Product Design

Team 45
Sai Shashank Kalakonda
Kalyan Adithya M
Jaiwanth Mandava
Abhishek Reddy G

Design Overview

Architectural design

1. Admin - It has the 'login' method to sign in. Admin has tools required for maintaining the service.
2. Monitors - They are another set of users given access to the map view of the corridors along with the suggested eradication technique and is also given access to data to export it for further use.
3. Public - These are the mobile app users who share their location for determining the paths. They can ping the server in case of an outbreak and they receive a video and a tip daily to educate them.
4. Maps - It constitutes the main logic of creation of paths and visualizing it.
5. Sensor - It instantaneously stores the information retrieved from each sensor and uploads it to the database.
6. Corridors - it stores information about each corridor and evaluates it to suggest an eradication technique.

System interfaces
User Interface
WEBAPP

Initial page will be asking for whether the user is a Monitor or Admin, after selecting the appropriate choice it asks for login credentials after validating both the admin and monitor is provided with a map view in the center of the webpage along with an option beside it for exporting the data or map. The map shows the corridors along with its eradicating technique when clicked upon. Beside these the Admin has access to a sensor management page where all the sensors along with its location is listed. To this list an add option is given beside to add a new sensor and a cross beside each sensor in the list to remove it for the database. The admin has another page where a new monitor can be added.
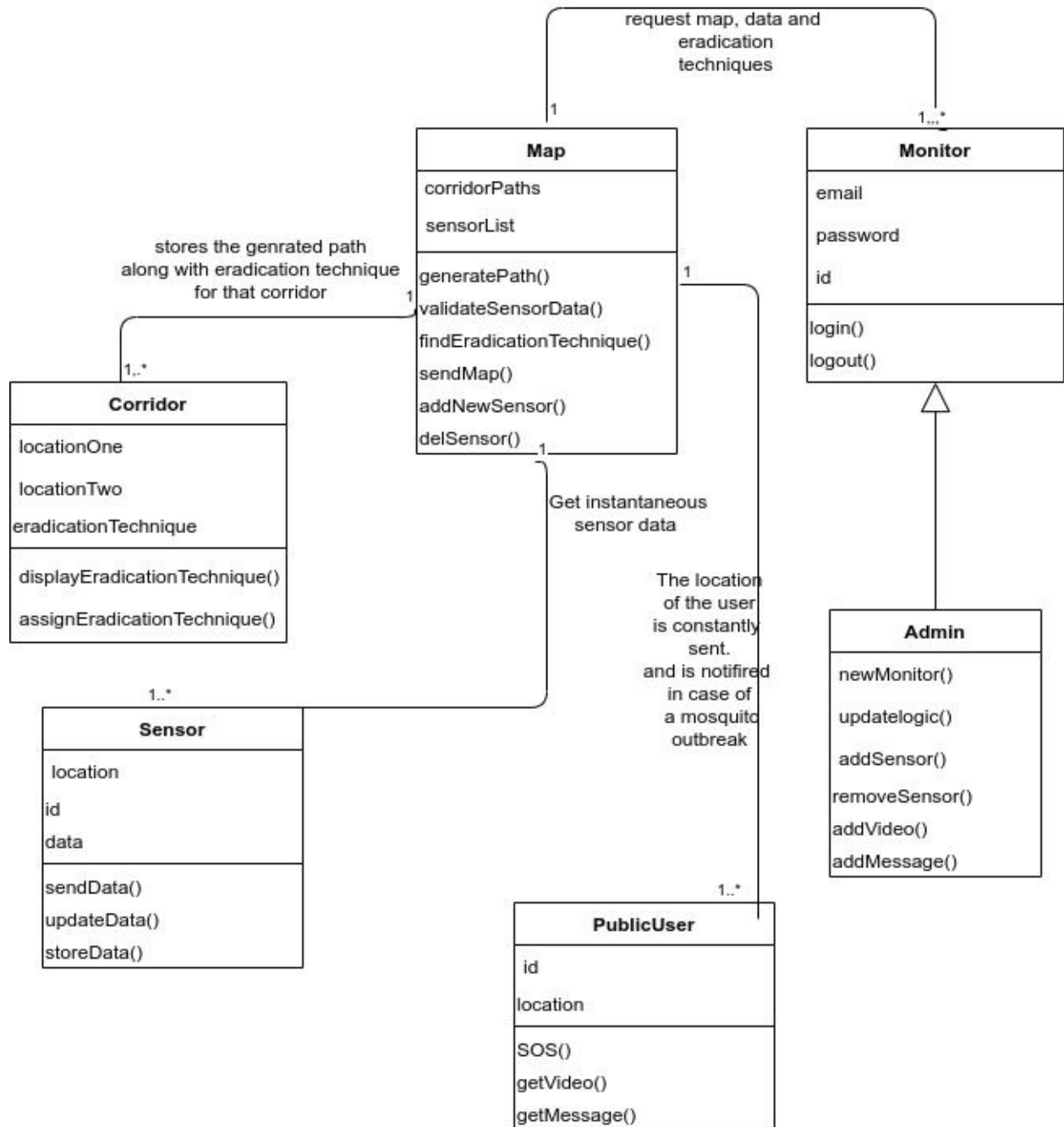
MOBILE APP

The initial page has an SOS button to ping the server and has an option to view a video and is show a tip for educating the user in methods of how to kill mosquitoes without chemicals.

APIs

We don't provide any API to interact with the system.

Model

Map
- corridorPaths
- sensorList
- generatePath()
- validateSensorData()
- findEradicationTechnique()
- sendMap()
- addNewSensor()
- delSensor()

Monitor
- email
- password
- id
- login()
- logout()

request map, data and eradication techniques

stores the genrated path along with eradication technique for that corridor

Corridor
- locationOne
- locationTwo
- eradicationTechnique
- displayEradicationTechnique()
- assignEradicationTechnique()

Sensor
- location
- id
- data
- sendData()
- updateData()
- storeData()

Get instantaneous sensor data

The location of the user is constantly sent. and is notifired in case of a mosquito outbreak

Admin
- newMonitor()
- updatelogic()
- addSensor()
- removeSensor()
- addVideo()
- addMessage()

PublicUser
- id
- location
- SOS()
- getVideo()
- getMessage()

Admin	Class State: ● Responsible for holding information regarding registered monitors, sensor locations, range of values that the parameters consider in determining the paths. Class behaviour: ● Login/Logout. ● Method for adding/removing sensors. ● Method for registering new monitors. ● Allowed to add new videos/messages to the database. ● Can tweak the logic as required by changing the ranges of the parameters considered in determining the paths.
Monitor	Class State: ● User, holding unique email, password. Class behaviour: ● Login/Logout. ● Viewing of corridor paths on map, also respective eradication technique. ● Methods to get sensor data, map on given date/time.
Public User	Class State: ● User, holds information on current location. Class behaviour: ● SOS button to inform Map entity regarding mosquito outbreaks. ● View displayed video, targeted message. ● Continuously allows access to his/her current location.
Map	Class State: ● Responsible for holding information regarding sensor locations, corridor paths and their eradication techniques. Class behaviour: ● Method to get sensor data from server. ● Method to validate data from the sensors. ● Generate paths based on sensor data(macro and micro). ● Evaluate relevant eradication technique. ● Add/remove sensor nodes. ● Display paths on the map and the map itself,
Corridors	Class State: ● Holds information regarding the origin and destination of the corridor path and relevant eradication technique. Class behaviour: ● Method to assign an eradication technique to the corridor path. ● Display eradication technique.
Sensor	Class State: ● Location and data of the sensor node. Class behaviour: ● Display relevant sensor data. ● Store sensor data in database.

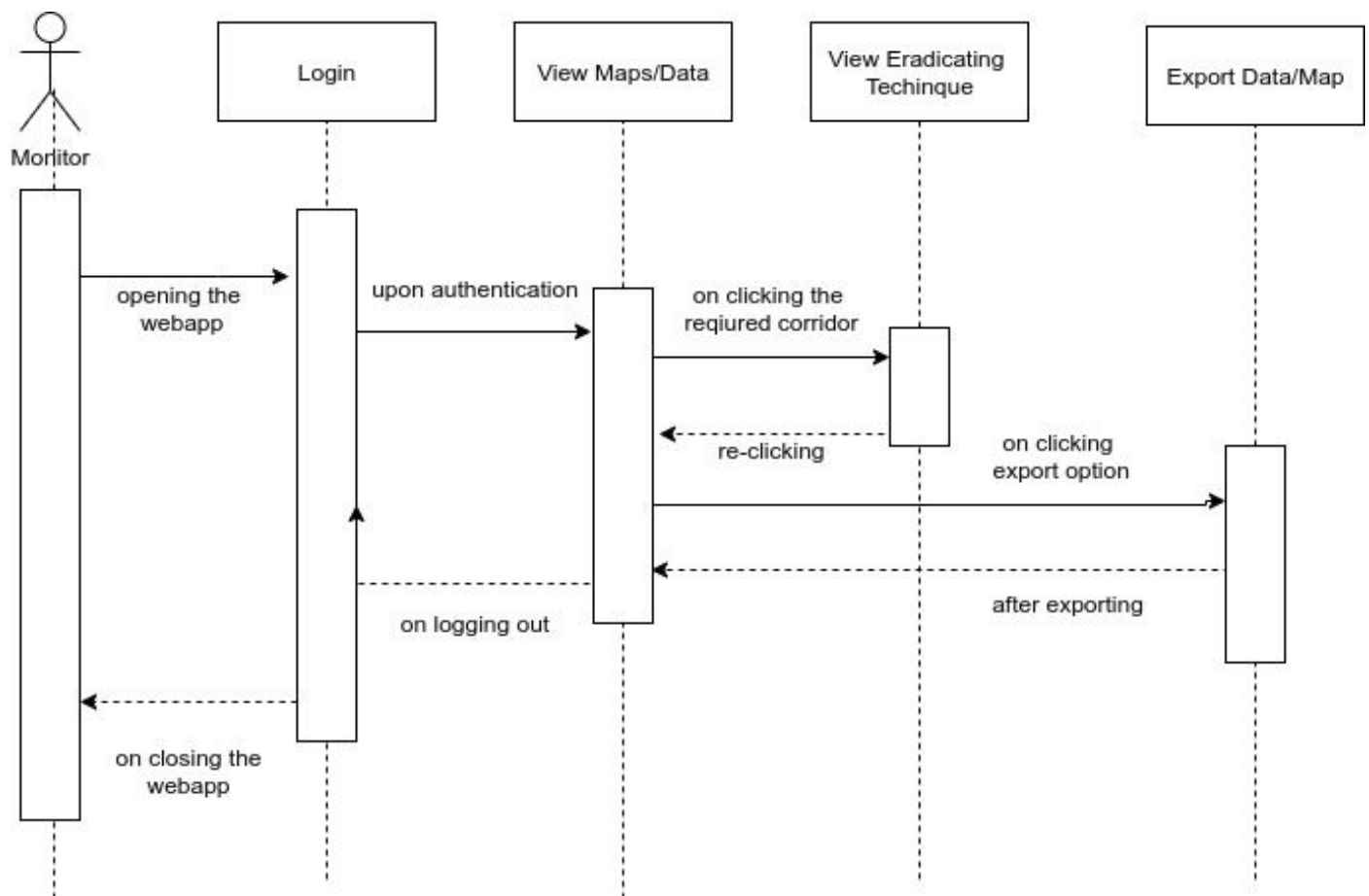|	● Update stored data.

Sequence Diagram

Sequence diagrams for each user is shown below:
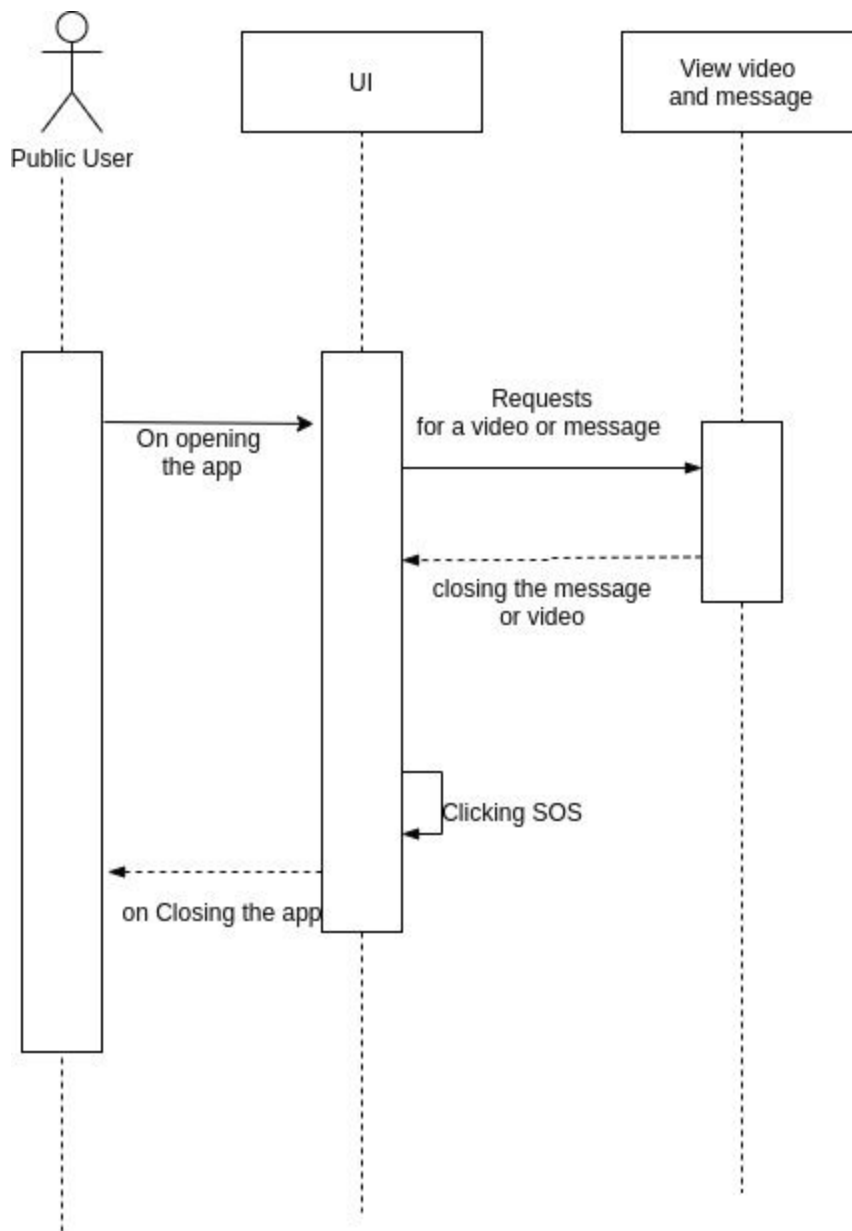
1) Admin

   Use this link

   https://iiitaphyd-my.sharepoint.com/:i:/g/personal/abhishek_gangapuram_students_iiit_ac_in/ERs_VhlZO_NApGfV1GvgVmsBtEsHwWzYIX6JprXjRamfrw?e=PLIASq

2) Monitor

3) Public User

Design Rationale

Each item rendered on map is identified by a class for easier implementation.