

Product Design

Team 45

Sai Shashank Kalakonda

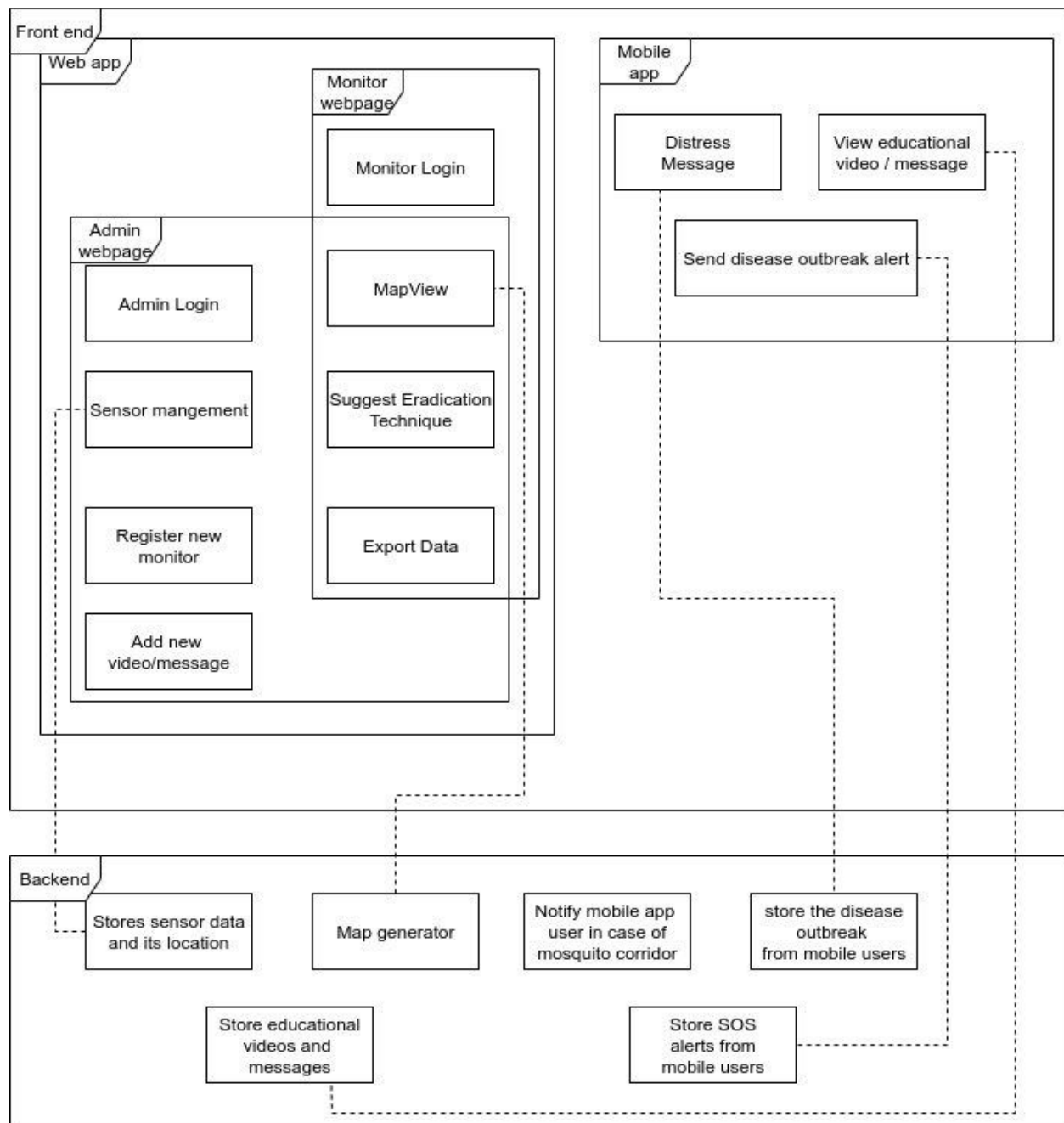
Kalyan Adithya M

Jaiwanth Mandava

Abhishek Reddy G

Design Overview

Architectural design



The system consists of two components, web application and mobile application which constitute the frontend of the system. Both components rely on backend support for required functionality as shown in the UML diagram using dashed lines.

1) Web application :

Supports two types of users, namely admin and monitor. Both these users have access to a few common features such as “Map/Data view” etc, as illustrated above. The web app is accessed after successful authentication of the user, upon which they are redirected to their respective set of pages with relevant functionality as described in the diagram.

2) Mobile application :

Users 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. This component works independently from Web application however indirectly interact with it through backend.

System interfaces

User Interface

WEB APP

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.

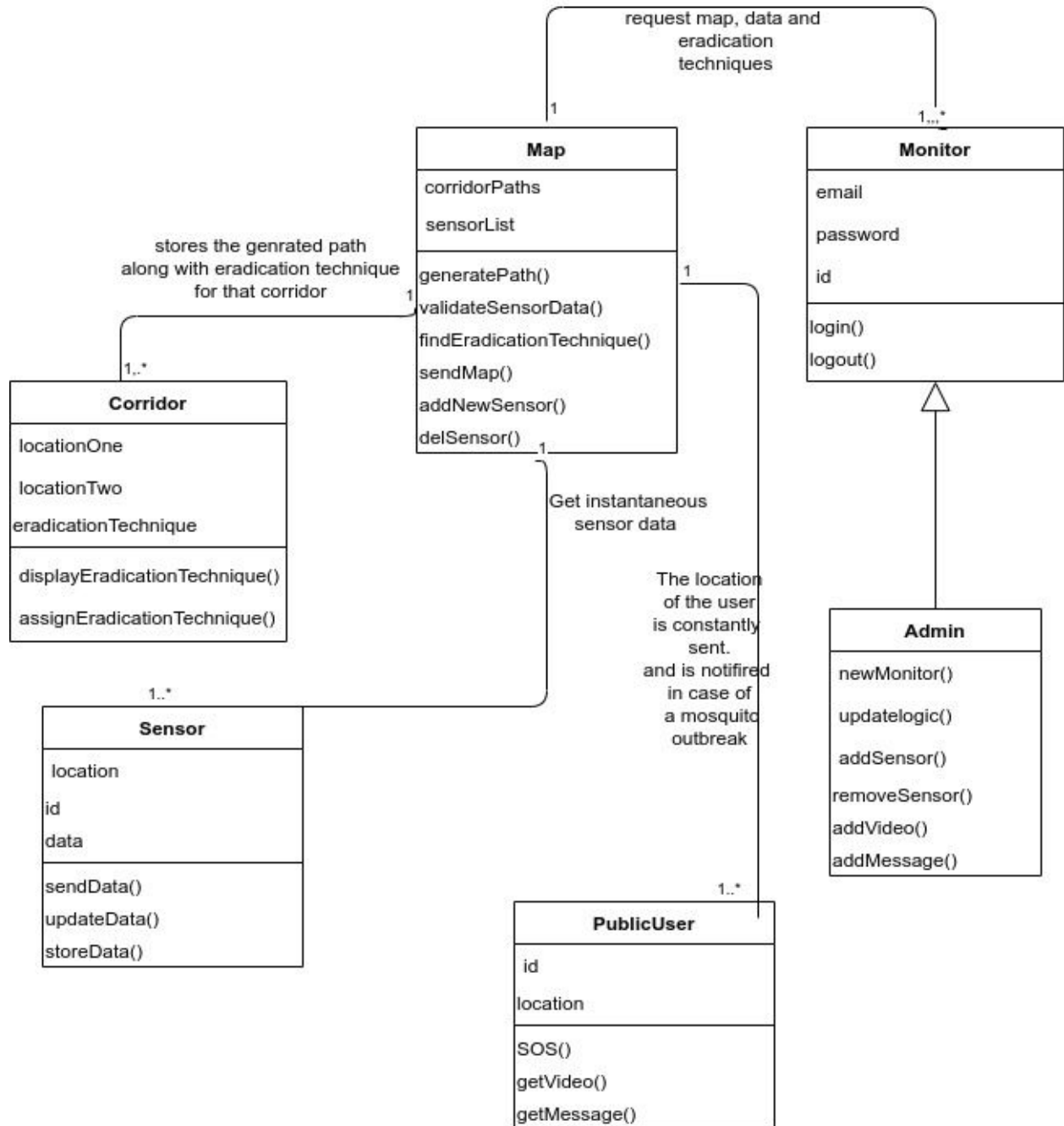
MOBILE APP

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



<i>Admin</i>	<p>Class State:</p> <ul style="list-style-type: none"> Responsible for holding information regarding registered monitors, sensor locations, range of values that the parameters consider in determining the paths. <p>Class behaviour:</p> <ul style="list-style-type: none"> 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	<p>Class State:</p> <ul style="list-style-type: none"> User, holding unique email, password. <p>Class behaviour:</p> <ul style="list-style-type: none"> Login/Logout. Viewing of corridor paths on map, also respective eradication technique. Methods to get sensor data, map on given date/time.
Public User	<p>Class State:</p> <ul style="list-style-type: none"> User, holds information on current location. <p>Class behaviour:</p> <ul style="list-style-type: none"> SOS button to inform Map entity regarding mosquito outbreaks. View displayed video, targeted message. Continuously allows access to his/her current location.
Map	<p>Class State:</p> <ul style="list-style-type: none"> Responsible for holding information regarding sensor locations, corridor paths and their eradication techniques. <p>Class behaviour:</p> <ul style="list-style-type: none"> 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	<p>Class State:</p> <ul style="list-style-type: none"> Holds information regarding the origin and destination of the corridor path and relevant eradication technique. <p>Class behaviour:</p> <ul style="list-style-type: none"> Method to assign an eradication technique to the corridor path. Display eradication technique.
Sensor	<p>Class State:</p> <ul style="list-style-type: none"> Location and data of the sensor node. <p>Class behaviour:</p> <ul style="list-style-type: none"> Display relevant sensor data. Store sensor data in database. Update stored data.

Sequence Diagram

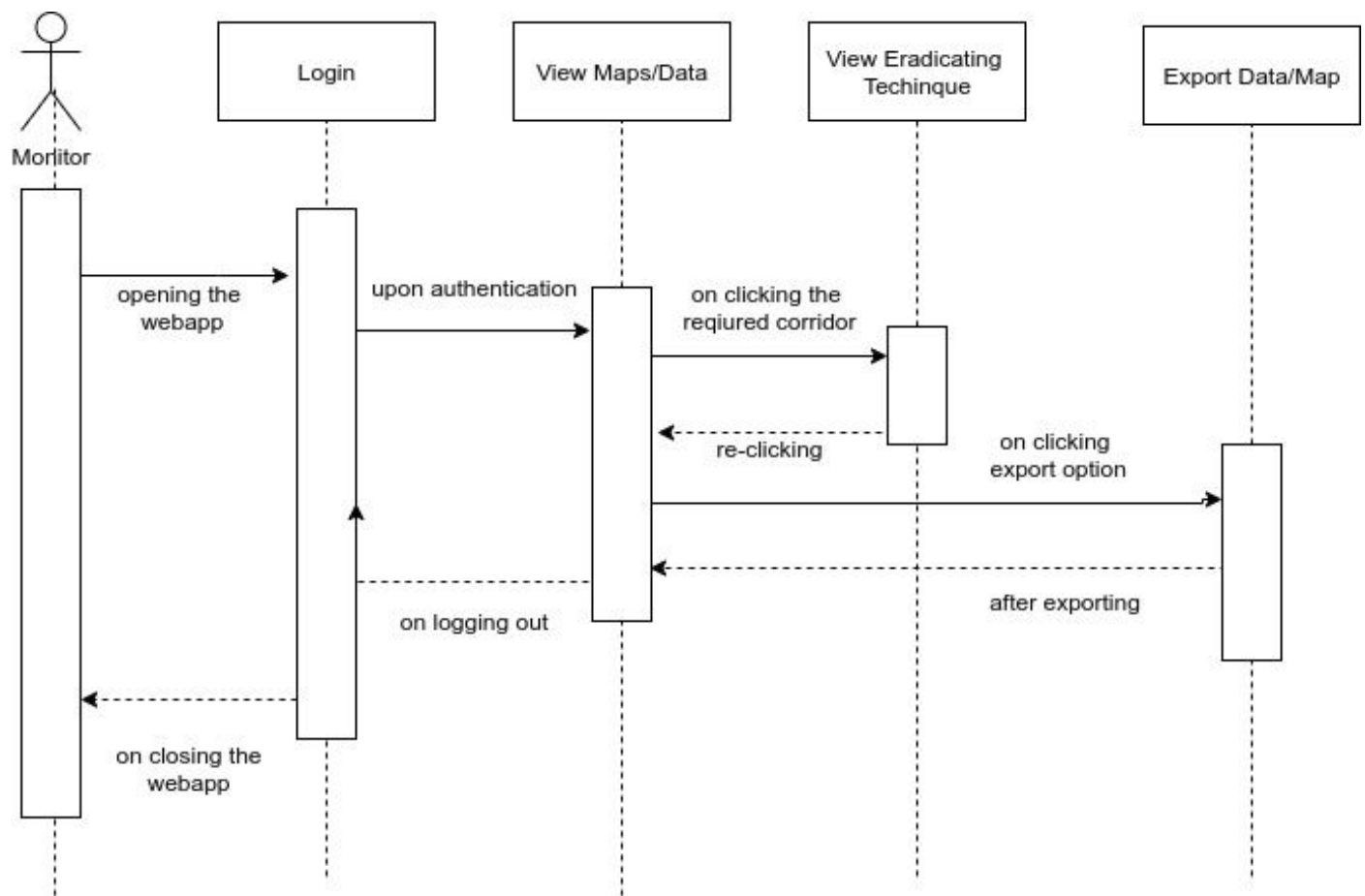
Sequence diagrams for each user is shown below:

1) Admin

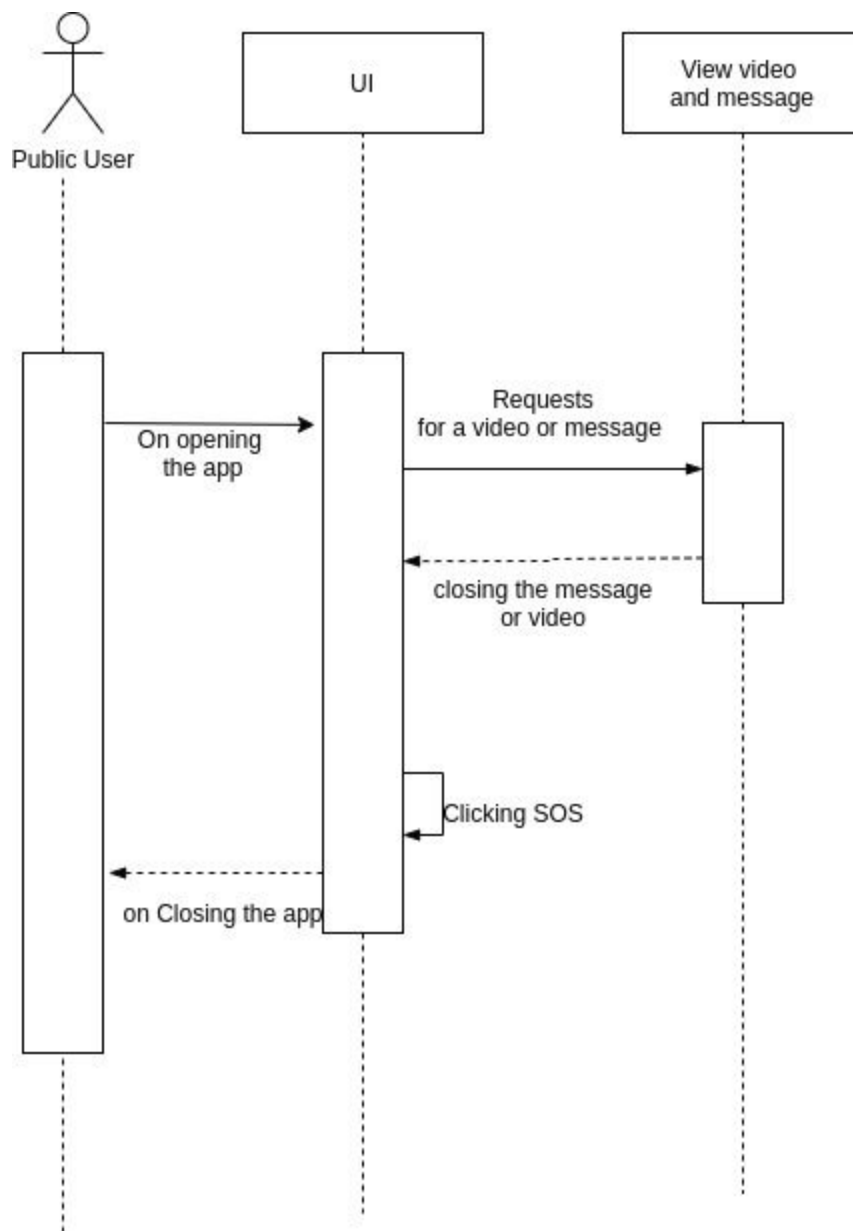
Use this link

https://iitaphyd-my.sharepoint.com/:i:/g/personal/abhishek_gangapuram_student_s_iit_ac_in/ERs_VhIZO_NApGfV1GvgVmsBtEsHwWzYIX6JprXjRamfrw?e=PLIASq

2) Monitor



3) Public User



Design Rationale

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