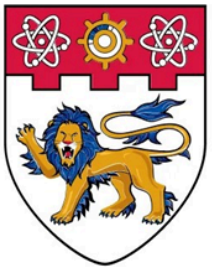


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006 - Software Engineering
Lab 2 Deliverables

Lab Group	SCEB
Team	Team 1
Members	AKANKSHA MATHUR (U2323265C)
	HU HAN (U2320036H)
	PHUA NAOMI (U2422699H)
	KARTHIK RAJ S/O MOHAN(U2320917J)
	LI YUJIA (U2320574C)
	WANG KE (U2320276E)

Table of Contents

1. Use Case Descriptions and Diagrams

1.1 Use Case Descriptions

1.1.1

Use Case ID	#1-1
Use Case Name	Log in/Sign up
Use Case History	**Date Created: February 6** **Date Last Updated: February 12**
Actor	User
Description	Users can log in or sign up to the platform to create and manage their account.
Preconditions	1. The user device must be connected to WiFi/Cellular Data. 2. The system currently does not have the user logged in 3. The user has an existing and verified account. connected to a valid email and password stored in the database to log in. 4. The hosted database must be online
Postconditions	1. User is authenticated and can access the platform.
Priority	High
Frequency of Use	High
Flow of events	1. User navigates to the login/signup page 2. Users input their email and password. 3. The system verifies the email and password . 4. If successful, the user is logged in; otherwise, an error message is displayed.
Alternative Flows	Forgot password flow 1. User selects “Forgot Password” option 2. The system prompts the user to enter their email 3. If email exists in database, a password reset link is sent to the user 4. The user follows link to reset password and login Signup flow 1. User selects “Sign Up” option 2. The user provides necessary details 3. System validates the inputs and checks for an existing account 4. If none, the user account is activated and the user is logged in

Exceptions	Incorrect Login credentials 1. System displays an error message 2. Does not allow user to login Multiple Failed Login attempts 1. Account is temporarily locked 2. An email is sent to user to reset their password
Includes	1. Email/Password authentication 2. Password reset function
Special Requirement	1. Password must be encrypted in database 2. Authentication process must be secure
Assumptions	1. Users have a valid email address
Notes/Issues	None

1.1.2

Use Case ID	#1-2
Case Name	Profile Update
Use Case History	**Date Created: 6 February** **Date Last Updated: 12 February**
Actor	User
Description	Users can update their personal information.
Preconditions	1. The user must be logged in. 2. The system must be online and connected to the database
Postconditions	1. User profile is successfully updated. 2. The system stores the updated information into the database
Priority	High
Frequency of Use	Varies
Flow of events	1. User accesses profile settings. 2. User modifies the necessary details (e.g changing password). 3. The system saves and updates the profile.
Alternative Flows	1. User cancels the update before saving changes
Exceptions	1. System is offline or disconnected from database, user will not be able to update 2. Invalid email or weak password

Includes	1. Email authentication 2. Password strength validation
Special Requirement	3. Password must be encrypted in database 4. Authentication process must be secure
Assumptions	None
Notes/Issues	None

1.1.3

Use Case ID	#1-3
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Case Name	Password Recovery
Actor	User
Description	Users can recover their password by providing the registered email.
Preconditions	1. User has a registered email. 2. The system's email service is online and operational
Postconditions	1. Password is reset, and the user can log in. 2. The new password is stored in the database
Priority	High
Frequency of Use	Low
Flow of events	1. User requests a password reset. 2. User enters the email associated with the account. 3. The system sends a password reset link. 4. User resets the password and logs in.
Alternative Flows	The user enters an invalid/unregistered email 1. System displays an error message and allows user to try again
Exceptions	1. Reset link expires before user uses it 2. The user enters a weak or invalid password, system sends an error message
Includes	1. Email validation 2. Password encryption
Special Requirement	1. The reset link expires after a set duration 2. Password strength check
Assumptions	None

Notes/Issues	None

1.1.4

Use Case ID	#1-4
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Case Name	Post
Actor	User
Description	Users can create posts to describe safety issues or emergencies or other relevant things.
Preconditions	1. 1User is logged in. 2. The system must be online and connected to the database
Postconditions	1. Post is successfully uploaded to the forum. 2. Other users can view/comment/like the post
Priority	High
Frequency of Use	High
Flow of events	1. User clicks the "Post" button. 2. User inputs title, description, and category. 3. User attaches tags or categories. 4. The system posts the message to the forum.
Alternative Flows	1. User cancels post before submitting, the system discards any unsaved changes 2. User saves a draft of the post
Exceptions	1. The system is offline, preventing post from being submitted 2. The post violates community guidelines and is flagged for review 3. The database is offline, preventing post from saving
Includes	1. Content moderation 2. User authentication verification
Special Requirement	1. Users follow community guidelines 2. The system can handle high numbers without
Assumptions	None
Notes/Issues	None

1.1.5

Use Case ID	#1-5
Use Case History	**Date Created: 6 February** **Date Last Updated: 21 February**
Case	Edit/Delete Post
Actor	User
Description	Users can modify or delete posts that they created
Preconditions	1. User is logged in. 2. The system must be online and connected to the database 3. User must be post owner
Postconditions	1. If edited, post is successfully modified and displayed with relevant changed 2. if deleted, post is successfully deleted and no longer visible 3. The database is updated accordingly
Priority	Medium
Frequency of Use	High
Flow of events	1. User access their post from their profile 2. The user makes edits/deletes post 3. The system is successfully updated
Alternative Flows	1. User cancels edit or delete action
Exceptions	1. The system is offline, preventing any changes to be made 2. User attempts to delete or edit a post that they do not own 3. System is not able to connect to the database
Includes	1. User authentication verification
Special Requirement	None
Assumptions	None
Notes/Issues	None

1.1.6

Use Case ID	#1-6
Case	View Posts
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	User
Description	Users can view community posts regarding safety issues.
Preconditions	1. The user must be logged in. 2. The system must be online and connected to the database
Postconditions	1. The system successfully displayed the post. 2. Users can interact with the post.
Priority	High
Frequency of Use	High
Flow of events	1. Users access the "View Posts" section. 2. System displays a list of posts.
Alternative Flows	If no posts are available, the system displays a "No posts found" message and suggests creating a new post.
Exceptions	1. Database connection failure: The system shows an error message and prompts the user to retry. 2. User is not logged in: The system prompts the user to log in before accessing posts.
Includes	Filtered Posts (Users can apply filters to view specific posts).
Special Requirement	The system should support pagination for large datasets.
Assumptions	Users have an active internet connection.
Notes/Issues	The platform must ensure that displayed posts are relevant and updated.

1.1.7

Use Case ID	#1-7
Case	Send Feedback
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	User
Description	Users can send feedback regarding posts or alerts to improve the platform.
Preconditions	1. User is logged in. 2. The system must be online and connected to the database 3. The post must be exist and be accessible
Postconditions	1. Feedback is sent and stored in the system. 2. Moderators may be able to review the content
Priority	Medium
Frequency of Use	Moderate
Flow of events	1. User selects a post or alert to provide feedback on. 2. User submits feedback. 3. The system processes and stores the feedback
Alternative Flows	If the system is down, the user will be prompted to try again later.
Exceptions	1. The feedback form is incomplete: The system prompts the user to provide the required details. 2. The post or alert is removed before submission: The system notifies the user and prevents submission.
Includes	1. View Alerts 2. View Posts
Special Requirement	The system should allow users to provide feedback anonymously if desired.
Assumptions	Users will provide constructive and relevant feedback.

Notes/Issues	There should be a moderation system in place to prevent spam or inappropriate feedback.

1.1.8

Use Case ID	#1-8
Case	Like/Dislike
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	User
Description	Users can like or dislike posts to indicate the usefulness or truthfulness.
Preconditions	1. User is logged in and viewing a post. 2. The system must be online and connected to the database 3. The post must exist and be visible to users
Postconditions	1. Post receives a like or dislike vote.
Priority	Medium
Frequency of Use	High
Flow of events	1. User can view the post 2. User clicks the "Like" or "Dislike" button. 3. The system records the vote.
Alternative Flows	If a user has already voted, clicking the opposite button removes the previous vote and applies the new one
Exceptions	The post is removed before the action: The system notifies the user.
Includes	View Posts
Special Requirement	The system should prevent spam voting (e.g., same user liking/disliking multiple times in a short period).
Assumptions	Users will use the like/dislike feature to indicate content quality.
Notes/Issues	Consider limiting the number of likes/dislikes a user can perform per day to prevent abuse.

1.1.9

Use Case ID	#1-9
Case	Report Post
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	User
Description	Users can report posts as fake news or inappropriate content.
Preconditions	1. User is logged in. 2. The system must be online and connected to the database 3. The post must exist and be visible to users
Postconditions	1. Post is flagged for review by a moderator. 2. The system records the report 3. If post received multiple reports, post may be hidden
Priority	High
Frequency of Use	Moderate
Flow of events	1. User selects a post to report. 2. User chooses the reason for reporting or writes other detailed report reasons. 3. The system sends the report to the moderator.
Alternative Flows	If the post has already been reviewed and dismissed, the system informs the user.
Exceptions	The post is removed before the report is submitted: The system notifies the user.
Includes	View Posts
Special Requirement	The system should track repeated false reports to prevent misuse.
Assumptions	Users will report posts responsibly and not for personal disagreements.
Notes/Issues	Need a clear policy on how reports are handled and who moderates them.

1.1.10

Use Case ID	#1-10
Case	View Alerts
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	User
Description	Users can view real-time safety alerts on the platform.
Preconditions	1. User is logged in. 2. The system must be online and connected to the database
Postconditions	1. Relevant alerts are displayed based on user settings.
Priority	High
Frequency of Use	High
Flow of events	1. User accesses the "View Alerts" section. 2. System displays current alerts.
Alternative Flows	If no alerts are available, the system displays a message: "No active alerts at the moment."
Exceptions	Database connection failure: The system shows an error message and prompts the user to retry.
Includes	Filtered Alerts (Users can apply filters to view specific alerts).
Special Requirement	Alerts must be displayed in real-time without significant delay.
Assumptions	Official sources provide accurate and timely alerts.
Notes/Issues	The system should allow users to filter alerts by location or type.

1.1.11

Use Case ID	#1-11
Case	Search Hospitals
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	User, Google Maps API
Description	Users can search for nearby hospitals using location-based data.
Preconditions	1. User is logged in 2. User has location services enabled. 3. The system must be online and connected to the database 4. The map services must be online
Postconditions	1. Hospitals are displayed based on search criteria.
Priority	High
Frequency of Use	High
Flow of events	1. User enters location or filter criteria. 2. The system fetches hospital data from external sources. 3. System displays the hospital list on a map or in a list format.
Alternative Flows	1. **User location not available:** The user does not have location services enabled or the app cannot access the location. The system could prompt the user to manually enter a location or choose from a predefined set of areas. 2. **Map service not available**: If the Google Maps API is temporarily unavailable, the system could fallback to a basic list display of hospitals or show a cached version of the map if available.
Exceptions	1. Location service failure 2. Database connectivity failure

	3. API error
Includes	1. **Authenticate User**: Verifies if the user is logged in and authenticated before allowing access to search functionalities. 2. **Enable Location Services**: Could be a use case where the system checks and prompts the user to enable location services if they are not already enabled.
Special Requirement	**Real-time data update**: The system must be capable of handling and displaying real-time updates without significant delays to ensure the alerts are current.
Assumptions	1. **User has active internet connection**: Assumed to access the database and external APIs for hospital data. 2. **Google Maps API is reliable**: Assumes that the Google Maps API is consistently available and provides accurate and up-to-date geographic information.
Notes/Issues	High dependency on Google Map API

1.1.12

Use Case ID	#1-12
Case	Display Alerts on Map
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	User, Google Maps API
Description	Alerts are displayed on a map for easier location tracking.
Preconditions	1. User is logged in. 2. User has location services enabled. 3. The system must be online and connected to the database 4. The map services must be online
Postconditions	1. Alerts are marked on the map.
Priority	High
Frequency of Use	High
Flow of events	1. User accesses the map. 2. System displays alerts with markers on the map.

Alternative Flows	1. **User disables location services**: If the user has location services disabled, the system could prompt the user to enable them or default to a general view based on the last known location or a predefined default location. 2. **Map service temporarily unavailable**: If the map service is down, the system could display alerts in a list format or provide basic information without a map visualization.
Exceptions	1. **Failure to load map**: This could occur if there is a problem with the map service provider or if the user has connectivity issues. 2. **Database access error**: Errors might occur in retrieving alert data due to database connectivity issues. 3. **No current alerts**: If there are no current alerts to display, the system should handle this gracefully, possibly by informing the user that there are no current issues.
Includes	1. **Authenticate User**: Ensures that the user is logged in and has appropriate permissions to access the map and alert data. 2. **Fetch Alert Data**: Represents the process of retrieving alert data from the database or an external API, which is necessary before displaying it on the map.
Special Requirement	**Real-time data update**: The system must be capable of handling and displaying real-time updates without significant delays to ensure the alerts are current.
Assumptions	1. **Reliable Internet Connection**: Assumes that users have a stable and reliable internet connection to receive updates and load map data. 2. **Accurate data source**: Assumes that the source of the alert data is reliable and provides timely and accurate information.
Notes/Issues	**Privacy considerations**: Handling location data requires careful consideration of privacy laws and user consent.

1.1.13

Use Case ID	#1-13
Case	Send Alerts
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Moderator

Description	Moderators can update alert information based on official sources.
Preconditions	1. Moderator is logged in and authorized. 2. The system must be online and connected to the database 3. Alerts must be verified before being sent out
Postconditions	1. Alert information is created or updated and stored in the system 2. The alerts are sent out to users.
Priority	High
Frequency of Use	High
Flow of events	1. Moderator receives updated data from external sources. 2. Moderator reviews and updates the system with the latest information. 3. Alerts are sent to users.
Alternative Flows	1. **Source data unavailable**: Occurs when external sources fail to provide updated data, possibly due to technical issues or connectivity problems. The system could fallback to the last known data with a timestamp, notifying users of potential delays in updates. 2. **Moderator unavailable**: If a moderator is not available, the system could be set up to either delay the alert or escalate the task to another authorized moderator.
Exceptions	1. **Verification failure**: When an alert cannot be verified due to conflicting information or lack of confirmation from reliable sources. 2. **System downtime**: When the system or the database is offline, preventing the processing of alerts.
Includes	**Log Activity**: Every action taken by a moderator, such as creating, updating, or sending an alert, is logged for audit and tracking purposes.
Special Requirement	**Real-time processing**: The system must process and send out alerts in real-time or near real-time to ensure that users receive timely information.
Assumptions	Reliable dara source
Notes/Issues	**User feedback mechanism**: Consider implementing a mechanism for users to provide feedback on alerts, which can help improve the quality and relevance of the information sent.

1.1.14

Use Case ID	#1-14
Case	Manage Feedback
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Moderator
Description	Moderators can manage feedback users give.
Preconditions	1. Moderator is logged in and authorized. 2. The system must be online and connected to the database 3. User feedback/reports must exist for post
Postconditions	1. User feedback is reviewed 2. Information is updated and posts reported are checked. 3. Misinforming posts are removed
Priority	Medium
Frequency of Use	Medium
Flow of events	1. Moderator receives feedback and reports. 2. Moderator reviews and updates the system with the latest information. 3. Fake posts will be deleted.
Alternative Flows	**Feedback not actionable**: In cases where the feedback or reports provided by users do not warrant any action (e.g., the feedback is vague or unrelated), the system might guide moderators to close the report without changes.
Exceptions	1. **Feedback system offline**: If the system handling feedback encounters downtime or connectivity issues, moderators might not be able to access new reports. 2. **Database access errors**: Occurs if there are issues connecting to the database to retrieve or update feedback information.

Includes	**Log Actions**: Each action taken by a moderator on feedback or reports should be logged for audit and review purposes.
Special Requirement	**Real-time updates**: The system should update feedback and moderation actions in real-time to ensure that all moderators have the latest information and to avoid redundant work.
Assumptions	**Moderators are trained**: Assumes that moderators have been trained and are familiar with the guidelines and procedures for handling user feedback and reports.
Notes/Issues	**Bias in moderation**: There's a potential issue with moderator bias affecting the neutrality of decisions, which needs to be monitored and managed through training and oversight.

1.1.15

Use Case ID	#1-15
Case	Encrypt User Data
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	System
Description	The system encrypts all users passwords and sensitive information before storing them
Preconditions	1. The user registers or updates personal information 2. The system must be online and connected to the database 3. The system has encryption mechanisms stored and working
Postconditions	1. User data is securely stored in an encrypted format 2. No plain-text personal information is stored in the database
Priority	High
Frequency of Use	High
Flow of events	1. User submits sensitive data 2. The system applies one-way hashing for passwords 3. encrypted data is stored in database
Alternative Flows	**Encryption failure**: If encryption fails due to a technical error or because the encryption mechanisms are not functioning correctly, the system should prevent the data from being saved and inform an administrator or log the incident.
Exceptions	**System offline**: If the system is offline or the database connection is lost during the encryption or storage process, the

	system must ensure that the data is not transmitted or stored in plaintext.
Includes	**User Authentication**: Before a user can update their information, they must be authenticated to ensure that sensitive data is being encrypted and stored by the authorized user.
Special Requirement	**Compliance with data protection regulations**: The encryption process must comply with relevant legal and regulatory requirements such as GDPR, HIPAA, or PCI DSS, which dictate specific standards for the encryption of personal data.
Assumptions	**Reliable encryption algorithms**: Assumes the system uses reliable and tested encryption algorithms approved for use in securing sensitive user data.
Notes/Issues	None

1.1.16

Use Case ID	#1-16
Case	Manage User Accounts
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Administrator
Description	Admins can suspend, delete or restore user accounts due to violations or user request
Preconditions	1. Admin is logged in and authorised 2. User accounts exists 3. The system must be online and connected to the database
Postconditions	1. User is deleted, suspended or restored 2. Changes are stored in the database
Priority	High
Frequency of Use	High
Flow of events	1. Admin searches for the relevant user 2. Admin selects respective action 3. System updates user status 4. User is notified about the change
Alternative Flows	**Suspension Failure:** If the system fails to suspend a user account, an error message will be shown, and the admin will have to retry or take an alternative action.

	Restoration Request Denial: If an admin attempts to restore a deleted account that cannot be restored, the system should notify the admin and provide an option for an alternative solution.
Exceptions	**Admin Permission:** If admin does not have permission to manage the account, an error message is shown and the action will not be performed.
Includes	**User Notification:** Whenever a change is made to the user's account, the user must be notified through email and/or in-app notification, depending on system settings.
Special Requirement	**Security:** The system should ensure that the user's information is protected. If a user's account is deleted, there should be a complete data removal. Any restoration to an account is made, no sensitive data should be compromised
Assumptions	Accounts are suspended, deleted or restored according to the policies of the application.
Notes/Issues	None

1.1.17

Use Case ID	#1-17
Case	Manage Moderators
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Administrator
Description	Admins can add, remove or modify moderator permissions
Preconditions	1. Admin is logged in and authorised 2. Moderator accounts exists 3. The system must be online and connected to the database
Postconditions	1. Moderator's roles are modified
Priority	Medium
Frequency of Use	Medium
Flow of events	1. Admin views the moderators profiles 2. Admin selects moderator to be modified 3. System updates moderator access

Alternative Flows	**Role Removal Failure:** If there is an issue removing or modifying a moderator's role, the system notifies the admin and does not proceed with the change.
Exceptions	**Admin Permissions:** If an admin does not have the required permissions to modify moderator roles, the system should deny the action and provide an error message.
Includes	**Audit Logging:** Changes made to moderator roles should be logged with the admin's information, the action taken, and the timestamp for security and accountability.
Special Requirement	**Requires authorised permission:** Only authorised administrators can make adjustments to the moderator roles.
Assumptions	Admins must be familiar with the role and responsibilities of modifying the moderators' permissions and roles.
Notes/Issues	None

1.1.18

Use Case ID	#1-18
Case	Manage Reports and Removes Post
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Administrator
Description	Admins can overview moderation decisions and handle appeals for reported posts/users
Preconditions	1. Admin is logged in and authorised 2. Reported post/user exists
Postconditions	1. The system updates the post/user modifications
Priority	High
Frequency of Use	Medium
Flow of events	1. Admin views reported posts/users 2. Admins oversees moderator decisions 3. Admin confirms or overrides decisions
Alternative Flows	**Moderation Decision Review Failure:** If the moderator's decision cannot be reviewed due to missing evidence or technical issues, the system should notify the admin and provide options to request additional information.
Exceptions	**Admin Permissions:** If an admin lacks the necessary permissions to override moderator decisions or handle

	appeals, the system will deny the action and display an appropriate error message.
Includes	**Notifications:** The outcome of whether the user's post is removed or not will be sent to the user through an in-app notification.
Special Requirement	**Compliance with Regulations:** The process must comply with relevant regulations, including GDPR or other legal requirements for handling user-generated content and personal data.
Assumptions	Admins can fairly assess the reports.
Notes/Issues	None

1.1.19

Use Case ID	#1-19
Case	Manage Maintenance
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Administrator
Description	Admins controls relevant system configurations (e.g API integration)
Preconditions	1. Admin is logged in and authorised
Postconditions	1. The system settings are updated
Priority	High
Frequency of Use	Medium
Flow of events	1. Admin access the settings 2. Admin modifies the relevant settings 3. The system validates input and applies changes 4. Users are notified if system is put under maintenance
Alternative Flows	**Invalid Configuration:** If the admin enters invalid settings or a configuration conflict occurs, the system will prompt the admin to correct the issue before applying any changes.
Exceptions	**Admin Permissions:** If an admin does not have the proper permissions to modify certain system settings ,an error message will be displayed.
Includes	**Notifications:** Users should be notified when there is a scheduled maintenance that will affect their access when the system is down.

Special Requirement	**Compliance with Regulations:** Any changes should comply with relevant legal and security requirements, such as data protection regulations.
Assumptions	Admins are trained and familiar with the system's maintenance procedures, including the potential impact of changes to the system.
Notes/Issues	Scheduled maintenance should be done during off-peaks hours.

1.1.20

Use Case ID	#1-20
Case	Sending Notifications and Alerts
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Administrator
Description	Admins can send out system-wide notifications and critical alerts to users
Preconditions	1. Admin is logged in and authorised 2. The system must be online and connected to the database 3. The alert/notification must be valid
Postconditions	1. The notification is successfully sent out to users 2. The systems stores the notification for tracking 3. Users receive the notification
Priority	High
Frequency of Use	Medium
Flow of events	1. Admin submits the notification details 2. The system validates the notification 3. Notification is sent to users via relevant channels 4. System stores the notification 5. User receives the notification
Alternative Flows	**Notification Delivery Failure:** If the notification cannot be delivered due to issues, the system logs the failure and notifies the admin with an error message.

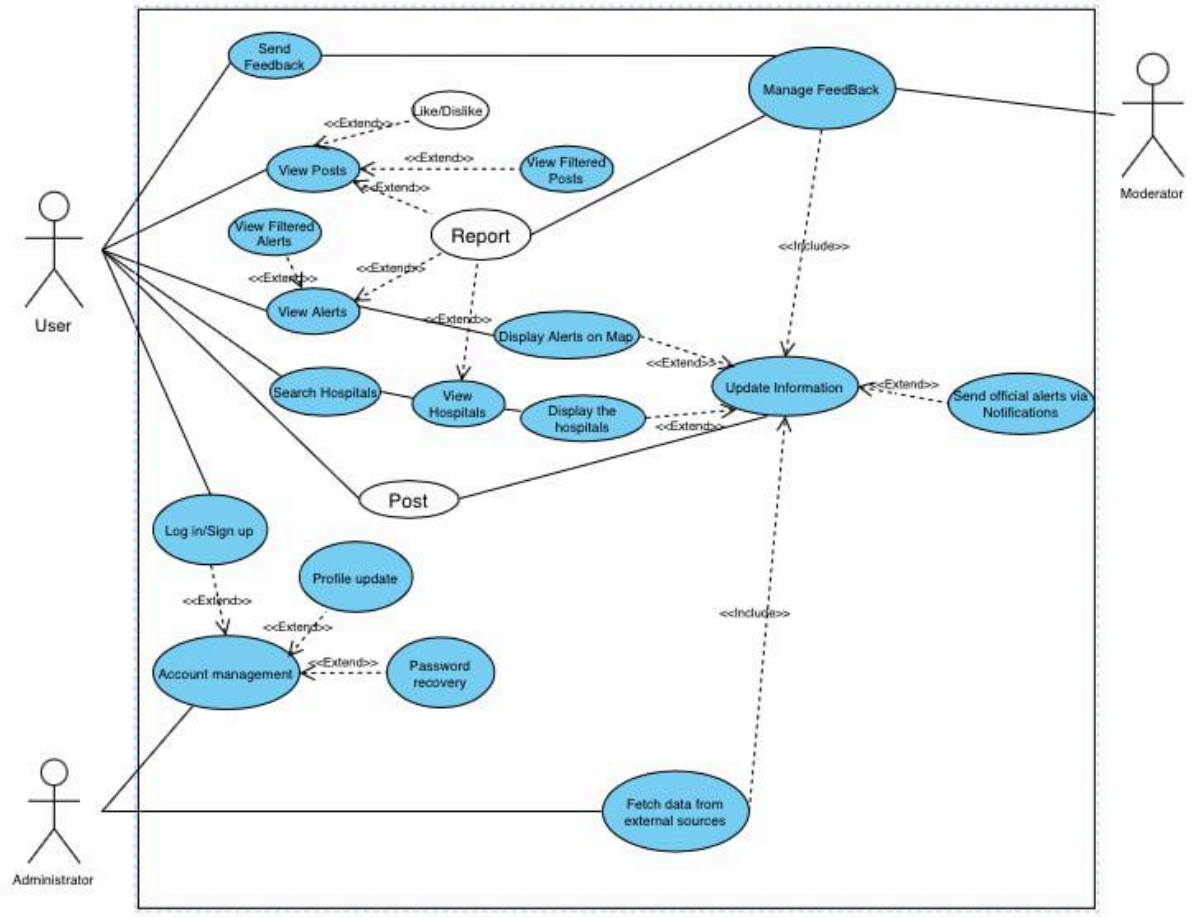
Exceptions	**System Offline**: If the system is offline or the database is not connected, the admin will be unable to submit or send the notification.
Includes	**System Logging and Tracking**: Every notification is tracked for auditing and reporting.
Special Requirement	**Security**: The system must ensure that only authorized admins can send notifications and that the notifications are sent securely.
Assumptions	Admins will always have access to the system and appropriate credentials.
Notes/Issues	Notifications should deliver any messages in a clear way.

1.1.21

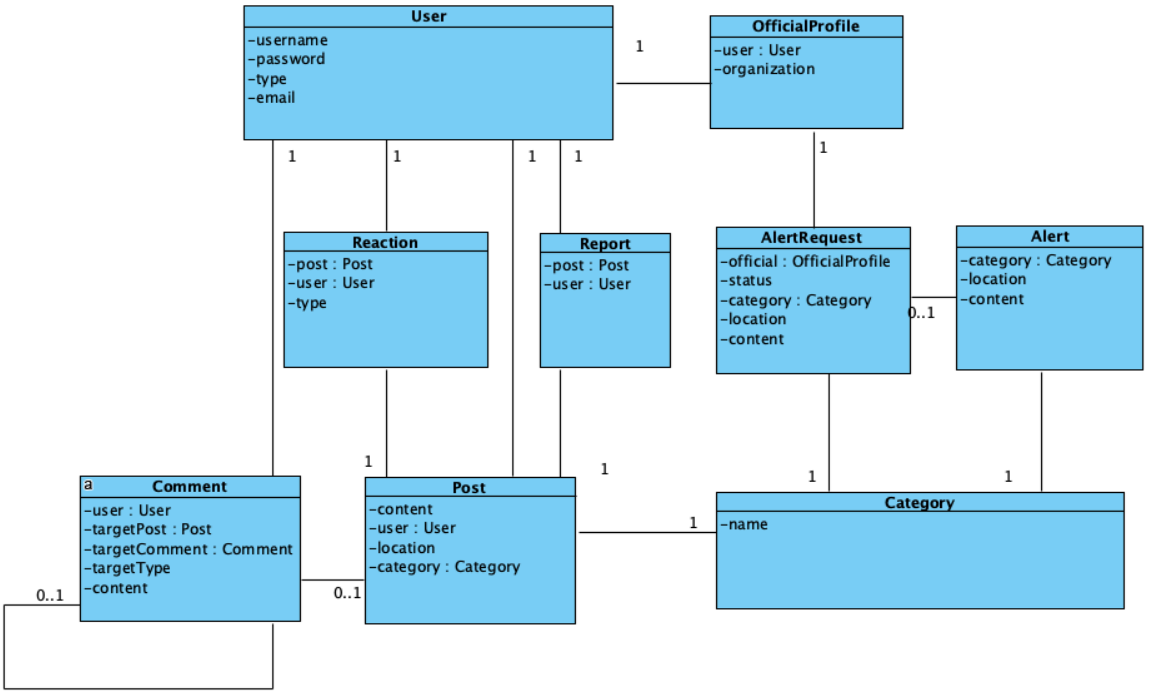
Use Case ID	#1-21
Case	Review and Manage API Integrations
Use Case History	**Date Created: 6 February** **Date Last Updated: 19 February**
Actor	Administrator
Description	Admins handles the external APIs for alerts, maps and hospitals
Preconditions	1. Admin is logged in and authorised 2. The system must be online and connected to external APIs 3. API keys must be valid and active
Postconditions	1. API configurations are updated and saved in the system 2. If new API keys are generated and validated 3. The system stores changes for tracking
Priority	High
Frequency of Use	Medium
Flow of events	1. System displays the relevant APIs 2. Admin selects an API and modifies the settings 3. The system saves the changes after validation
Alternative Flows	**Invalid API Key:** If an invalid API key is entered, the system will display an error message indicating the issue, and the admin will be prompted to provide a valid key.

Exceptions	**External API Downtime:** If the external API is down, the system will notify the admin and allow them to update settings once the connection is restored.
Includes	**API Key Management:** Admins can generate, revoke, and validate API keys for each external integration. This process is included within the flow of managing API integrations.
Special Requirement	None
Assumptions	Admins are familiar with the external APIs used in the system.
Notes/Issues	API must be up to date.

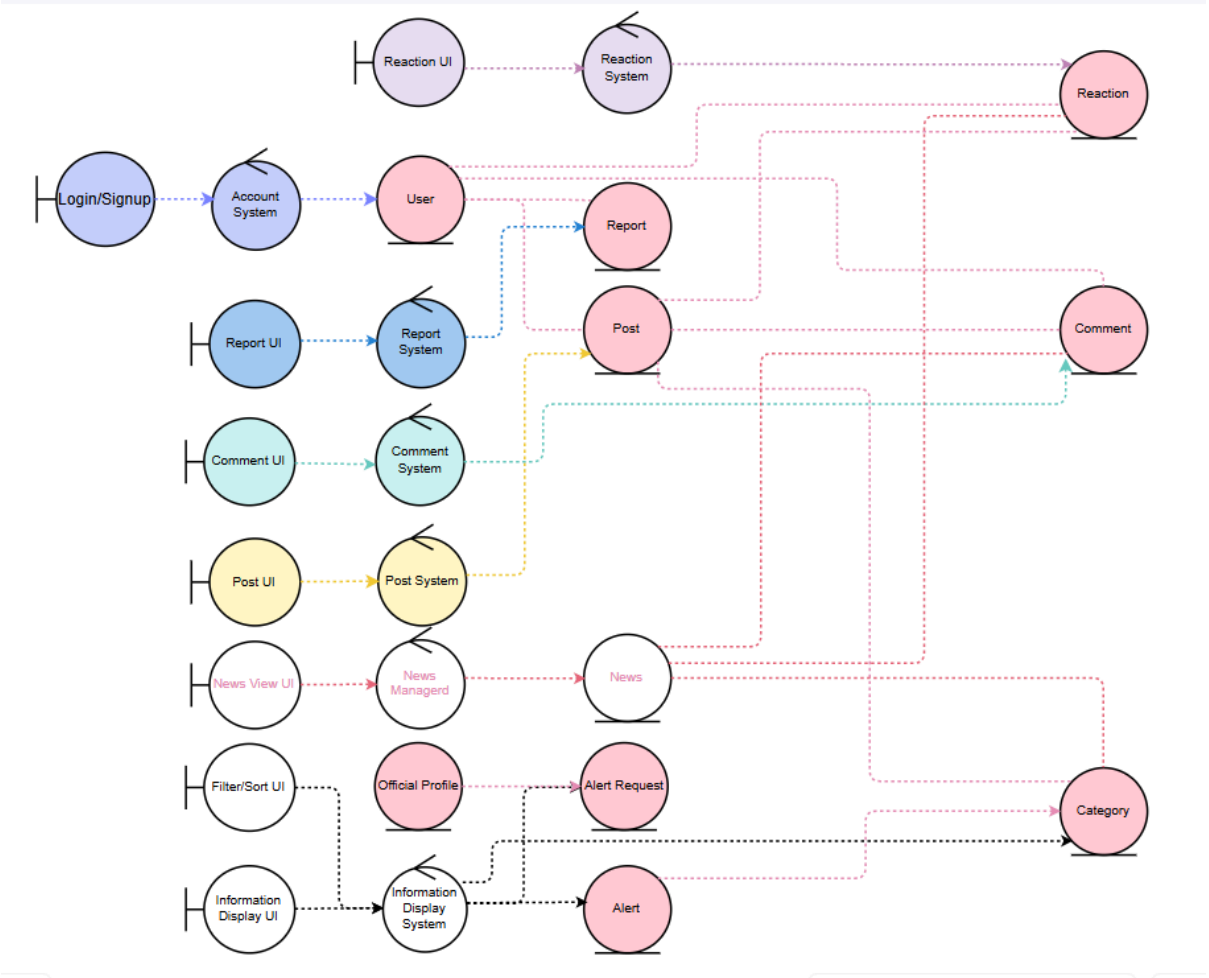
1.2 Use Case Diagram



2. Entity Class Diagram

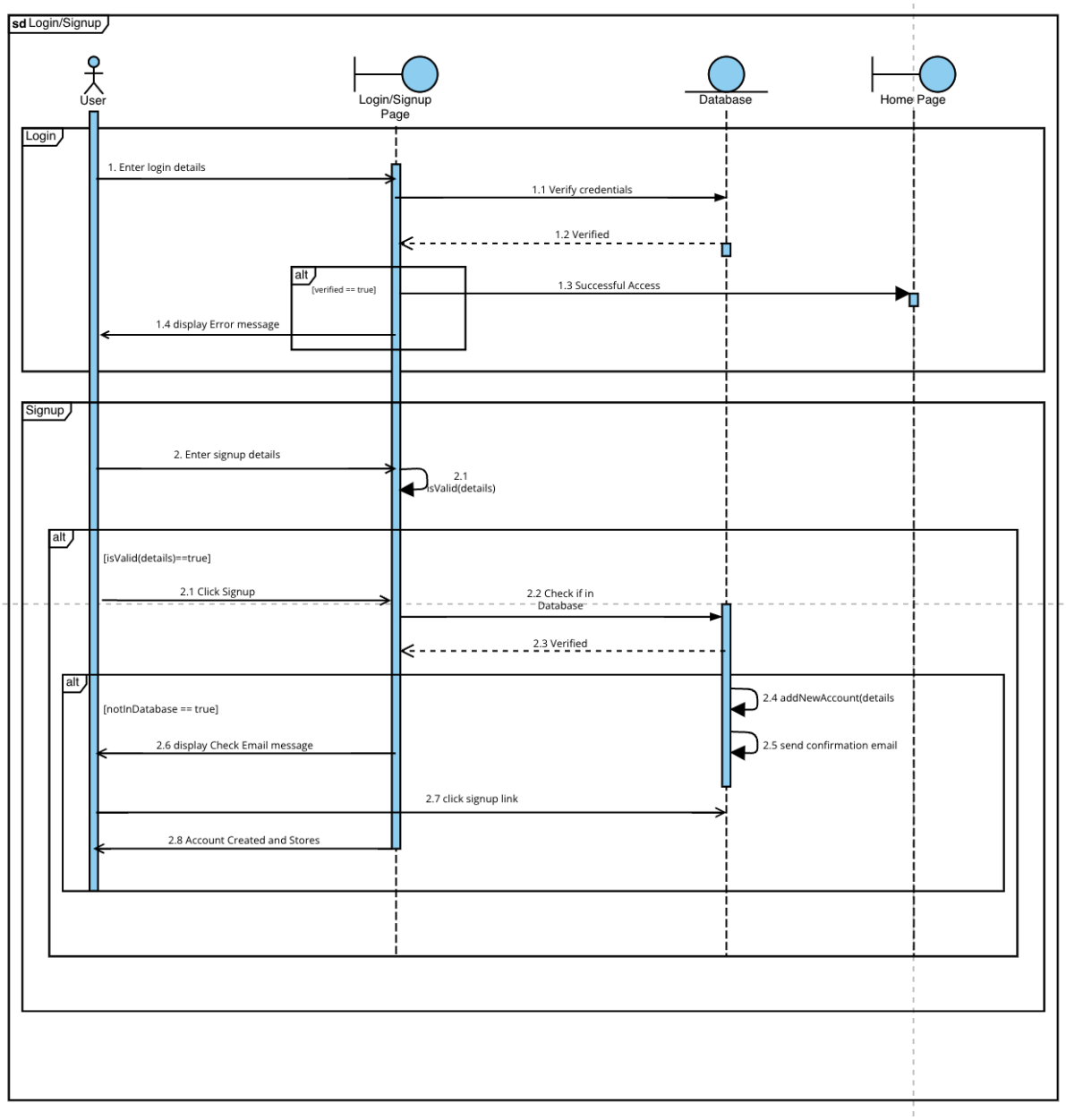


3. Bounded & Control Class Diagram

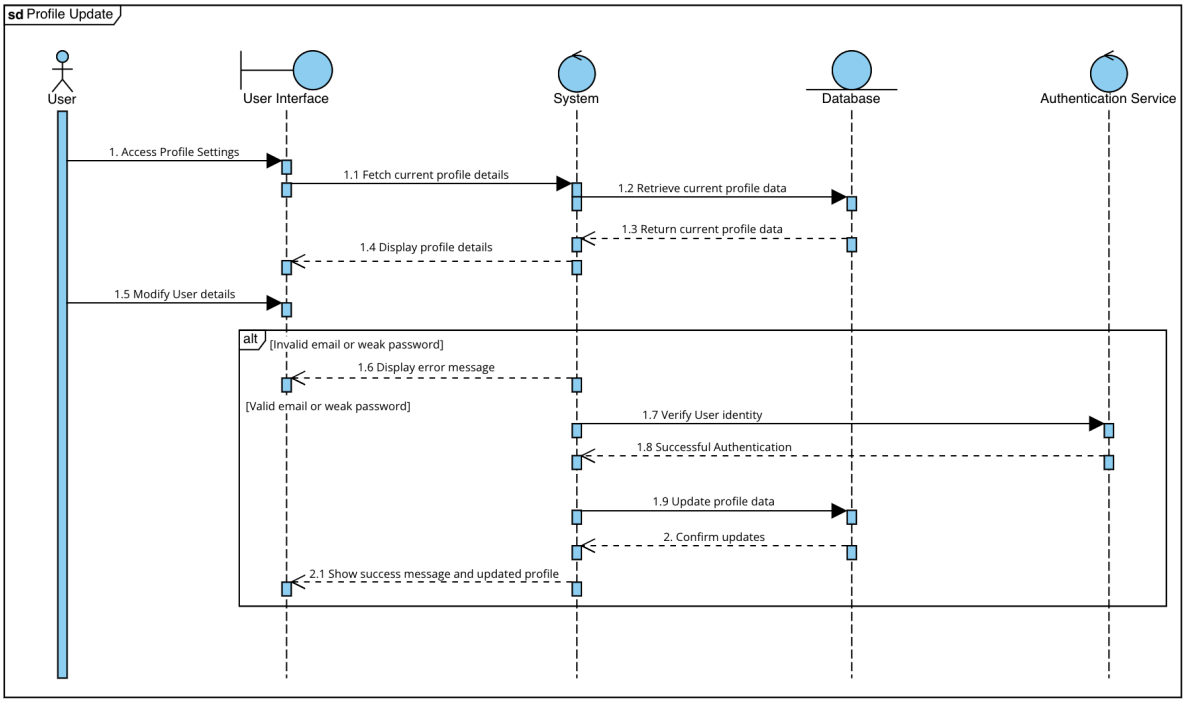


4. Sequence Diagrams

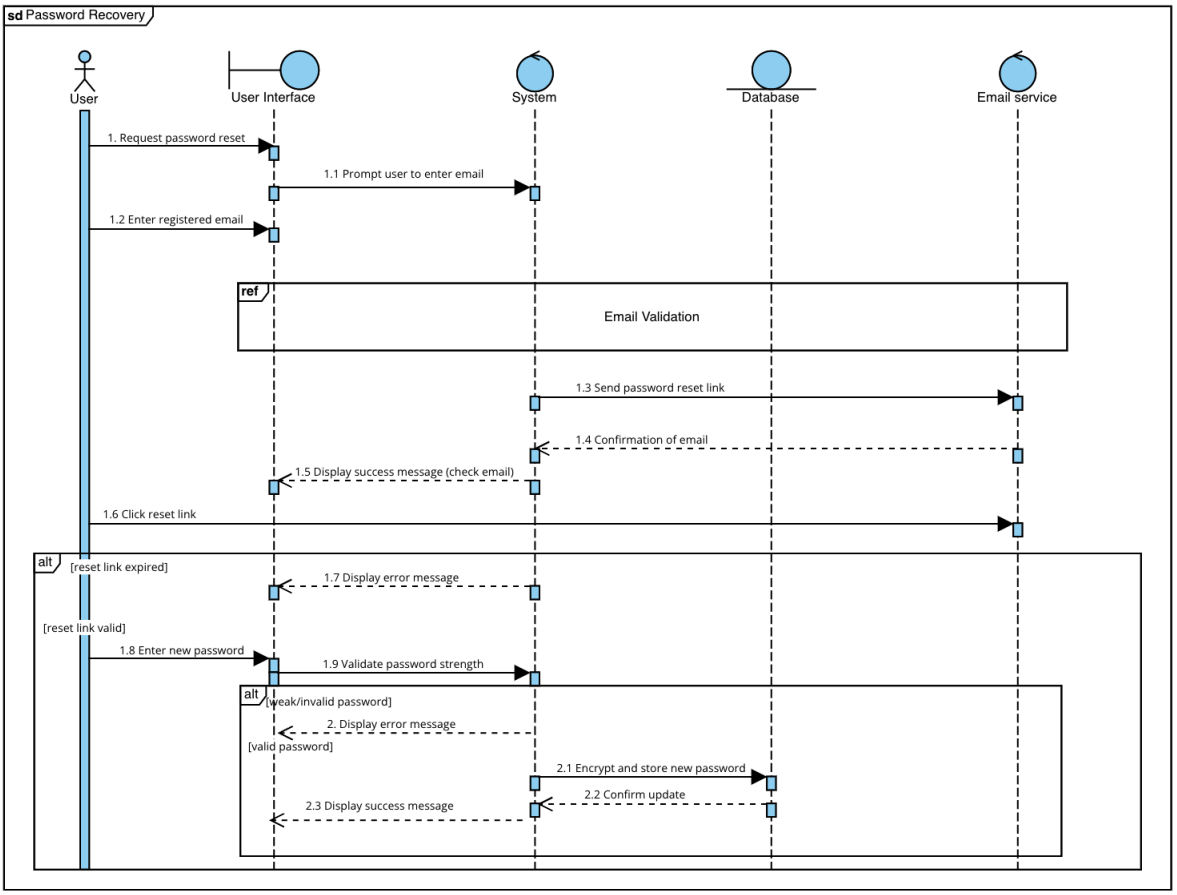
User Case 1



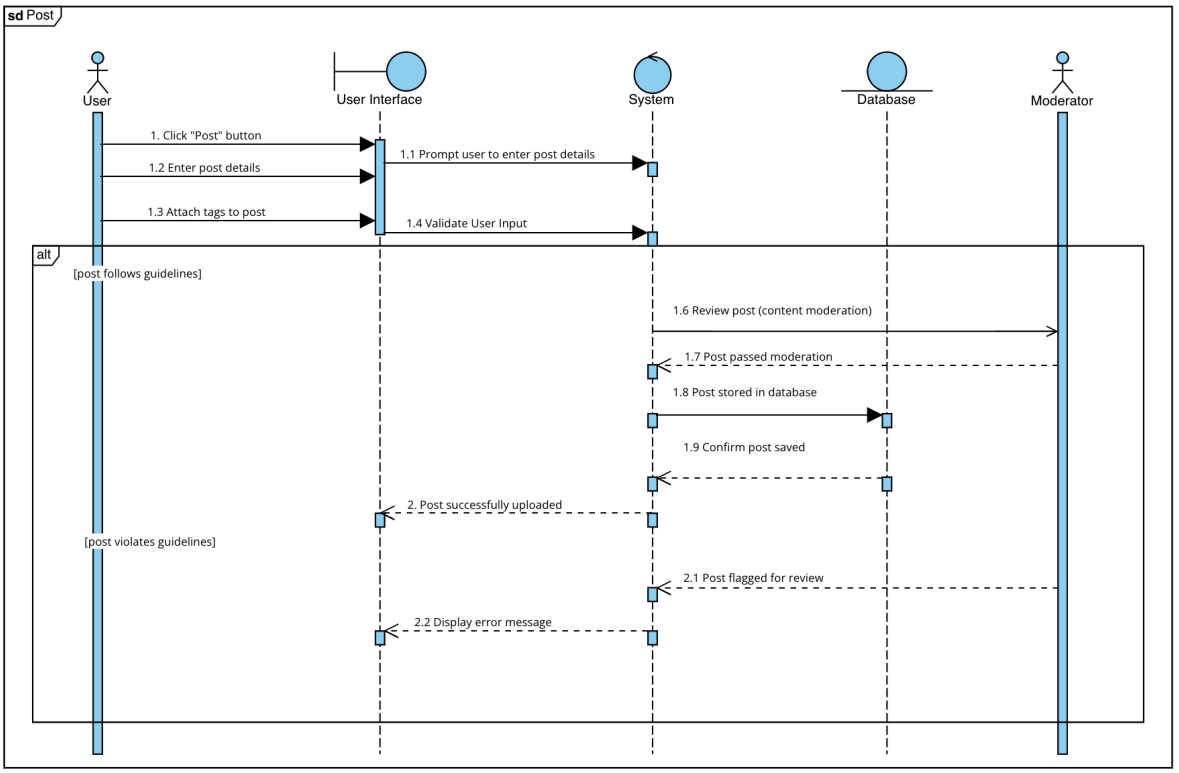
User Case 2



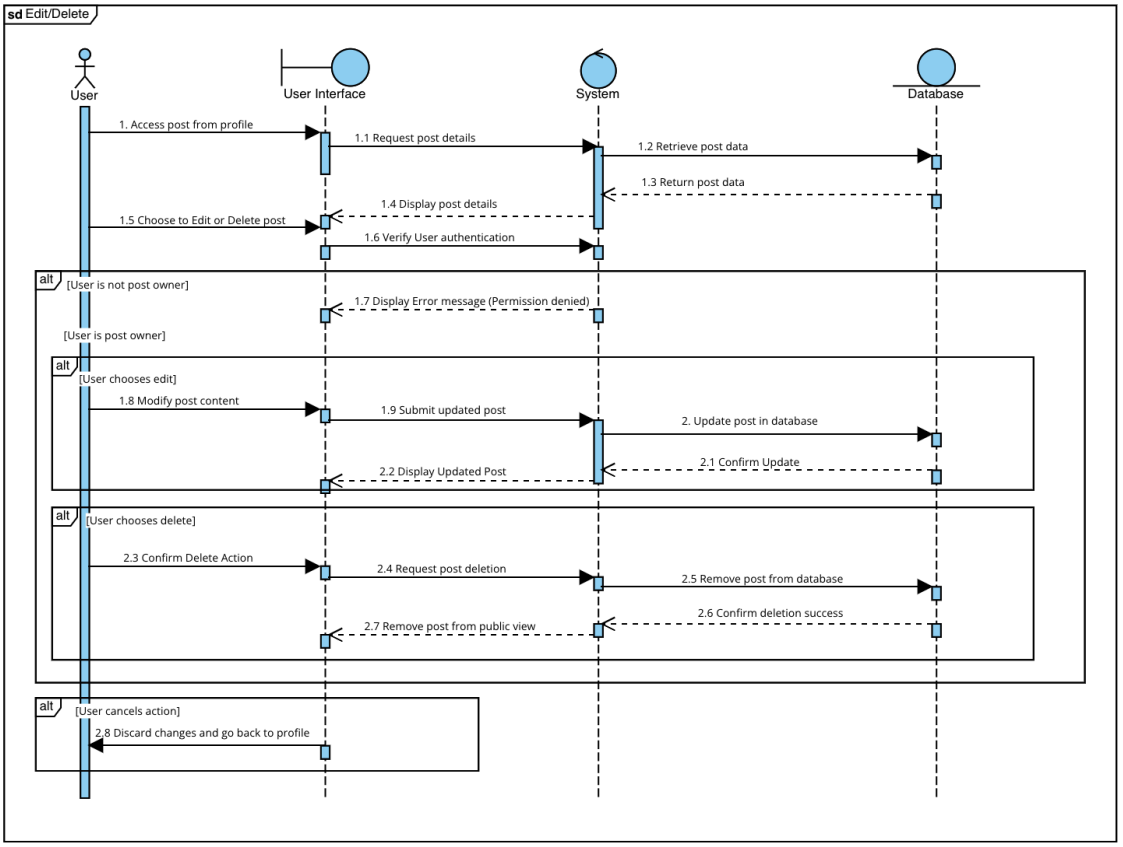
User Case 3



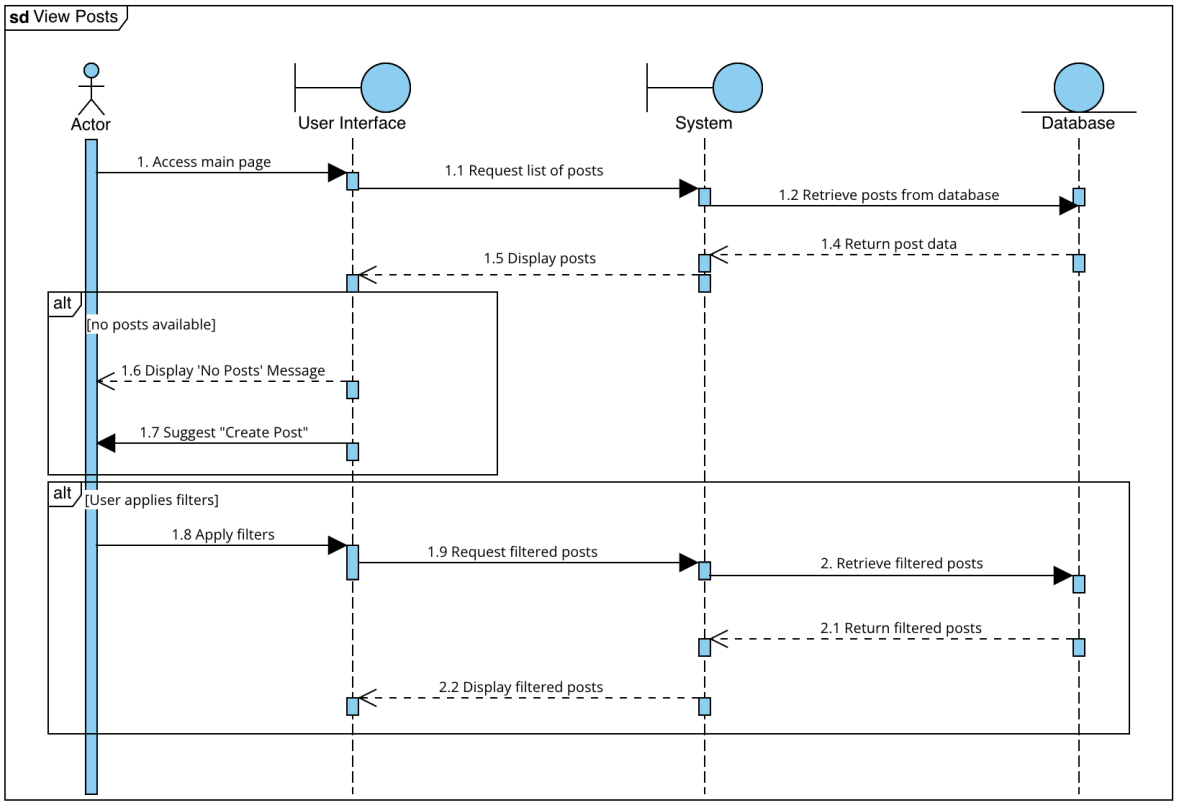
User Case 4



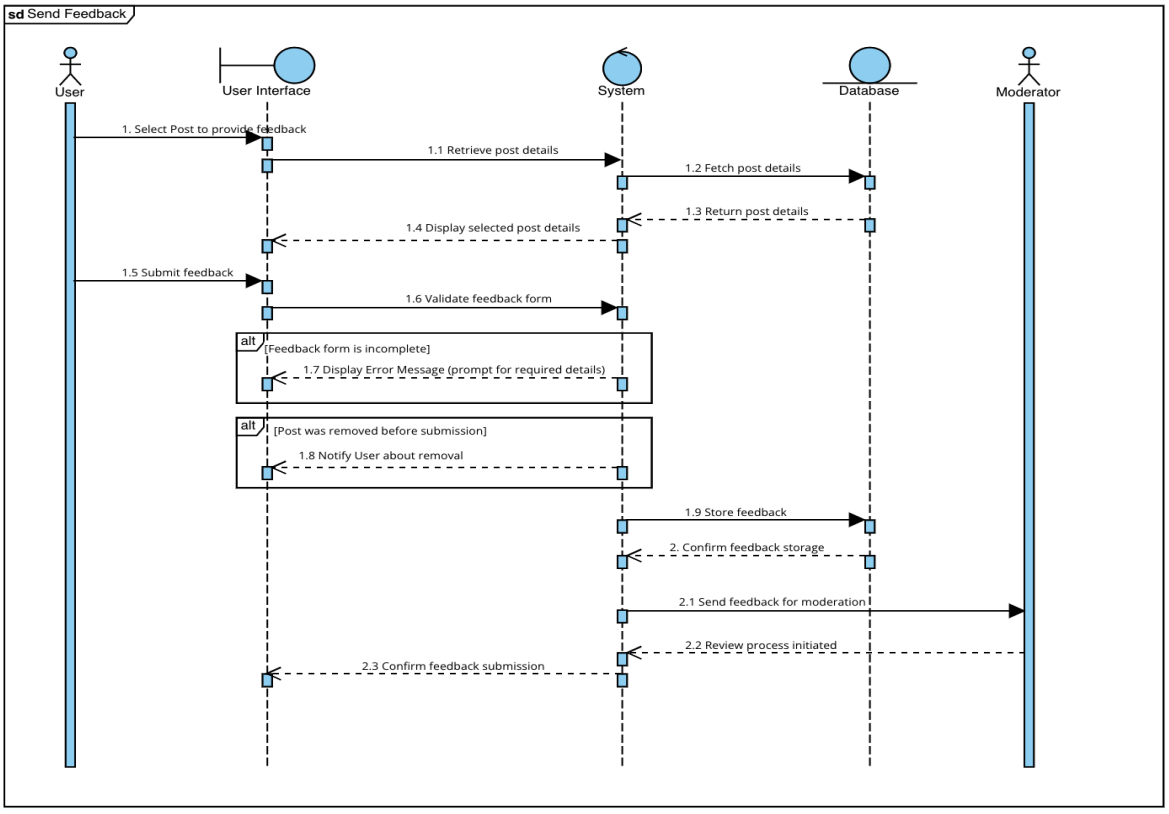
User Case 5



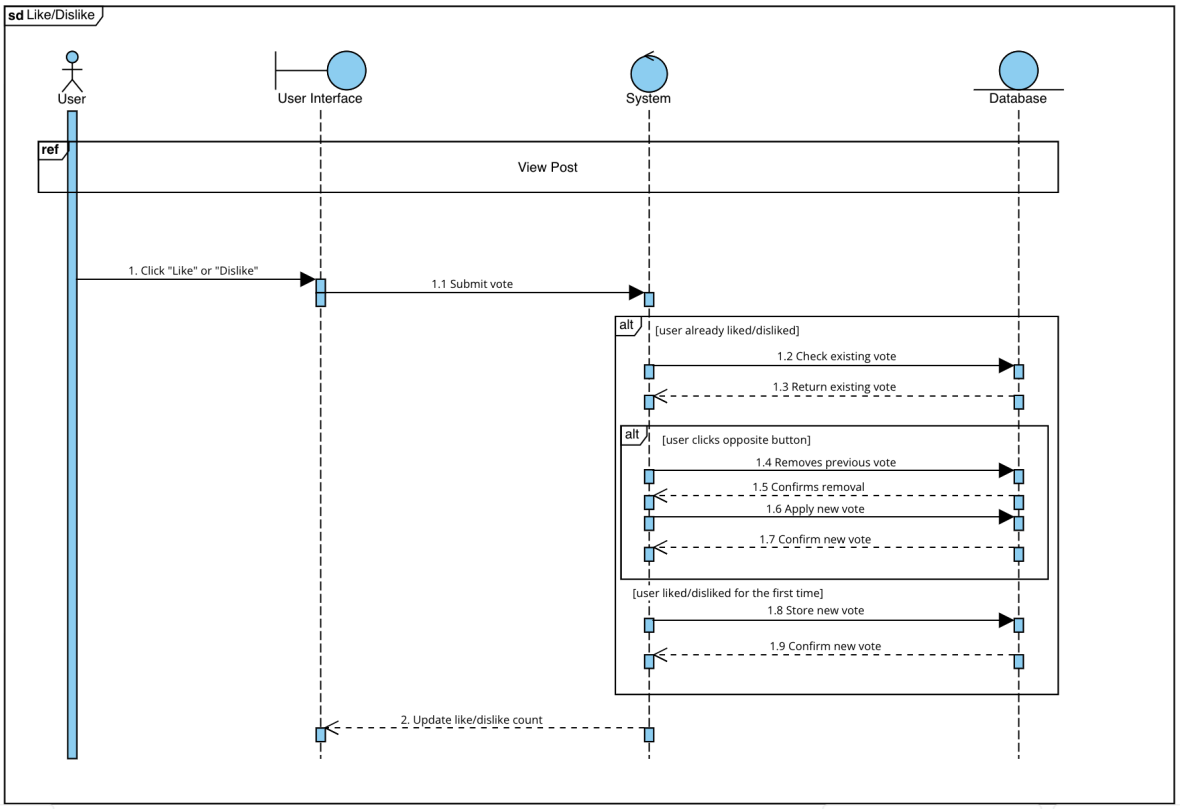
User Case 6



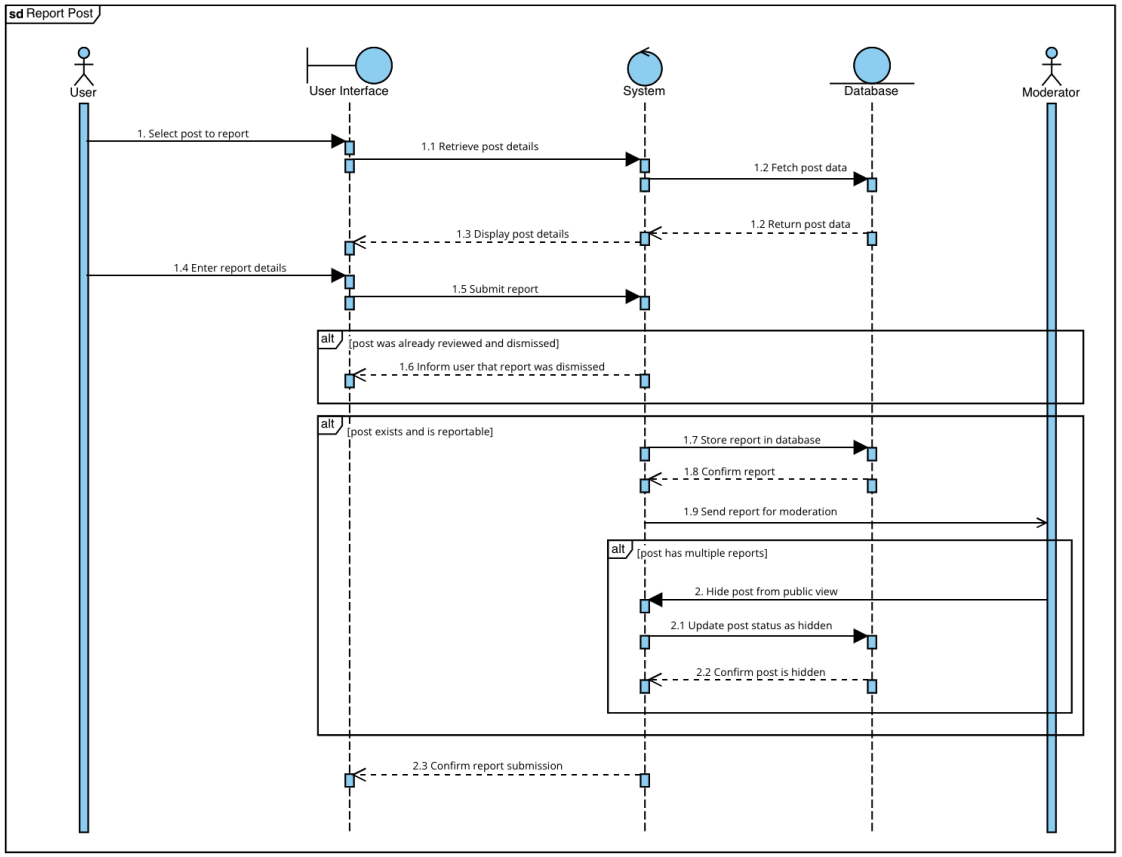
User Case 7



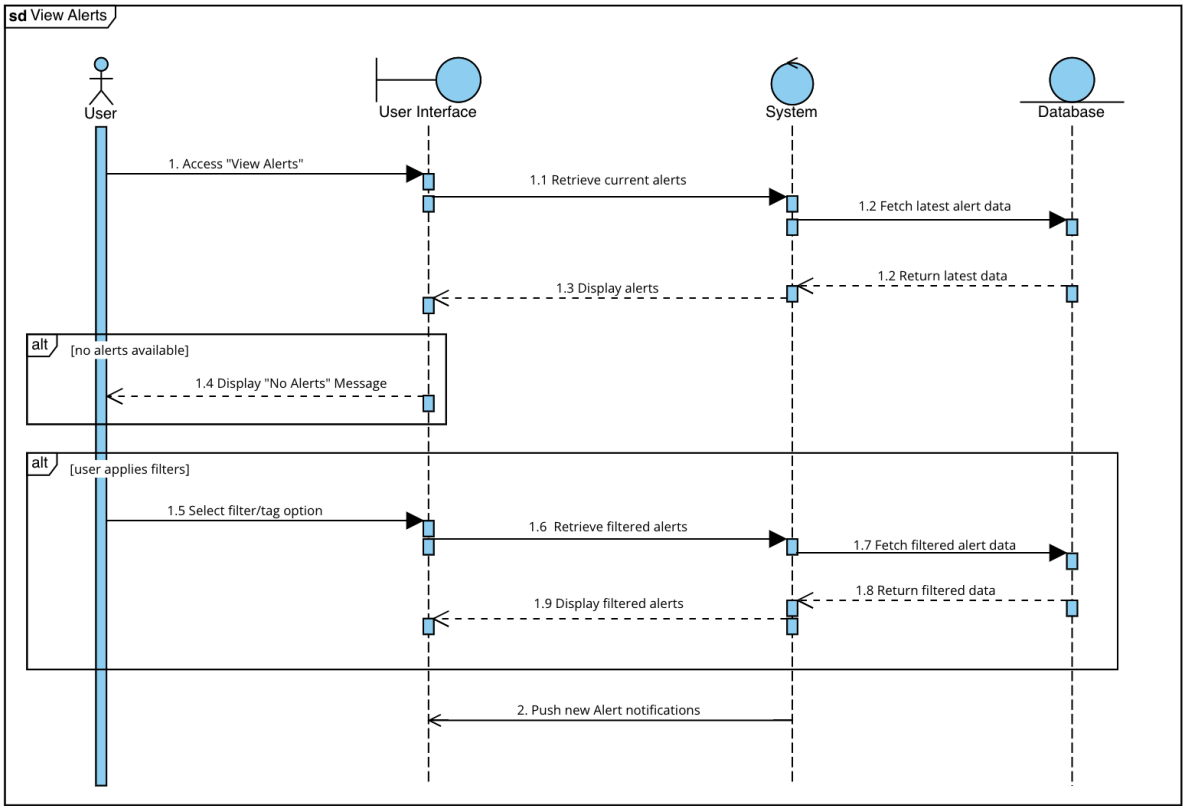
User Case 8



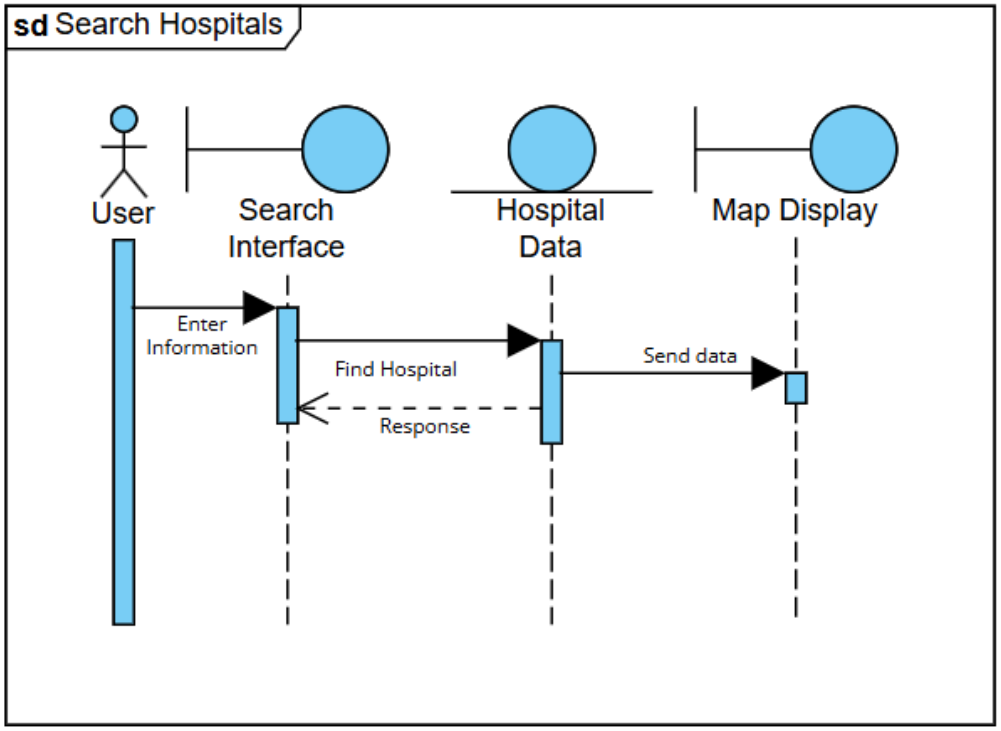
User Case 9



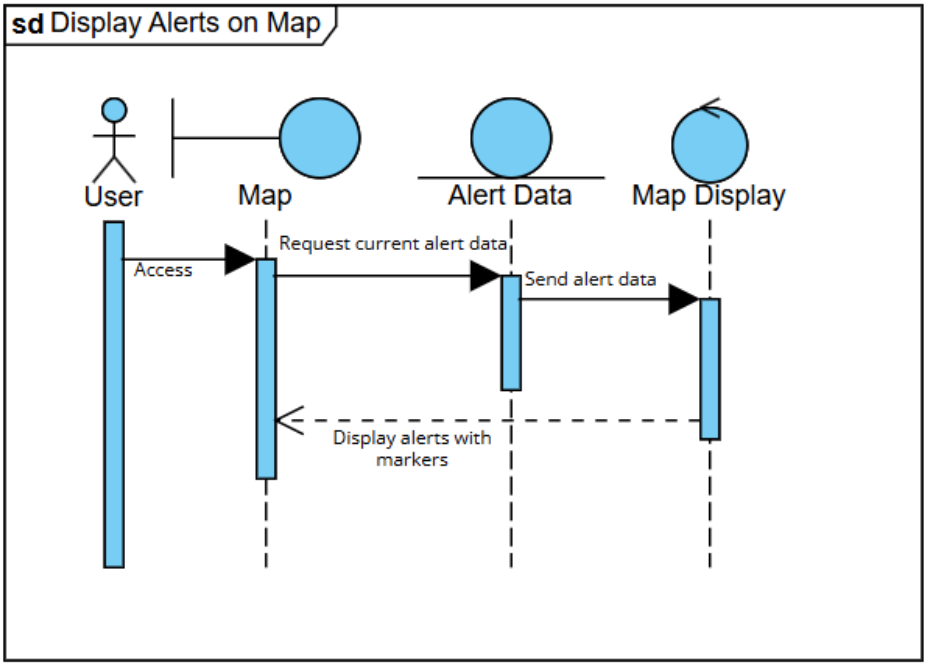
User Case 10



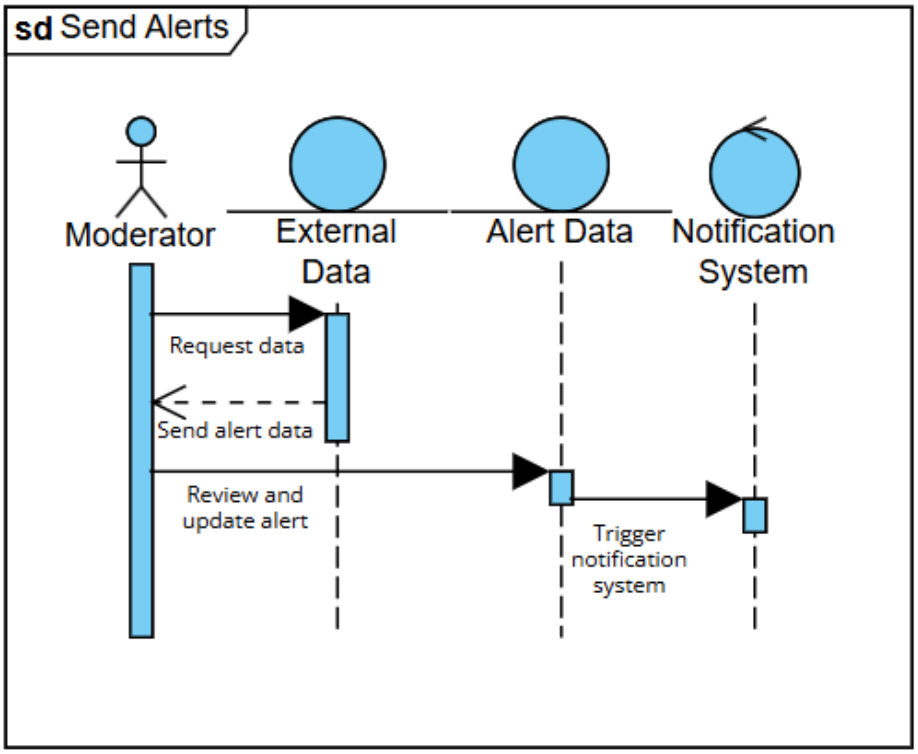
User Case 11



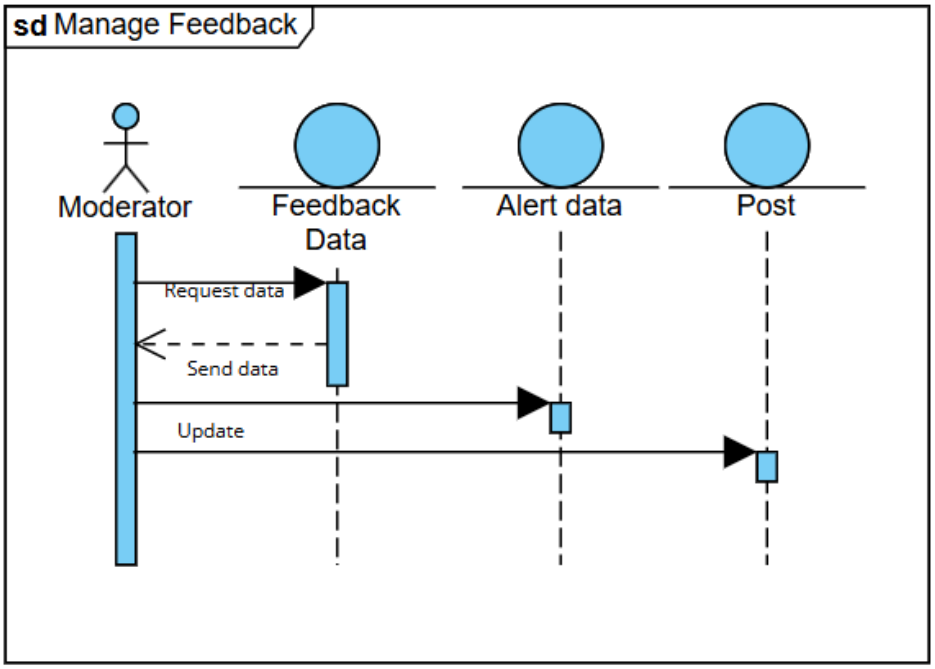
User Case 12



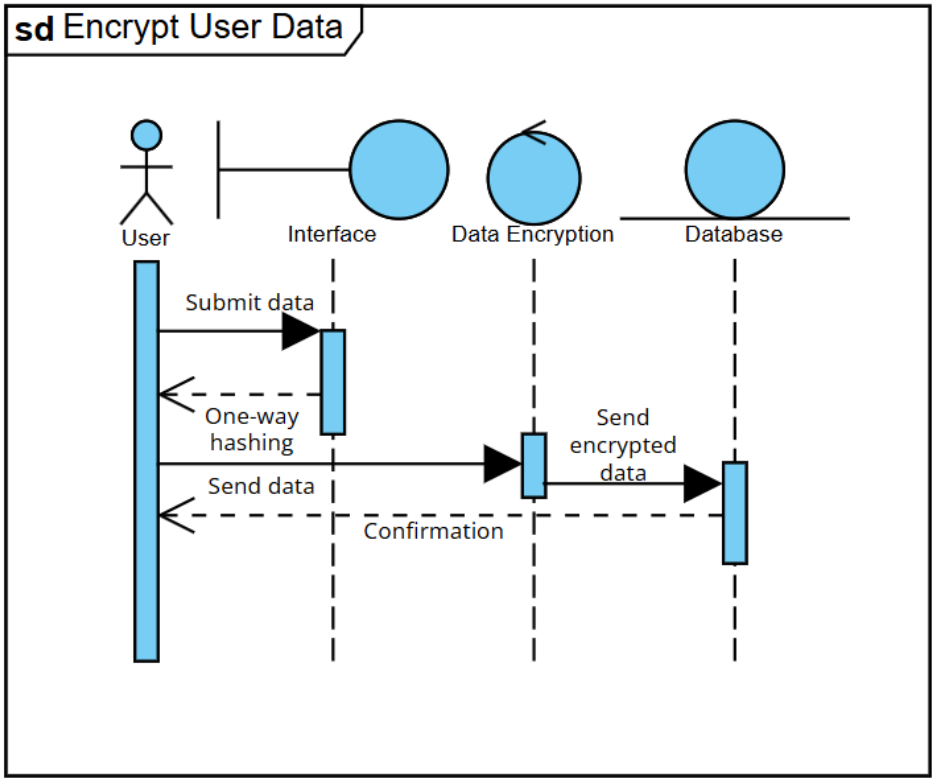
User Case 13



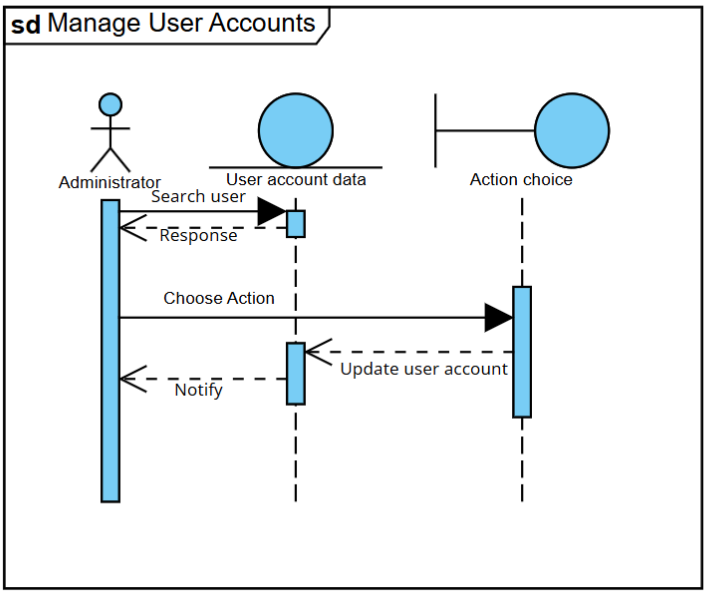
User Case 14



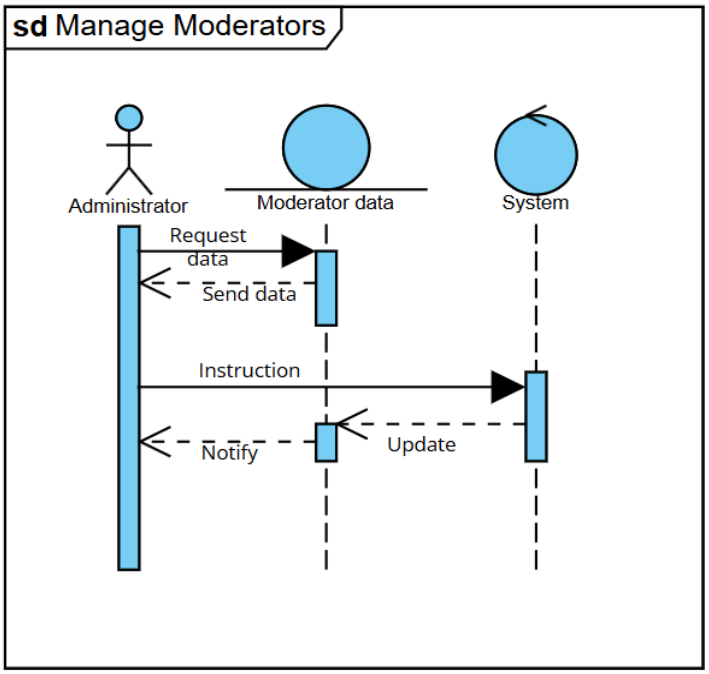
User Case 15



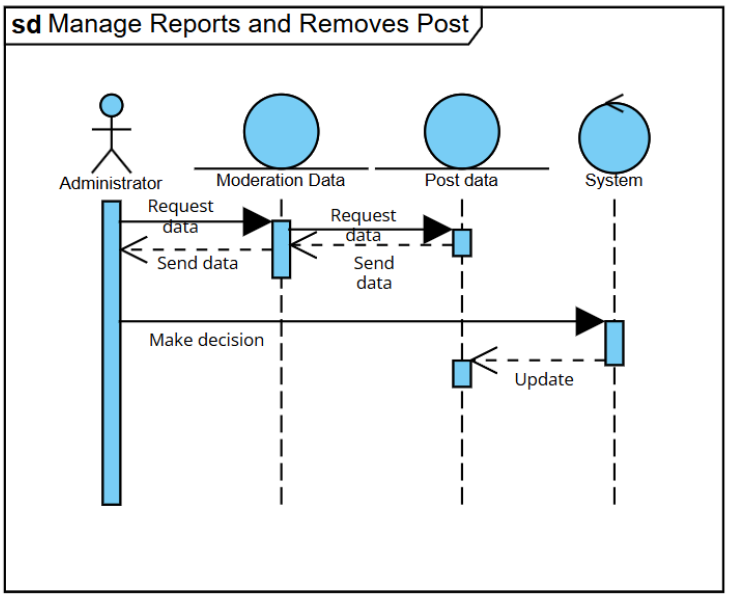
User Case 16



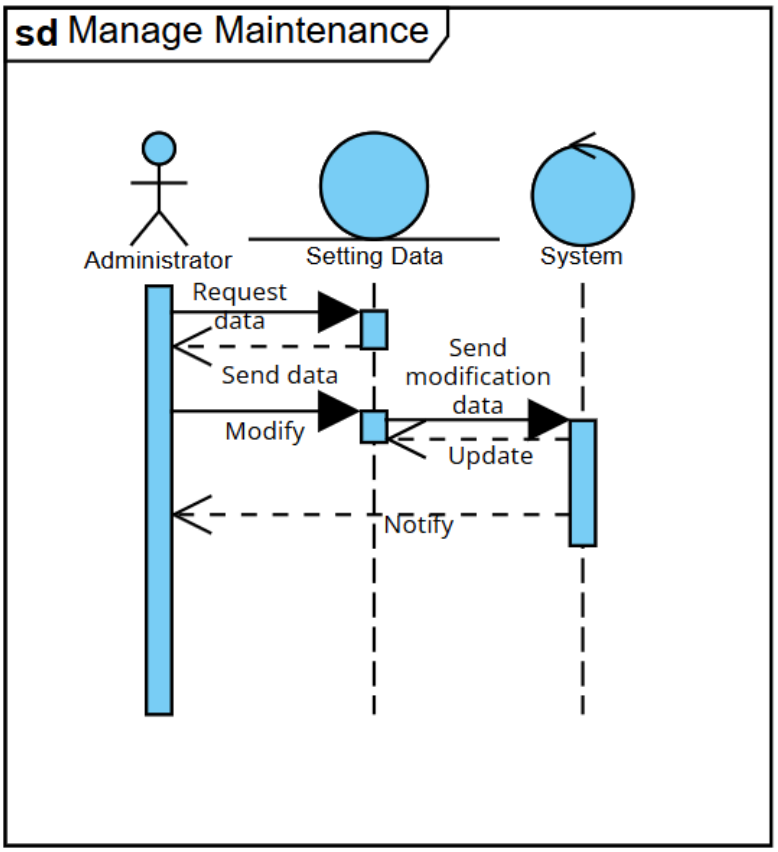
User Case 17



User Case 18



User Case 19



5. Initial Dialog Map