3.4 System Architecture

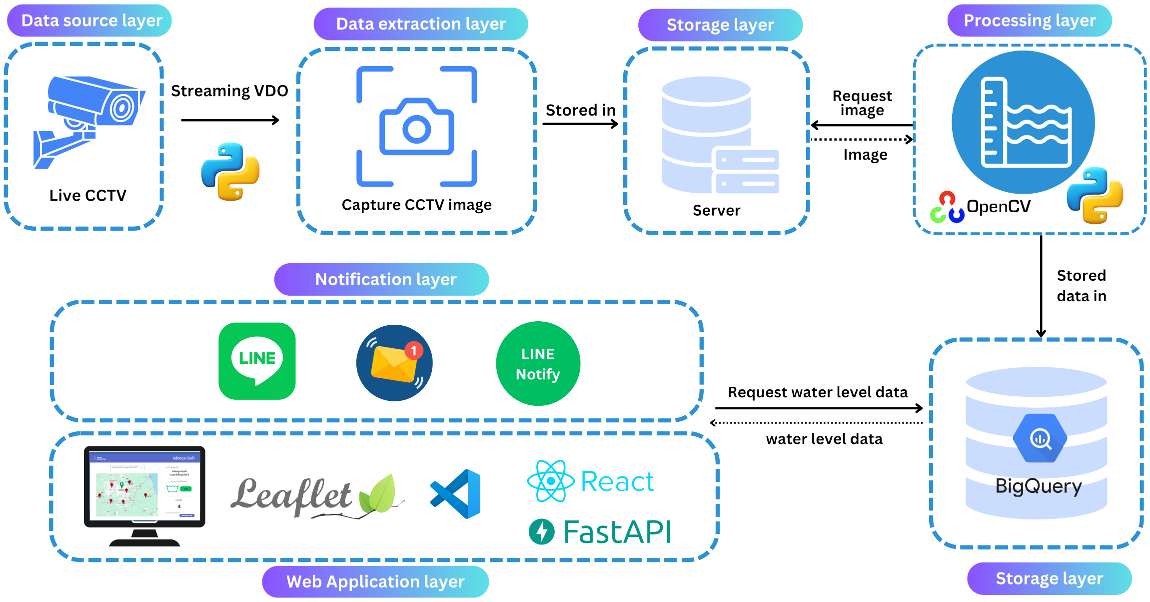

Figure 3.1 System architecture diagram

In this system architecture diagram as shown in Figure 3.1, it comprises 6 layers as follow; data source layer, data extraction layer, storage layer, processing layer, notification layer and web application layer.

- Data source layer - Data is collected from live surveillance cameras accessible on a public website.
- Data extraction layer - The system retrieves the live video feed, periodically capturing images at two-minutes intervals.
- Storage layer - Captured images are stored on depa's remote server within the storage layer. Processed water level data is subsequently stored in the BigQuery database on Google Cloud Platform.
- Processing layer - OpenCV is utilized for image processing techniques in the processing layer, enabling the detection of water levels from the stored images on depa's remote server.
- Notification layer - The system retrieves water level data from the BigQuery database and sends notification messages to users via Line Notify in the event of a zone change.
- Web Application layer - The system retrieves water level data from the database and subsequently the processed data is presented to users through the website interface created using React as the front-end and FastAPI as the back-end.

3.4.1 Data source and Data extraction layers

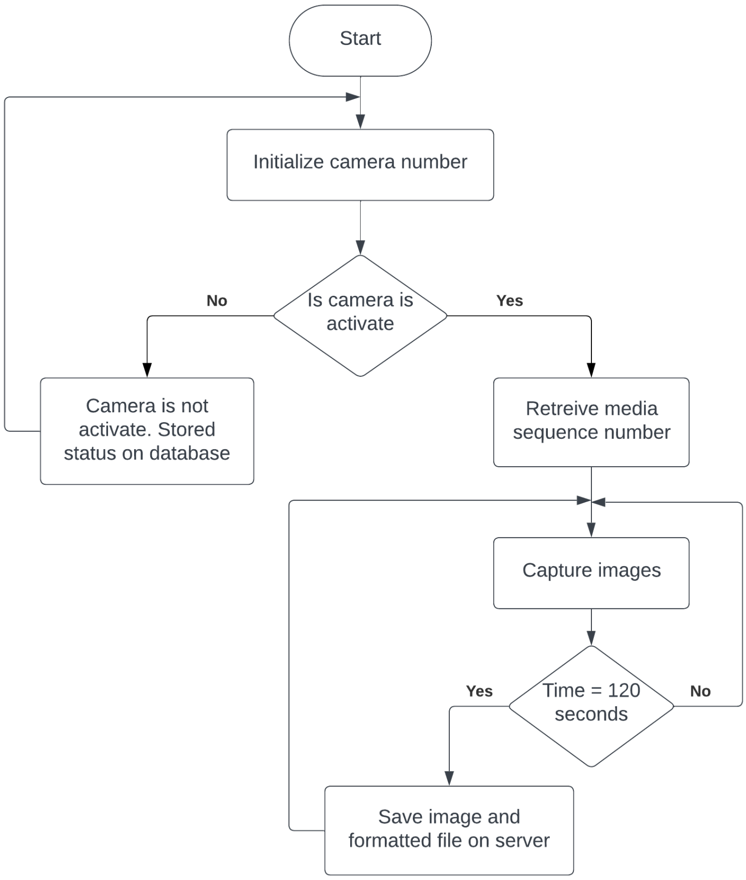

Figure 3.2 Flowchart of the image capture process

The data collection process initiates by selecting 3 specific camera locations for monitoring water levels as Khlong Tha Yai, Khlong Liap Thang Rot Fai, and Khlong Na Muang. The Flowchart of the image capture process as shown in Figure 3.2, the data source is from surveillance cameras, specifically the CCTV video feeds retrieved from a publicly accessible website that broadcasts live CCTV streams in Nakhon Si Thammarat, available at https://noc.nakhoncity.org/live-cctv. The camera number is identified from this website, and the media sequence number of the camera is retrieved. Subsequently, a Python script is created to automatically capture images every 2 minutes, which is deployed on the depa's server. The captured image files are formatted to include the camera name and date-time, and then stored on the server.

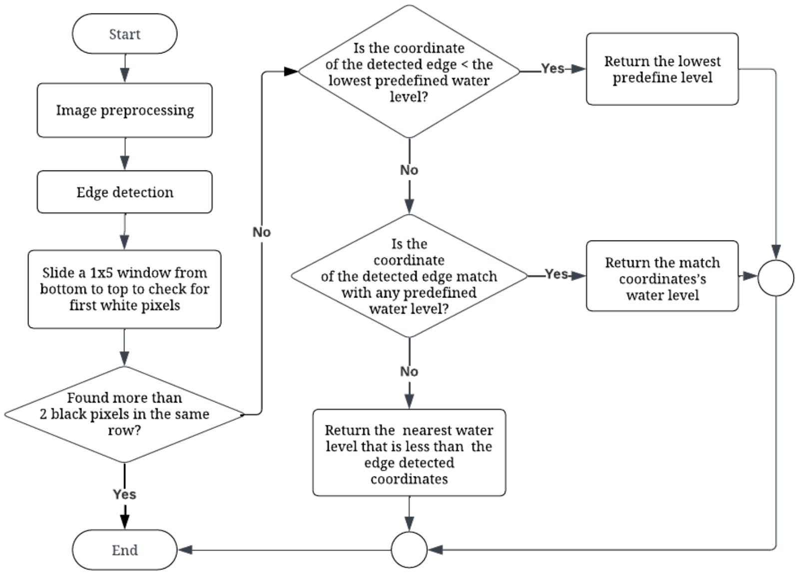

Figure 3.23 Water level estimation algorithm flowchart

If more than 2 black pixels are found, indicated that the edge is not strong enough to be the waterline, the process would end and the previous water level data would be use instead. Otherwise, the process would further check if the coordinates of the detected edge is less than the lowest predefined water level, if yes, indicated that the water is even below the lowest level, the lowest predefined water level would be used. Otherwise, the algorithm will continue to check if the coordinate of the detected edge is match with any predefined water level. If yes, the matched coordinate's water level would be used, otherwise the algorithm would return the nearest water level that is less than the edge detected coordinates.

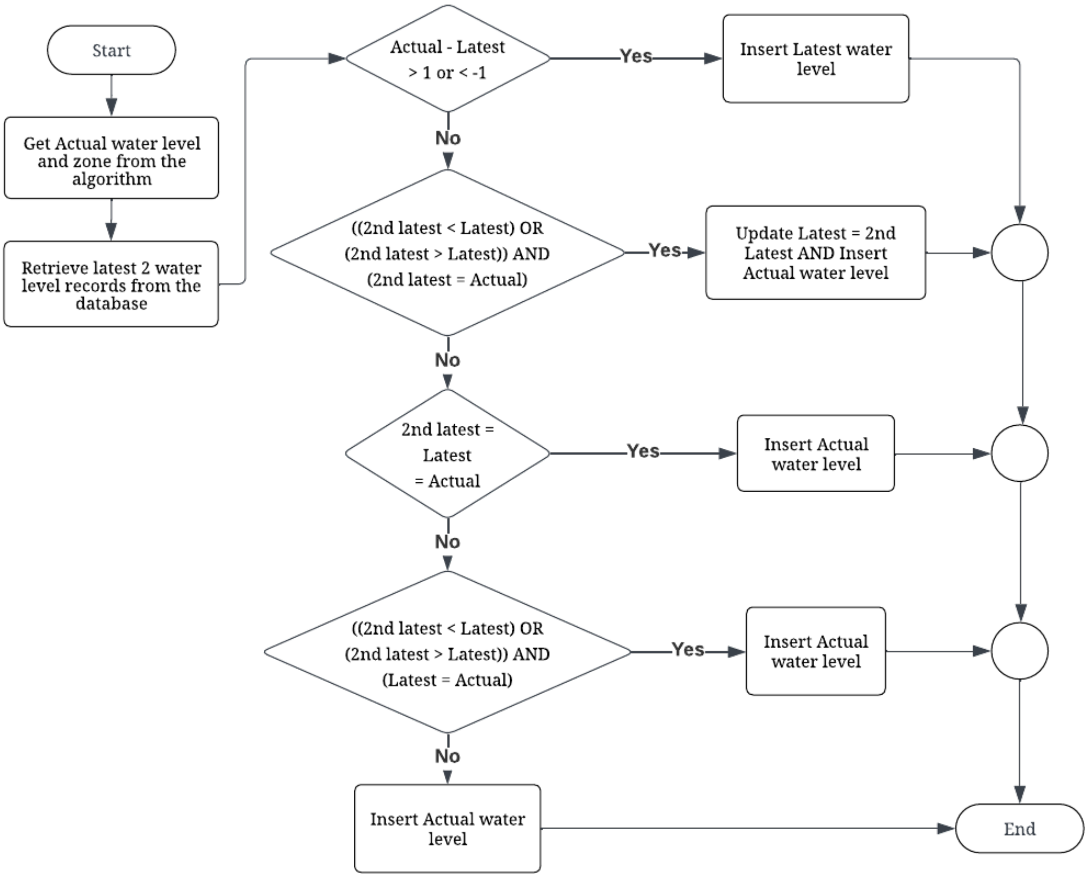

Figure 3.24 Water level data insertion criteria

To enhance the accuracy of the water level, we add the criteria for inserting result data to prevent inaccuracies. This involved evaluating differences between the actual and latest water levels, as well as comparing the actual, latest, and second latest water levels. Additionally, we implemented updating erroneous previous records to mitigate the impact of incorrect data, as shown in Figure 3.24.

From Figure 3.24, starting with getting the actual water level data which refers to the output water level obtained from the water level estimation algorithm, which has not yet been inserted into the database. Next, we retrieve the latest 2 water level records from the database to check against the 5 main conditions, as follows:

1. If the difference between the actual water level and the latest water level exceeds ±1 level, then replace the actual water level with the latest one and insert it to the database.
2. If the second latest level is lower or greater than the latest level, and the second latest level matches the actual water level, then update the latest water level in the database to match the second latest level and insert the actual water level to the database.
3. If the second latest, latest, and actual water levels are all identical, we insert the actual water level into the database.
4. If the second latest level is lower or greater than the latest level, and the latest level matches the actual water level, then insert the actual water level to the database.
5. If no condition is matched, then insert the actual water level into the database.

3.4.4 Notification layer

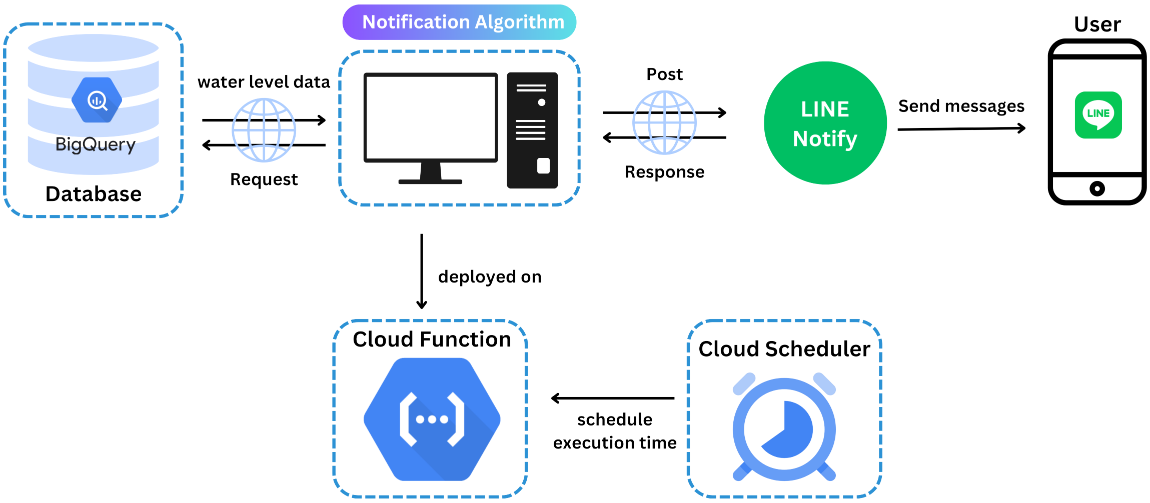

Figure 3.25 Notification system diagram

In the notification system diagram, as shown in Figure 3.25, the program initiates a request for water level data to the Google BigQuery database. The database then returns the resulting data in JSON format to the program.

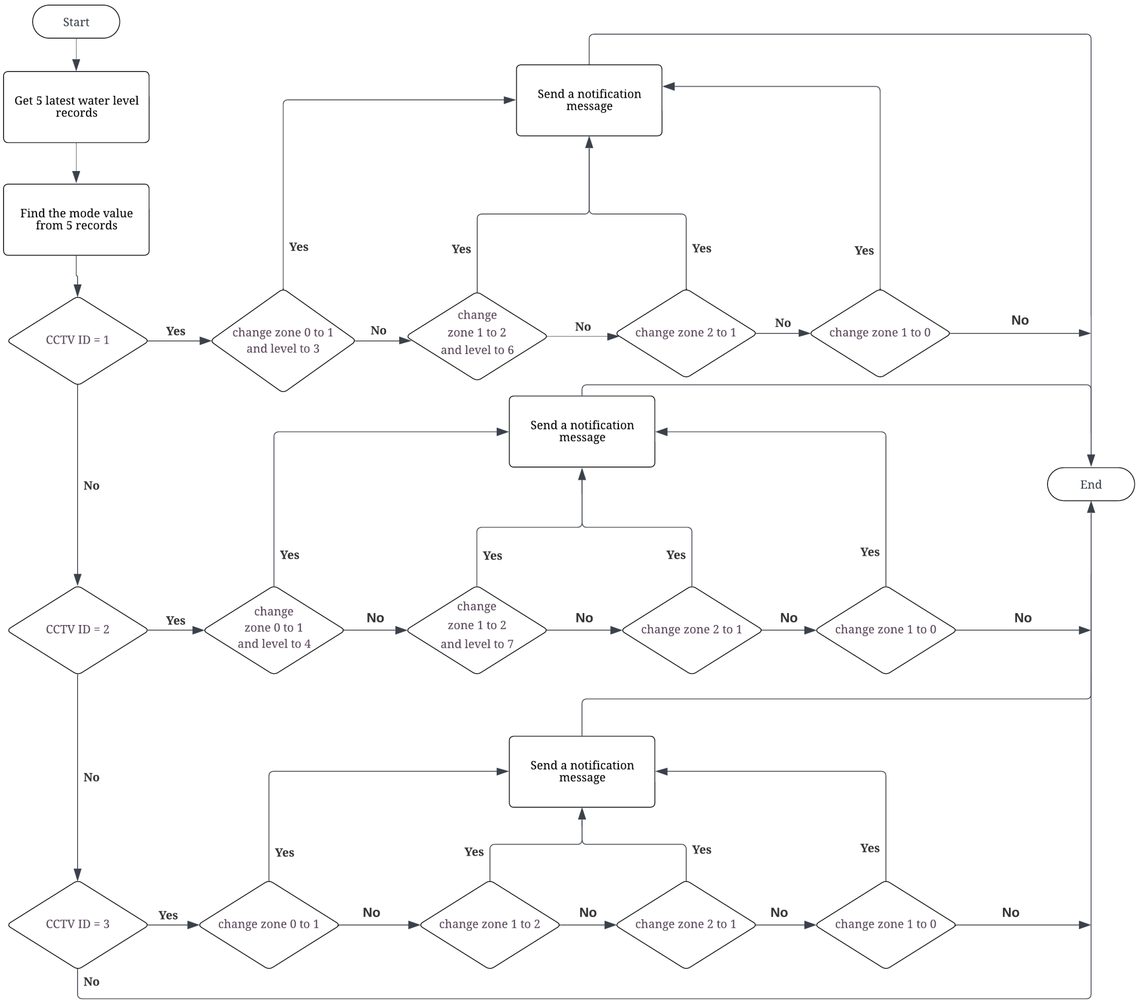

Figure 3.26 Notification algorithm flowchart

In the Notification Algorithm flowchart, as shown in Figure 3.26, the program's algorithm retrieves the 5 latest water level records from the database and calculates the mode value from these records in order to identify the most frequently recurring water level value. Subsequently, it checks the criteria for each camera, as the notification criteria vary for each camera. Zone 0 represents the normal zone, Zone 1 represents the warning zone, and Zone 2 represents the danger zone.

- For CCTV 1 or Khlong Tha Yai, notifications are sent in four cases: firstly, when the zone changes from 0 to 1 and the water level changes to 3; secondly, when the zone changes from 1 to 2 and the water level is changing to 6; thirdly, when the zone changes from 2 to 1; and finally, when the zone changes from 1 to 0.
- For CCTV 2 or Khlong Na Muang, notifications are sent in four cases: firstly, when the zone changes from 0 to 1 and the water level changes to 4; secondly, when the zone changes from 1 to 2 and the water level is changing to 7; thirdly, when the zone changes from 2 to 1; and finally, when the zone changes from 1 to 0.
- For CCTV 3 or Khlong Liap Thang Rot Fai, the criteria remains unchanged, as notifications are sent whenever a zone is changed.

3.4.5 Web Application layer

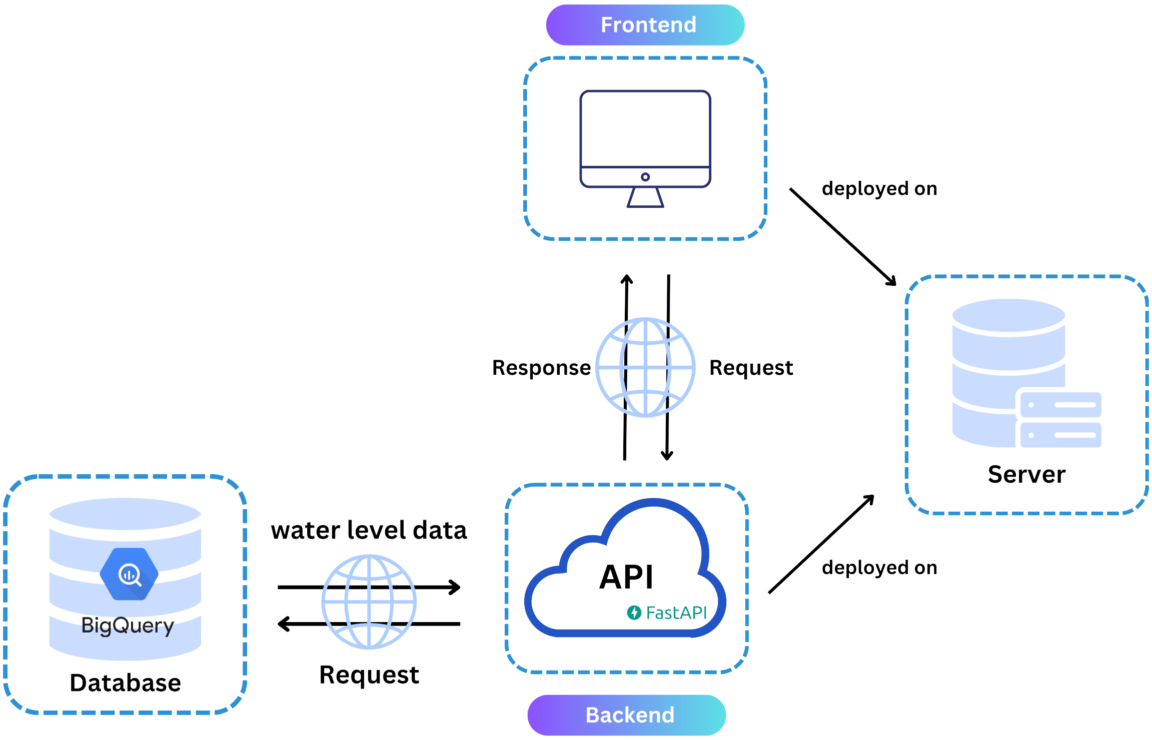

Figure 3.30 Web application diagram

In the web application diagram, as shown in Figure 3.30, users have the capability to access the website interface through a web browser. The front-end initiates a request for water level data from the API. The API sends an SQL query to the Google BigQuery database. Subsequently, the database sends the resulting data back to the API. The API furnishes the data in JSON format to the frontend. The frontend fetches data, enabling users to view the data on the website interface through their web browsers.

For deployment, both the frontend and APIs components are deployed on the depa's server.

3.5 Use Cases Diagram

3.5.1 Web Application

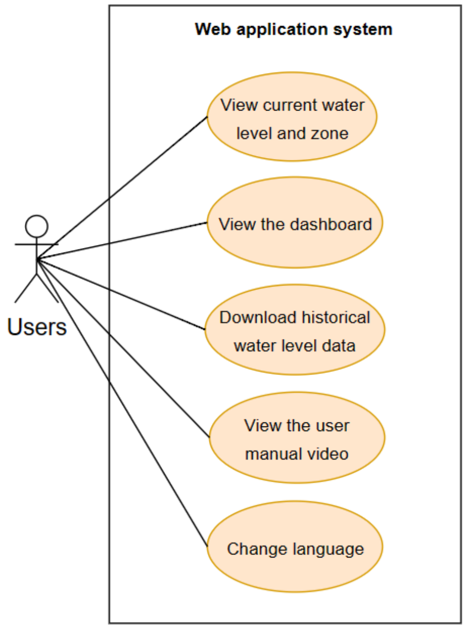

Figure 3.31 Use Cases Diagram for web application

The use case diagram for the web application, as shown in Figure 3.31, illustrates interactions between actors and key functionalities in the system. Municipality officers, Residents of Nakhon Si Thammarat, and General users, referred to as "Users," can access the web application to view current water levels, and zone, historical water level data through the dashboard, and the user manual video, to download historical water level data and also change the language of the web application. Users access and view historical water level data for the purpose of analysis, trend identification, and decision-making.

3.5.2 Notification System

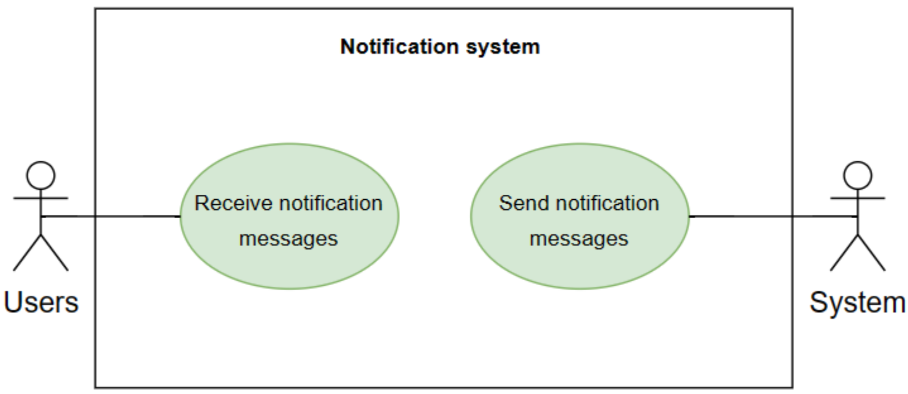

Figure 3.32 Use Cases Diagram for notification system

The use case diagram for the notification system, as shown in Figure 3.32, illustrates interactions between actors and key functionalities in the system. Municipality officers, referred to as "Users," receive notification messages. "System" sends notification messages to users in case of a zone change.

3.6 Use Cases Narrative

3.6.1 Web Application

There are 5 scenarios of the web application as shown below.

Table 3.1 Scenario 1

Scenario 1	View current water level and zone
Actors	Users
Goal	Users can view the water level and zone at the selected location.
Precondition	1. User can access the website through a web browser. 2. Users must navigate to the homepage. 3. Users must select a location from the dropdown menu.
Post-condition	If successful, the system displays water level, and zone for the selected location.
Main success scenario	1. User accesses the system through a web browser. 2. Users navigate to the homepage by clicking 'home' on the navigation bar. 3. User selects a specific location from the dropdown menu. 4. The system displays a map. 5. The system displays the real-time water level, and zone for the selected location.
Exceptional scenario	-

Table 3.2 Scenario 2

Scenario 2	View the dashboard
Actors	Users
Goal	The system displays the PowerBI dashboard.
Precondition	1. Users can access the website through a web browser. 2. Users must navigate to the dashboard page. 3. Users must select the start and end date. 4. Users must select a location.
Post-condition	If successful, the system displays historical water level data and graphs on the PowerBI dashboard.
Main success scenario	1. Users access the system through a web browser. 2. Users navigate to the dashboard page by clicking 'dashboard' on the navigation bar. 3. On the dashboard, users select the start and end date. 4. Users select a specific location to view the water level information. 5. The system displays historical water level data and graphs.
Exceptional scenario	-

Table 3.3 Scenario 3

Scenario 3	Download historical water level data
Actors	Users
Goal	Users can download historical water level data as a .csv file.
Precondition	1. Users can access the website through a web browser. 2. Users must navigate to the homepage. 3. Users must select a specific location and year from dropdown menus. 4. Users must click on the Download button.
Post-condition	If successful, the system generates and provides a downloadable file containing the historical water level data.
Main success scenario	1. Users access the system through a web browser. 2. Users navigate to the homepage by clicking ‘home’ on the navigation bar. 3. Users select a specific location from a dropdown menu. 4. Users select a specific year from a dropdown menu. 5. Users click on the Download button under the map. 6. The system provides historical water level data in the form of a .csv file.
Exceptional scenario	-

Table 3.4 Scenario 4

Scenario 4	View the user manual video
Actors	Users
Goal	The system displays the user manual video.
Precondition	1. Users can access the website through a web browser. 2. Users must navigate to the manual page.
Post-condition	If successful, the system displays the user manual video.
Main success scenario	1. Users access the system through a web browser. 2. Users navigate to the manual page by clicking ‘manual’ on the navigation bar. 3. The system displays the user manual video.
Exceptional scenario	-

Table 3.5 Scenario 5

Scenario 5	Change the language of the content
Actors	Users
Goal	The system alters the language of the content within the web application.
Precondition	1. Users can access the website through a web browser. 2. Users must select a language.
Post-condition	If successful, the system switches the language to the one that has been selected.
Main success scenario	1. Users access the system through a web browser. 2. Users select a language from the navigation bar. 3. the system switches the language to the one that has been selected.
Exceptional scenario	-

3.6.2 Notification System

There are 2 scenarios of the notification system as shown below.

Table 3.6 Scenario 6

Scenario 6	Receive notification messages
Actors	Users
Goal	Users receive notification messages in case of a zone change.
Precondition	1. Users must add Line Notify as a friend. 2. The zone is changed from green to yellow, yellow to red, red to yellow, or yellow to green.
Post-condition	If successful, user receives notification messages.
Main success scenario	1. The zone is changed from green to yellow, yellow to red, red to yellow, or yellow to green. 2. The system sends notification messages to users via Line Notify. 3. User receives notification messages.
Exceptional scenario	-

Table 3.7 Scenario 7

Scenario 7	Send notification messages
Actors	System
Goal	The system sends notification messages to users in case of a zone change.
Precondition	1. The system has Line Notify access token. 2. The zone is changed from green to yellow, yellow to red, red to yellow, or yellow to green.
Post-condition	If successful, the system sends notification messages to users.
Main success scenario	1. The system gets a Line Notify token. 2. The zone is changed from green to yellow, yellow to red, red to yellow, or yellow to green. 3. The system sends notification messages to users via Line Notify.
Exceptional scenario	-

3.7 Database Schema

3.7.1 ER Diagram

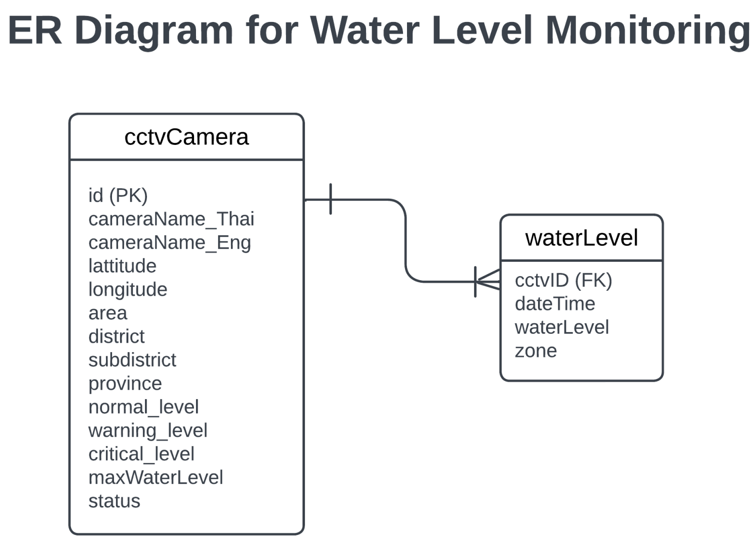

Figure 3.33 ER Diagram

This ER diagram in Figure 3.33 illustrates the connection between two tables; 'cctvCamera' and 'waterLevel.' The 'cctvCamera' table encompasses details about each camera, including its name, geographical area, latitude, longitude, normal, warning, critical, and max water level, as well as its status (active or inactive). On the other hand, the 'waterLevel' table captures the real-time water data, recording the date-time, current water level value and zone. The details of each table are shown here.

3.7.2 Class Diagram

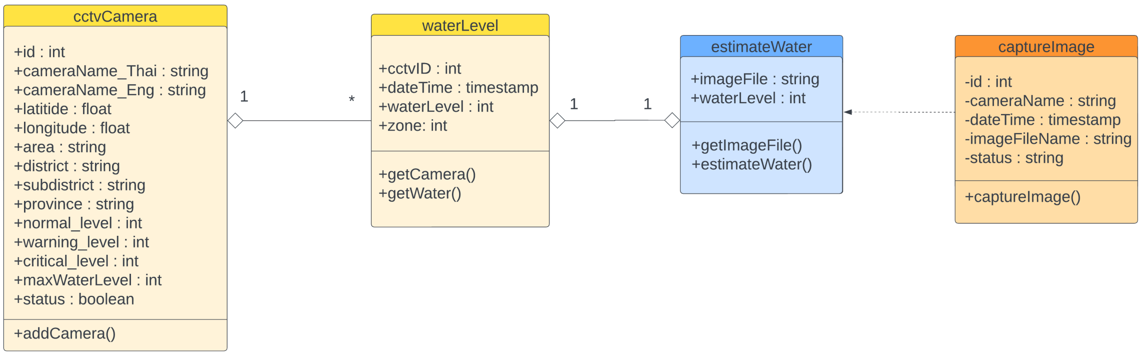

Figure 3.34 Class diagram

The class diagram includes four classes, represented in Figure 3.34. The details of each class are outlined here.

- cctvCamera class
  This class is responsible for storing camera information such as camera ID, name, latitude, longitude, maximum water level, and status. The method “addCamera()” in this class facilitates the retrieval of camera information.

- waterLevel class
  In “waterLevel” class stores current water level data obtained from the image processing. It provides methods “getCamera()” for obtaining camera information and “getWater()” for retrieving water level data.

- estimateWater class
  Utilized to estimate water level using the image processing algorithm based on captured images. The method “getImageFile()” returns the filename of the image, and “estimateWater()” provides the water level data after estimation.

- captureImage class
  The “captureImage” class is designed for capturing images every 2 minutes. Its method “captureImage()” returns the image filename format.

These four classes are interconnected to facilitate the flow of information, obtaining camera details, capturing images, estimating water levels, and retrieving water level data.

cctvID	dateTime	waterLevel	zone
1	2023-10-10 06:10:00 UTC	1	0
1	2023-10-10 06:06:00 UTC	1	0
1	2023-10-10 06:02:00 UTC	1	0
1	2023-10-10 06:16:00 UTC	1	0
1	2023-10-10 06:08:00 UTC	1	0
1	2023-10-10 06:04:00 UTC	1	0
1	2023-10-10 06:12:00 UTC	1	0
1	2023-10-10 06:00:00 UTC	1	0
1	2023-10-10 06:14:00 UTC	1	0
1	2023-10-10 06:26:00 UTC	2	0
1	2023-10-10 06:18:00 UTC	2	0
1	2023-10-10 06:24:00 UTC	2	0
1	2023-10-10 06:22:00 UTC	2	0
1	2023-10-10 06:20:00 UTC	2	0
1	2023-10-10 06:28:00 UTC	2	0
1	2023-10-10 06:38:00 UTC	3	0
1	2023-10-10 06:40:00 UTC	3	0

Figure 3.43 The downloaded water level data

From Figure 3.39, when users click the download button (I), a file will be saved in CSV format on the user's device, containing essential historical water level data as CCTV ID, date and time, water level, and zone, as shown in Figure 3.43.

3.9 Navigation Diagram

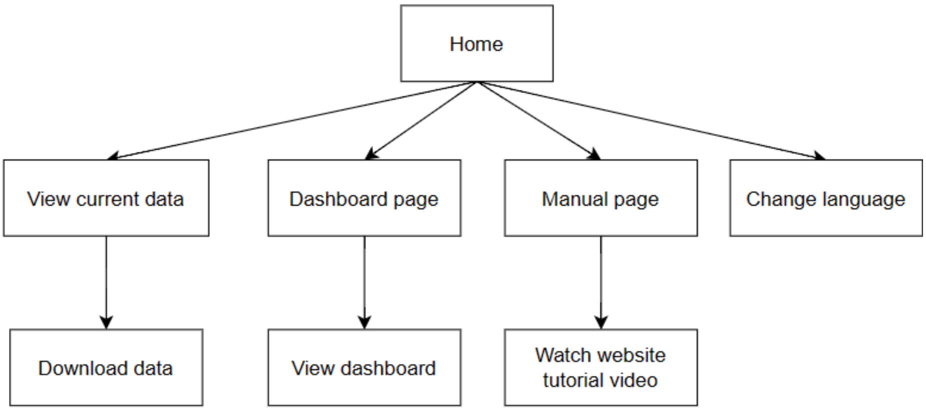

Figure 3.44 Navigation Diagram

This navigation diagram, as shown in Figure 3.44, illustrates the user's route, commencing at the 'Home,' which serves as the homepage. In homepage, users have options to view current water level data and download historical water level data. Users can navigate to the 'Dashboard page' for viewing a stacked area chart of water levels or navigate to the 'Manual page' for watching tutorial video on how to use the website. Furthermore, users have the option to switch the language of the displayed website content.

3.10 Sequence Diagram

3.10.1 Web Application Sequence Diagram

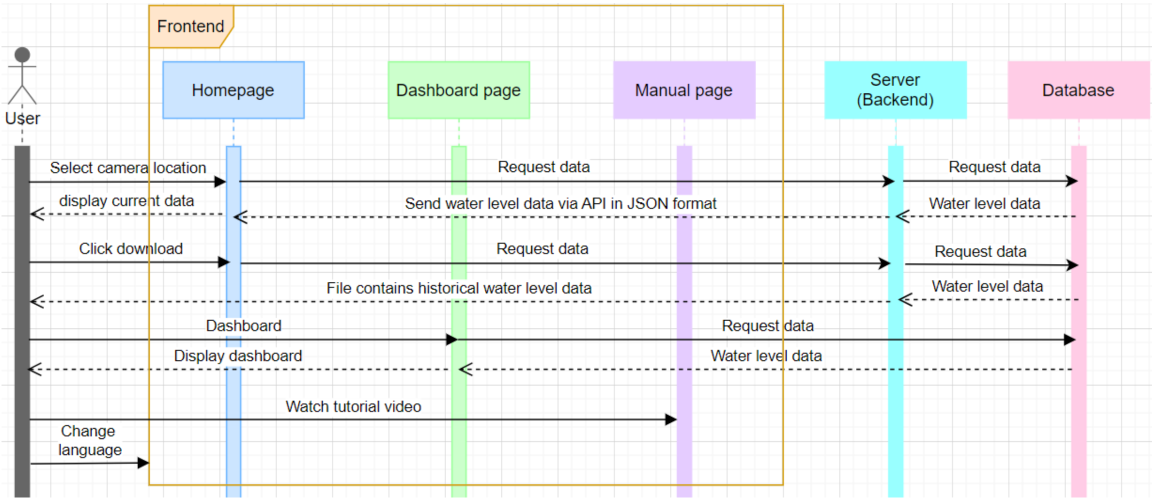

Figure 3.45 Web Application Sequence Diagram

The web application sequence diagram, as shown in Figure 3.45, visualizes the interactions between components in the web application. When user visit the homepage and selects the desired camera location, a data request is sent to the backend which is deployed on depa's server. Subsequently, the backend sends a request of current water level data to the Google BigQuery database. The database, in turn, returns the resulting water level data to the backend, which then sends the water level data back to the homepage via API in JSON format. The homepage retrieves and displays current water level data to users.

When users click the download button on the homepage, a data request is sent to the backend. Subsequently, the backend sends a request of historical water level data to the Google BigQuery database. The database, in turn, returns the resulting historical data to the backend. The backend will provide users with the historical water level data file in .csv format, which will be saved on the user's device.

When users visit the dashboard page, PowerBI sends a request for water level data to the Google BigQuery database. The database then returns the resulting water level data to PowerBI, which retrieves and displays the dashboard to users.

When users visit the manual page, users have the option to watch a tutorial video that explains how to use the website. Furthermore, users can change the language of the website's content. This change will affect the language of the homepage, dashboard page, and manual page according to the user's selection.

3.10.2 Notification System Sequence Diagram

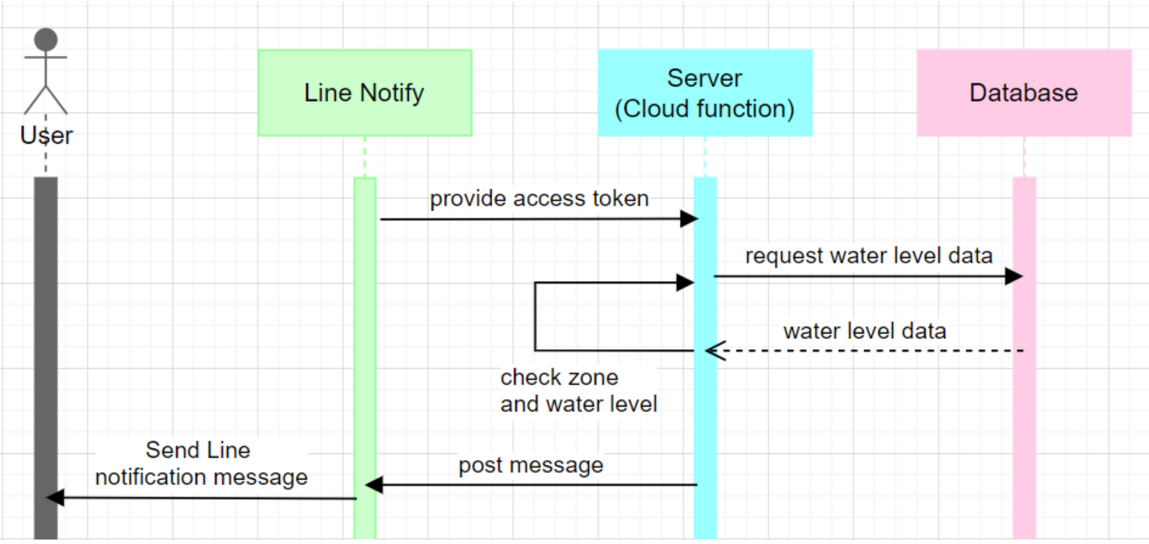

Figure 3.46 Notification System Sequence Diagram

The notification system sequence diagram, as shown in Figure 3.46, visualizes the interactions between components in the notification system. Line Notify will provide an access token to the notification algorithm, deployed on Cloud function, facilitating communication that allows the server to send notification messages back to Line Notify. The algorithm sends a request for water level data to the Google BigQuery database. The database, in turn, returns the resulting water level data to the algorithm. The algorithm will check for a zone change. If a zone change is detected, the program triggers the posting of notification messages to Line Notify. Subsequently, Line Notify sends these notification messages to users via Line. In the absence of a zone change, no notification is sent, and the server continues to check.

3.11 User roles

3.11.1 User roles in the web application

The user roles in the web application are municipality officers, residents of Nakhon Si Thammarat, and general users. The key tasks associated with these roles are:

- View current water level data.
- Download historical water level data.
- View the dashboard.
- Watch the tutorial video demonstrating how to use the website.
- Change the language of the displayed website's content.

3.11.2 User roles in the notification system

The user role in the notification System is municipality officers. The key task associated with this role is:

- Receive the notification messages in case of a zone change.