre-configured with appropriate parameters for evaluating severity, urgency, and relevance.
Notes and Issues:	

Use Case ID:	7.1		
Use Case Name:	Admin Management		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	Registered User, System
Description:	allows admins to manage various aspects of the system through the admin dashboard, including user accounts, user submissions, and the reward catalog. Admins can perform actions such as activating/deactivating accounts, monitoring submissions, and managing the reward catalog.
Preconditions:	The admin must have an active account with appropriate privileges. The admin must be logged into their account.

Postconditions:	Admin actions such as user management, submission monitoring, and reward catalog management are successfully executed. The system records and updates all changes made by the admin
Priority:	
Frequency of Use:	

Flow of Events:	1. The admin can access the admin dashboard to do various actions such as managing user accounts, user submission, points, and rewards.
	2. If the admin choses to manage user accounts the system displays a list of all user accounts.
	    1. The admin can view, deactivate, or activate user accounts.
	    2. The admin can reset passwords for user accounts if needed.
	3. If the admin choses to monitor user submissions the system displays a list of all user submissions.
	    1. The admin can filter submissions by date, status, or severity.
	    2. The admin can view details of each submission.
	4. If the admin choses to overseas points and rewards the system displays a summary of points distribution across users.
	    1. The admin can adjust points for individual users.
	    2. The admin can view the history of points transactions for each user.
	5. If the admin choses to manages the reward catalog
	
	1. The admin can add new gift cards and vouchers:
	
	1. The system allows the admin to enter the name, description, and value of the new item.
	
	2. The system allows the admin to specify the points required for redemption.
	
	3. The system saves the new item in the reward catalog.

	2. The admin can update existing gift cards and vouchers: 1. The system allows the admin to select an existing item for editing. 2. The system allows the admin to update the name, description, value, or points required for the item. 3. The system saves the updated details. 3. The admin can remove gift cards and vouchers: 1. The system allows the admin to select an existing item for removal. 2. The system confirms the removal action before deletion. 3. The system updates the catalog to reflect the removal. 6. If the admin choses the exit option the admin will go back to the home page.
Alternative Flows:	AF-S9: If admin action fails. 1. The system should display an error message 2. The admin can retry the action.
Exceptions:	EX-1: If the system is down due to connectivity issues. 1. The system rolls back any changes recently made. 2. The admin can retry once the system is back online.
Includes:	
Special Requirements:	The system must securely handle sensitive information like user account details and submission data.
Assumptions:	
Notes and Issues:	

Use Case ID:	7.2		
Use Case Name:	Authority Notification and Response		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	Authority, System

Description:	allows the system to notify authorities when a severe or high-impact submission is made. Authorities can then respond by updating the status of the submission, ensuring that the issue is addressed and tracked.
Preconditions:	1. The authority must have an active account with appropriate privileges. 2. The authority must be logged into their account. 3. A severe or high-impact submission has been made.
Postconditions:	The related users and authorities are notified of severe or high-impact submissions. Authorities update the status of the submission, and the system records these updates.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system notifies authorities of severe or high-impact submissions, including the details in the notification. 2. The authority can access the submission linked to the notification. 3. The authority can update the status of the submission (e.g., in progress, resolved). 4. The system records the progress updates and makes them visible to relevant users and admins. 5. The authority can choose to exit and he will go back to the home page.
Alternative Flows:	AF-S10: Notification Delivery Issues**:** 1. If a notification fails to reach the authority due to a network or system issue, the system logs the failure and retries sending the notification. 2. If retry fails, the system alerts the admin to resolve the issue manually.
Exceptions:	EX-1: System Failure During Authority Response:

	1. If the system experiences a failure during the authority's response, the system should save the current status and inform the authority to retry the action once the system is restored.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	8.1		
Use Case Name:	User notifications		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	1. Registered user (resident of the neighborhood) 2. System (reporting application)
Description:	User notification use case allows users to receive notifications for key events such as successful submissions, points earned, and reward redemptions. Notifications can be in the form of emails or in-app alerts.
Preconditions:	1. The user has signed up for an account in the app. 2. The user has logged into his account. 3. The user has allowed the app to send notifications. 4. The user has performed an action that sends notification.
Postconditions:	The user receives relevant notification via email or SMS or in-app alert.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user performs an action such as reporting an issue that triggers a notification. 2. The system generates a notification message corresponding to the event. 3. The system checks the user's notification preferences (email

	or SMS or in-app). 4. The system sends the notification via the preferred method. 5. The user receives the notification and can view it in their email inbox or SMS or in-app notifications section.
Alternative Flows:	If the user has not specified a notification preference, the system defaults to in-app notifications.
Exceptions:	If the user's email address or contact number is invalid or the in-app notification system is down, the user is alerted the next time they log in.
Includes:	
Special Requirements:	
Assumptions:	The system's notification service is functioning correctly and has minimal downtime.
Notes and Issues:	

Use Case ID:	8.2		
Use Case Name:	Authority alerts		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	1. Registered authorities (officers, entities) 2. System (reporting application)
Description:	Authority alerts use case allows authorities to receive alerts when a user submits a report classified as high-severity or critical based on the severity rating. These alerts are sent promptly to ensure immediate action can be taken.
Preconditions:	1. The system has classified a fault report as high-severity based on predefined criteria. 2. The relevant authorities are registered in the system and available to receive alerts.
Postconditions:	1. The relevant authorities receive an alert for the high-severity issue.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system classifies a submitted fault report as high-

	severity based on its content. 2. The system generates an alert message consisting the details of the high-severity issue and its location. 3. The system identifies the appropriate authorities based on the location and type of issue. 4. The system sends the alert to the relevant authorities via email, SMS or any suitable communication channel. 5. The authorities receive the alert and acknowledge it in the system and take follow-up actions.
Alternative Flows:	If the authorities fail to acknowledge the alert within a specified time, the system escalates the alert to higher-level officials or sends a reminder.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	The system is able to correctly classify the severity of the fault reports based on the information provided by users and the predefined criteria.
Notes and Issues:	

Use Case ID:	9.1		
Use Case Name:	View reports extends: <search for reports> <sort reports>		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	User, System, Database
Description:	View Reports use case allows users to view all active and past reports they have made, displayed from most recent to least recent
Preconditions:	1. User is logged in 2. There is an internet connection 3. Report tab is selected
Postconditions:	Database query is successful
Flow of Events:	1. User clicks Report tab at the bottom 2. The system queries the database to retrieve all reports made

	by User 3. Reports are retrieved and sorted by most recent to least recent reports 4. Reports are displayed under ‘Active Reports” and ‘Past Reports” respectively
Alternative Flows:	1. User has not made a report before - shows None
Includes:	User Login Use Case
Assumptions:	Internet connection

Use Case ID:	9.2		
Use Case Name:	Sort reports		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	User, System, Database
Description:	Sort Reports allow users to sort their reports based on categories of issue, severity of issues, and time of report
Preconditions:	1. User is logged in
Postconditions:	Database query is successful
Flow of Events:	1. User enters keywords in the search bar 2. The system queries the database and retrieve reports whose title or description includes the keyword 3. The reports are displayed based on current sorting option (default: most recent - least recent)
Alternative Flows:	1. No reports found - shows None
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	1. Internet connection
Notes and Issues:	

Use Case ID:	9.3		
Use Case Name:	Search for reports		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

○

Actor:	User, System, Database
Description:	Search for Reports allow users to search for their submitted reports by entering keywords
Preconditions:	2. User is logged in
Postconditions:	Database query is successful
Flow of Events:	4. User enters keywords in the search bar 5. The system queries the database and retrieve reports whose title or description includes the keyword 6. The reports are displayed based on current sorting option (default: most recent - least recent)
Alternative Flows:	2. No reports found - shows None
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	2. Internet connection
Notes and Issues:	

Use Case ID:	10		
Use Case Name:	Reporting & Analytics		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	Registered User, System, Admin
Description:	This use case allows admin to generate reports on existing users data such as user submission, points distribution, and reward redemption. The admin can search for users through their email or username, view detailed reports on user submissions, point transaction, and analyse the severity and types of issues reported

Preconditions:	1. Admin must have an active account in the system 2. Admin must be logged into their account 3. System must have existing data on user submission, points distribution and reward redemptions
Postconditions:	1. Admin successfully generates and view the required reports 2. Admin can analyze the severity of issue reported
Priority:	
Frequency of Use:	
Flow of Events:	1. Admin navigates to the “Reporting and Analytics” section 2. System displays “Reporting and Analytics” dashboard to the Admin 3. Admin enters the required details to search for user 4. System validates the input given by the admin and generate the report
Alternative Flows:	1. If admin provides an invalid input system 2. System will not display reports to admin 3. System will prompt admin again for the correct input
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Non-Functional Requirements

1. **Performance**
    - **Response Time:** The application should handle user interactions (e.g., uploading photos, redeeming rewards) with minimal latency (e.g., under 3 seconds).
    - **Load Handling:** The system should efficiently handle multiple simultaneous users and submissions without significant degradation in performance.
2. **Scalability**
    - **Growth Management:** The application should be scalable to accommodate increasing numbers of users, submissions, and data without performance degradation.
3. **Security**
    - **Data Protection:** User data, including photos and personal information, must be stored securely and encrypted.
    - **Access Control:** The system should implement role-based access controls to ensure that only authorized users and admins can perform certain actions.
    - **Password Storage:** Users' passwords should be stored securely using hashing algorithms.
4. **Usability**
    - **User Interface:** The UI should be intuitive and user-friendly, with clear navigation and instructions.
    - **Accessibility:** The application should be accessible to users with disabilities, complying with standards such as WCAG (Web Content Accessibility Guidelines).
5. **Reliability**
    - **Uptime:** The application should be available and functional 24/7, with minimal downtime.
    - **Error Handling:** The system should gracefully handle errors and provide meaningful error messages to users.
    - **Backup and Recovery:** Regular backups of user data and submissions should be maintained, and the system should include a disaster recovery plan to restore functionality and data in case of a major failure.
6. **Compatibility**
    - **Browser Support:** The web application should work across major browsers (e.g., Chrome, Firefox, Safari, Edge).
    - **Device Compatibility:** The application should be responsive and functional on various devices, including desktops, tablets, and smartphones.
7. **Maintainability**
    - **Code Quality:** The codebase should be well-documented and follow coding standards to facilitate future maintenance and updates.
    - **Testing:** The application should have automated unit tests to ensure functionality and stability.
8. **Data Privacy**

- **Compliance:** The application must comply with PDPA.
- **Privacy Policy:** A clear privacy policy should be provided to users, explaining how their data is collected, used, and protected.

IV) Data Dictionary:

1. Photo: A visual image file uploaded by the user of acceptable file type and file size.

2. Description: A text field providing a detailed explanation of the photo or issue being reported, not exceeding the character limit set by the system.

3. Metadata: Additional information (such as location) providing context about the photo or issue if the user wishes.

4. Notification: Alerts or messages sent to users or authorities when key events occur.

5. Notification Method: The method or channel through which notifications are delivered (e.g., email, SMS, app push notifications). The user will select his preferred method.

6. High Severity Issues: Issues that are classified as critical (high rating) by the system based on predefined criteria and require immediate attention, triggering urgent notifications to relevant parties.

7. Predefined Criteria: Set rules or conditions used to evaluate whether an issue qualifies as high severity or for other classifications. Set by the engineer to standardize issue evaluation.

8. Relevant Authorities: Entities (e.g., government departments, maintenance teams) are notified when certain issues are reported or escalate. It is defined based on the type of issues, its severity, and its location; multiple authorities can be notified at the same time.

9. User: A participant in the system who has the ability to browse the reward catalog, accumulate points, and redeem rewards. Each user has a unique identifier and a points balance that tracks the number of points they have earned and can spend.

10. Reward Catalog: A collection of available gift cards and vouchers that users can redeem using their points. The catalog includes items categorized by type, value, and availability. Users can filter and browse the catalog to find items that interest them.

11. Gift Card/Voucher: An item within the reward catalog that users can redeem in exchange for points. Each gift card or voucher has a specific value, a detailed description, and a

points requirement. Gift cards and vouchers can belong to different categories, and their availability may vary.

12. Redemption: The process by which a user exchanges their points for a selected gift card or voucher from the reward catalog. This process includes verifying the user's points balance, deducting the required points, and delivering the redeemed item to the user.

13. Transaction: A record of points-related activities, including the accumulation and deduction of points. Transactions are logged whenever a user redeems a gift card or voucher, providing a history of all points movements in the user's account.

14. Admin: An individual with elevated privileges within the system responsible for managing user accounts, monitoring user submissions, and overseeing the points and rewards system. Admins have access to specialized tools and dashboards that allow them to perform these tasks effectively.

15. Admin Dashboard: A specialized interface accessible only to admins, providing tools to manage user accounts, monitor submissions, and oversee points and rewards. The dashboard displays relevant information and allows admins to take actions such as activating/deactivating accounts, resetting passwords, and managing the reward catalog.

16. User Account Management: The set of functionalities within the admin dashboard that allows admins to manage user accounts. This includes viewing all user accounts, activating or deactivating accounts, and resetting passwords. These actions help maintain user access and security.

17. User Submissions Monitoring: A feature of the admin dashboard that allows admins to monitor and manage user submissions. This includes viewing a list of all submissions, filtering submissions by criteria such as date, status, or severity, and viewing detailed information about each submission. It ensures that admins can efficiently track and address issues reported by users.

18. Points and Rewards Oversight: An admin functionality that allows admins to monitor and manage the distribution of points and rewards across users. The dashboard provides a summary of points distribution, allows admins to adjust points for individual users, and view the history of points transactions. This ensures the integrity and fairness of the rewards system.

19. Reward Catalog Management: A set of actions that admins can perform within the admin dashboard to manage the reward catalog. This includes adding new gift cards and vouchers, updating existing items, and removing items from the catalog. The system provides tools for admins to enter item details, adjust points requirements, and confirm removal actions, ensuring the reward catalog remains up-to-date and accurate.

20. Authorities: Individuals or entities who are notified by the system when a severe or high-impact submission is made. Authorities are responsible for addressing and resolving these submissions and can update the progress of solving the issue within the system.

21. Notification System: A feature that alerts relevant authorities when a severe or high-impact submission is made by a user. The system classifies submissions based on severity and impact and sends notifications with the submission details to the appropriate authorities. This ensures that critical issues are promptly addressed.

22. Submission Severity Classification: The process by which the system evaluates and categorizes user submissions based on their severity and impact. Submissions classified as severe or high-impact trigger notifications to relevant authorities, ensuring that significant issues receive immediate attention.

23. Progress Update on Submissions: A functionality that allows authorities to update the status of a submission linked to a notification. Authorities can mark the submission as "in progress," "resolved," or other statuses as appropriate. The system records these updates and makes them visible to relevant users and admins, providing transparency in the resolution process.

24. User Submission: A user submission consists of images of the issue, the description that was provided, location of the issue, time, and date of the report.

25. Report: A report consists of a User Submission, a self-generated title of “<Issue> at <Location>, <Time & Date>” and authority responses, if any.

26. Points Transaction: A record of points awarded, spent, or accumulated by a user.

27. Reward Redemption: A record of rewards redeemed by a user using their points.

28. Severity Rating: A score determined by OLLAMA LLM on a scale of 1 to 10 that measures the severity of the reported issue.

29. Relevance Rating: A score from 1 to 10 that evaluates how closely the photo aligns with the description provided by the user.

30. Urgency Rating: A score from 1 to 10 that measures the urgency of resolving the reported issue.

31. Moderation Outcome: The result of a manual or automated review process determining whether points are awarded, rejected, or adjusted for a submission.

32. Points Capping: The maximum number of points that can be awarded for a single submission.

33. Points Transaction History:
    A detailed log of all points-related activities for each user, including points earned, deducted, and spent.
    Attributes:
    ● Submission Status: The status of each submission (e.g., pending review, approved, rejected).
    ● Points Earned/Deducted Reason: A brief explanation for why points were added or deducted (e.g., rejected report, claimed reward).
    ● Transaction Categories: Sections for categorizing points transactions (earned from reports, deducted from rejected reports, redeemed for rewards).

34. Points Adjustment Notification:
    A message notifying the user when their points are adjusted due to moderation or other factors.

35. Severity Categories:
    Classifications used to assess and rank submissions based on their impact.
    Attributes:
    ● Safety Hazards: A category for issues that pose a risk to public safety.
    ● Environmental Issues: A category for issues that affect the environment.
    ● Public Health Risks: A category for issues that endanger public health.

36. Severity Ranking:
    A detailed justification of the severity ranking, including key factors such as proximity to critical zones (e.g., school zones). User has the option to indicate if the user has manually reported the issue as "urgent."

37. Severity Threshold:
    The predefined score that determines whether the system will take certain actions (e.g., reporting to authorities).
    Attributes:
    ● Threshold Score: A severity score (e.g., 7 or above) that triggers automated reporting or user notifications.
    ● Action Trigger: The action to be taken when the threshold is met (e.g., notify authorities, escalate to manual review).

38. Authority Notification:
    Attributes:
    ● Escalation Trigger: The predefined severity score that triggers notifications to relevant authorities.
    ● Report Details: Includes the location, date, time, severity ranking, issue description, and photo submitted.

● Photo Quality Check: Confirmation that the submitted photo meets predefined quality standards for reporting.

39. User Report Status:
    The ability for users to track the status of their submission after it has been escalated to authorities.
    Attributes:
    ● Submission Status: The current progress of the report (e.g., "Forwarded to Authorities," "Under Review," "Resolved").
    ● Update Alerts: Notifications sent to users about the status changes of their submission.

40. Manual Review Report:
    A report generated for admins that contains the necessary details to evaluate a submission.
    Attributes:
    ● Original Severity Ranking: The severity score originally assigned by the OLLAMA LLM.
    ● Submission Metadata: Details such as the photo, description, and relevance score.
    ● Admin Comments: Notes left by the admin explaining adjustments to the severity ranking.
    ● Photo Relevance Score: How well the photo corresponds to the description, helping in decision-making.

41. Admin Adjustment Log:
    A record of any changes made by admins during the manual review process.
    Attributes:
    ● Adjustment Reason: Explanation for changing the severity ranking (e.g., misclassification).
    ● Updated Severity Ranking: The new score assigned after admin review.
    ● Adjusting Admin: The admin or moderator responsible for making the adjustment.

42. User Appeal:
    The ability for users to challenge the severity ranking or adjustments made during manual review.
    Attributes:
    ● Appeal Status: The current state of the user’s appeal (e.g., "Submitted," "Under Review," "Resolved").
    ● Appeal Evidence: Additional information or evidence provided by the user to support the appeal.