CSE307 Term Project

Project Name: SOS Emergency Service App

Section: 01

by

Name	ID
Saquib Ali Khan	1821908

Section 1: **2**
Project Introduction: 2
History leading to project requests: 3
Identifying problems, solutions and opportunities: 3
Project goal and objective: 3

Section 2: **4**
Product Description: 4
Context Level Data Flow Diagram: 5
Software Key Features: 5

Section 3: **6**
Information Gathering Methods: 6
Major functionalities offered by the system: 7
Use Case Diagram: 7
Normal Scenario: 7
Functional Requirements: 8
Non-Functional Requirements: 9

Section 4: **10**
Entity Relationship Diagram: 10
Logical Data Flow Diagram: 11
Physical Data Flow Diagram: 12
Activity Diagram: 13
Sequence Diagram: 17
Class Diagram: 18
State-chart Diagram: 19
CRUD matrix: 19

Section 5: **20**
Structure English pseudo code for the system: 20
User Interface Prototype: 21

Section 1:

Project Introduction:

Our aim is to create an app for emergency services to help users get in touch with proper help with a few clicks of a button.

History leading to project requests:

This app will help users contact emergency services discreetly and will be much easier than having to communicate with a phone call. The user will be able to alert the police of a crime happening without alerting the offender. Users can also get medical help in case of situations where the user cannot verbally communicate.

Identifying problems, solutions and opportunities:

Problems	Solutions
Government approval is needed.	Can be solved by contacting government officials.
The employees might not be set out to fill the roles needed.	They can be trained before being put to work officially.
Some payment methods can go down sometimes.	All the popular options can be added so that the customer can have options when one goes down

Opportunities:

1. The reliance on a phone call for emergency service can be reduced.
2. By attracting investors like insurance companies a steady income can be created.
3. More features can be added from feedback to help local societies.

Project goal and objective:

Goal 1: Make a mobile application

Objectives:

- Make a mobile application with simple UI
- Make the application communicate with the caller through in app messages so that the caller can know exactly what to say.

Goal 2: Create hype in social media platforms

Objectives:

- Promote the app on social media.
- The users can download the app in their iphone or android phones from the link in the advertisement.

Goal 3: Hire more people and refine app based on feedback

Objectives: Improve the app based on user feedback and hire more people to handle more users to handle traffic.

Goal 4: Attract more investors

Objectives: After initial app release, we will try to make it a government approved official app for emergency services and also look for investors whose advertisement we can display in our app.

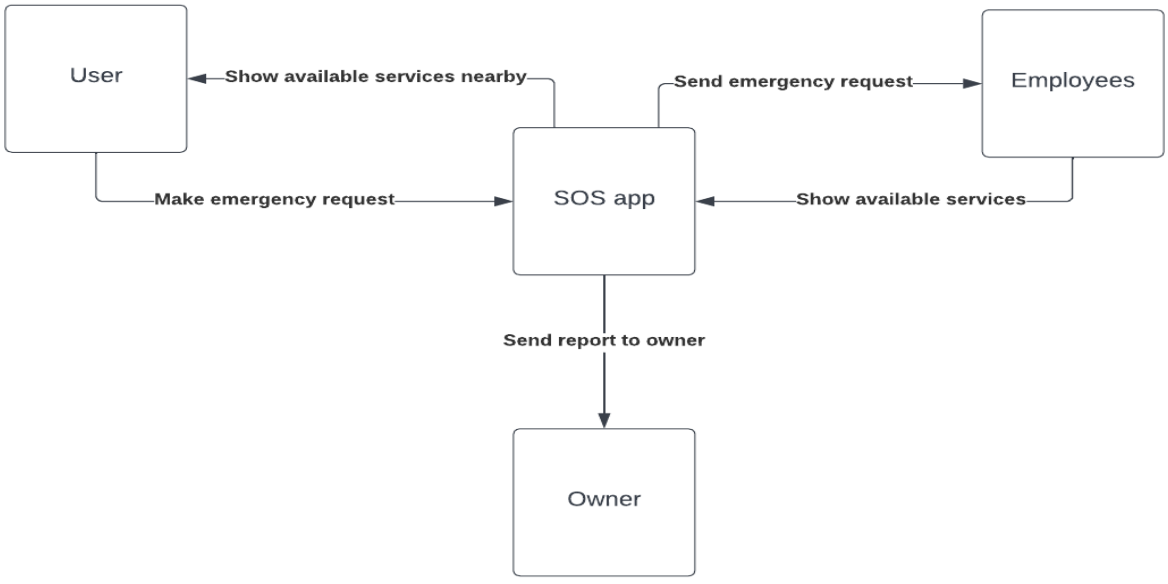
Section 2:

Product Description:

A. Product Summary:
“SOS App” is a mobile application that can be used to contact any emergency service through in app messages discreetly.

B. Product Stakeholders:
- Founder
- Co-founders
- App developer
- Server Admin
- Employees to play roles
- Maintenance crew
- Customer

Context Level Data Flow Diagram:



Software Key Features:

- Users can create an account with email and phone number.
- The user will be tracked in case of emergency.
- There will be a safespot button to show the user the nearest safe zone in his maps and alert officials near him.

Section 3:

Information Gathering Methods:

Interviewing:
Interviews can be taken in the test phase before the app launch to find problems with the system. Government officials and random users will be picked for interviews.

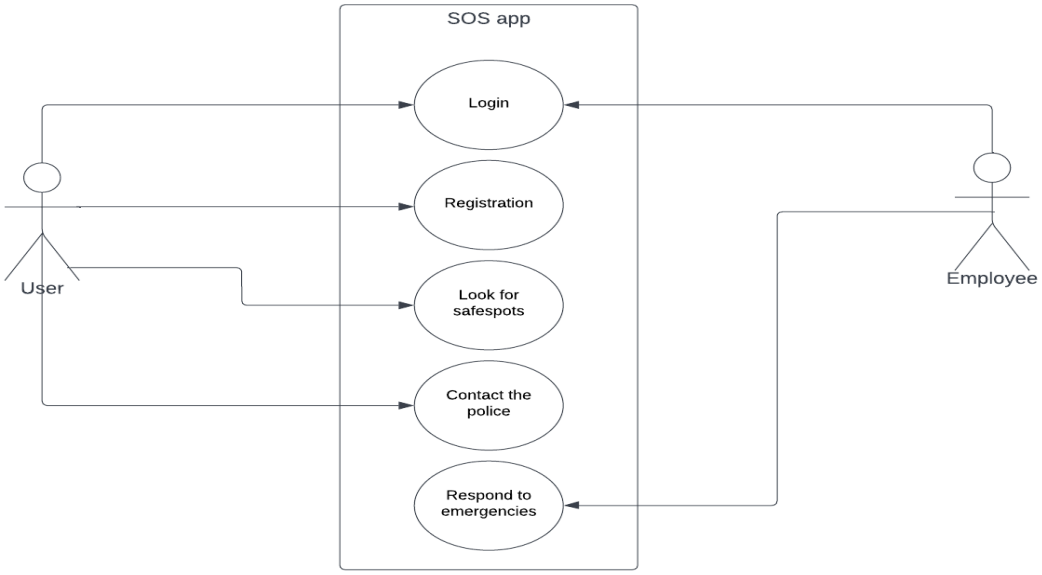
Joint Application Design (JAD):
Randomly selected users and government officials can be invited to join a joint application design to test new features before releasing them. They can also be encouraged to do a discussion and give opinions on features they might like in future for the app.

Questionnaires:
Questionnaires can be created and given to random users and government officials to get feedback on the app that will be used to improve the app.

Major functionalities offered by the system:

- Location tracking service will be added to the system.
- Distress signals will be added to the system in case of emergencies.
- Chatting features will be added to communicate with the operator discreetly
- Recording features will be added to the system.

Use Case Diagram:



Normal Scenario:

Use case name : Login	UniqueID: 111A
Level: Blue	
Description: User logs in to account	
Triggering event : Customer uses the app to get into the login page and enters the userID and password and clicks on the	
Steps performed	Information types
1 . Customer use the app to enter the userID and password and clicks on the login button to log in	userID , password
2 . A successful login confirmation is shown and user enters his/her account	User shown homepage
Precondition: A mobile phone with internet connection and customer has completed registration	
Postconditions : Customer successfully logged in	
Assumptions: Customer has valid userID and password	
Risk: Medium	
Priority: High	

Use case name : Registration	UniqueID:112B
Actors : Customer	
Level: Blue	
Description: Customer registers	
Triggering event : Customer uses the app to enter information and clicks on the register button to register	
Steps performed	**Information types**
1. Customer enters the registration information and clicks submit	Name , Contact , Address
2 . Customer is led to a successful registration webpage and is prompted to login for the first time	Registration confirmation
Precondition: A mobile with internet connection and valid information	
Postconditions : Successfully registered	
Assumptions: Customer has valid credentials and access to internet	
Risk: Medium	
Priority: High	

Functional Requirements:

Database:
Databases are needed to store users, operators, etc.

Backup:
There should be a backup server for recovery.

Report a problem:
There should be an option to report any problem with the app.

Government approval required:
Government approval is required to make the app official at some point.

Algorithm:
There should be an algorithm that shows the user the nearest safespot based on the location data.

Non-Functional Requirements:

Security:
The app needs to be extra secure with the best security out there.

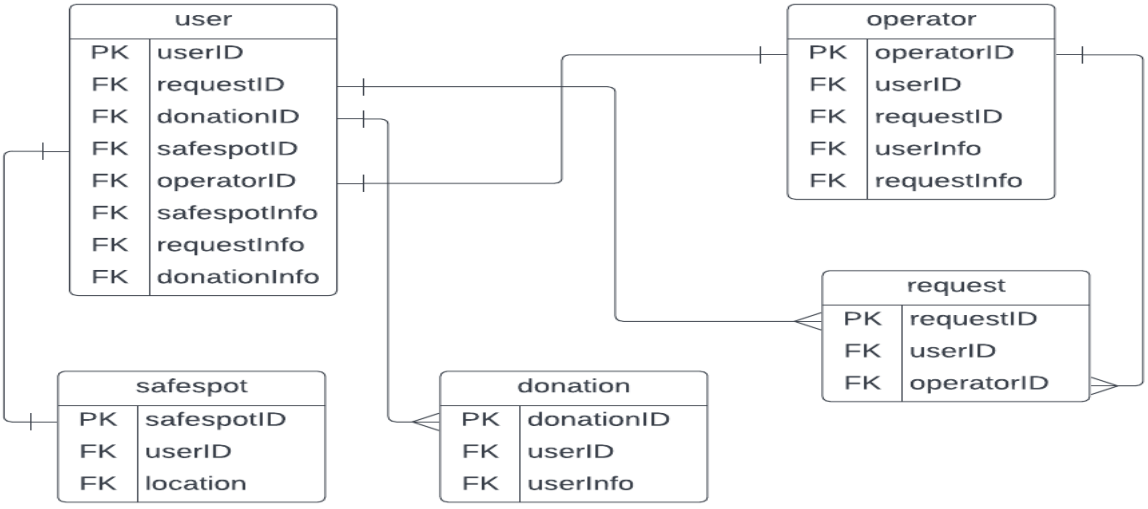
Efficient:
Duplicate functions should be avoided which will make the application much more efficient.

Performance:
The app needs to respond in seconds in case there is an emergency situation.

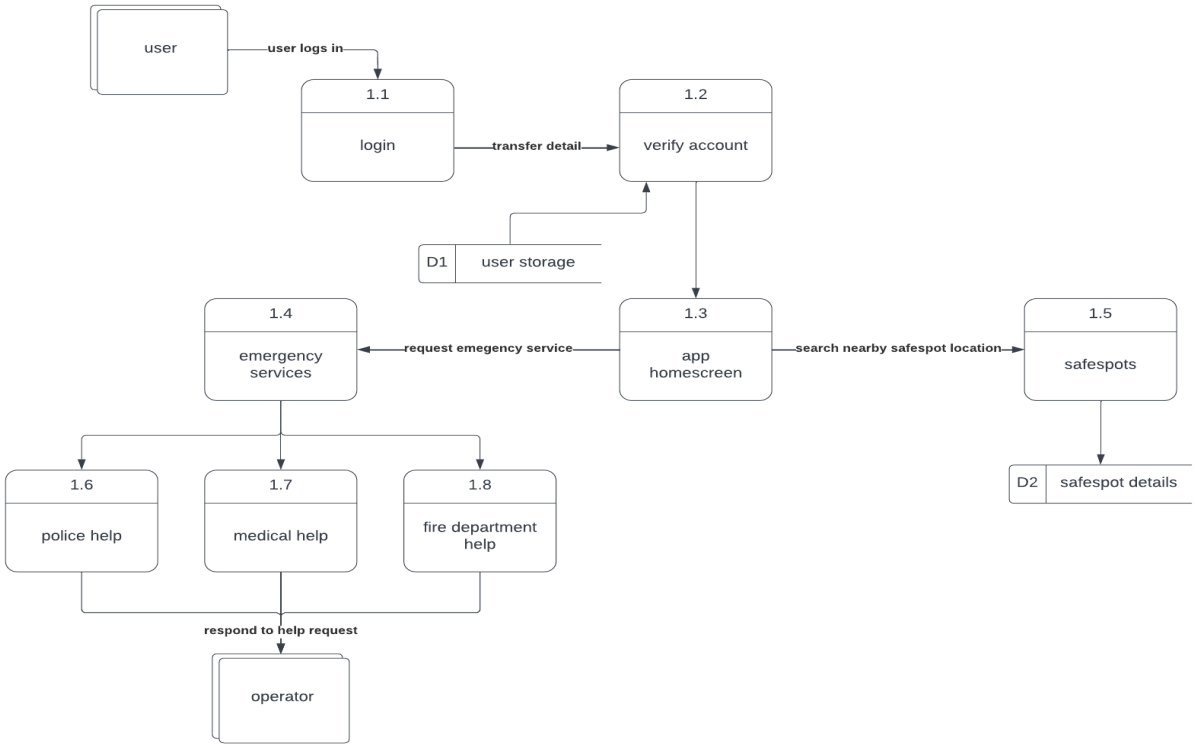
Service:
The app needs to be able to function in all kinds of mobile devices.

Section 4:

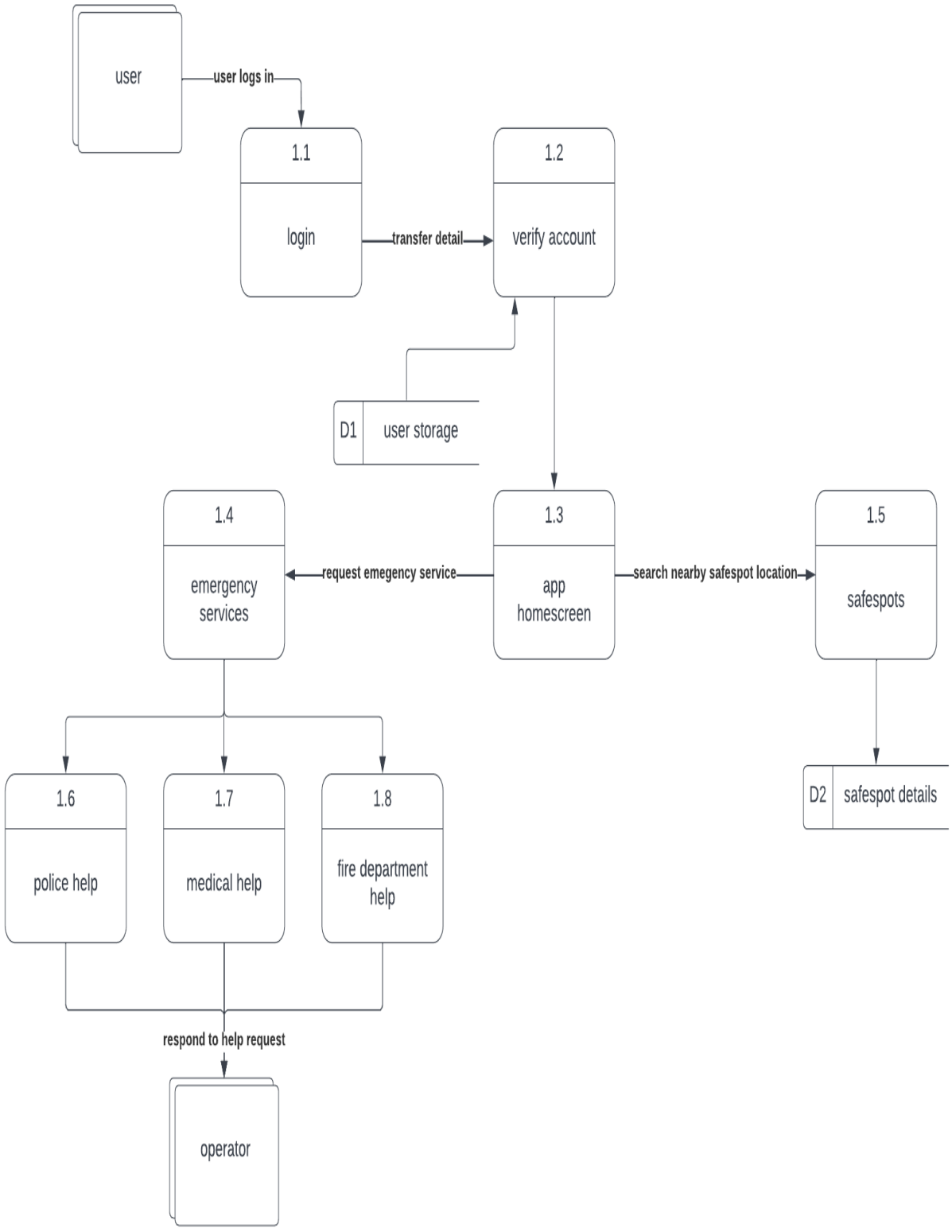
Entity Relationship Diagram:



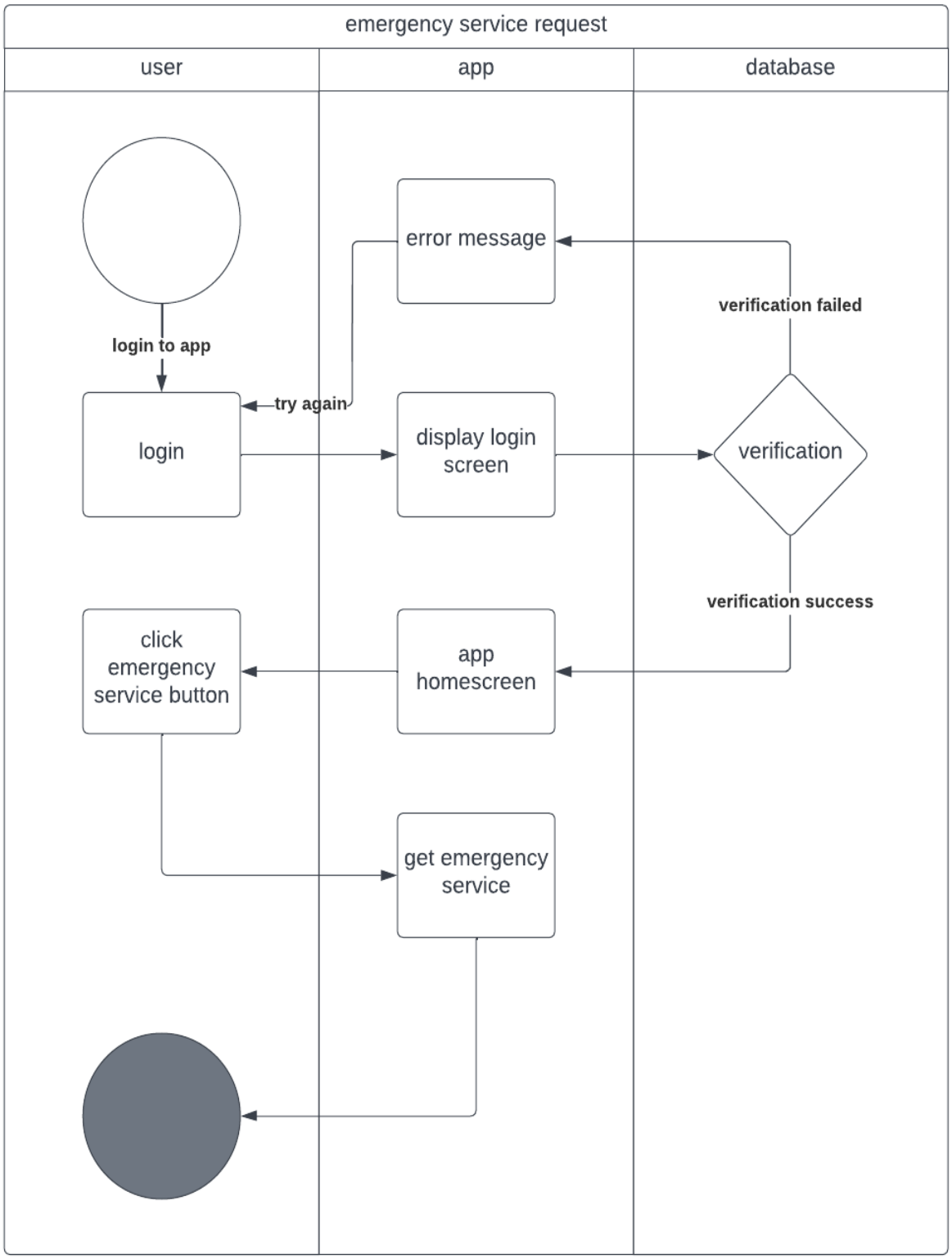
Logical Data Flow Diagram:

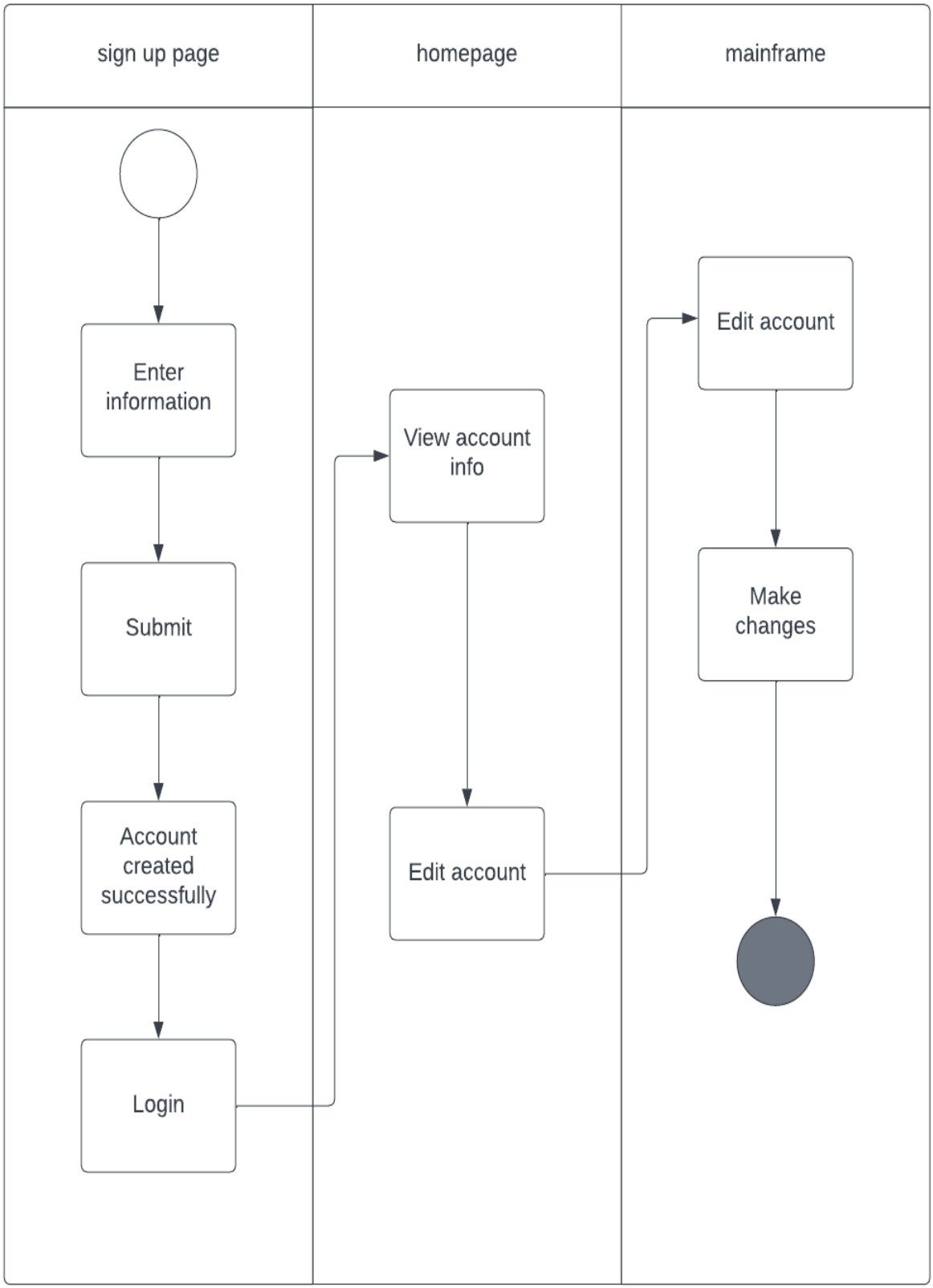


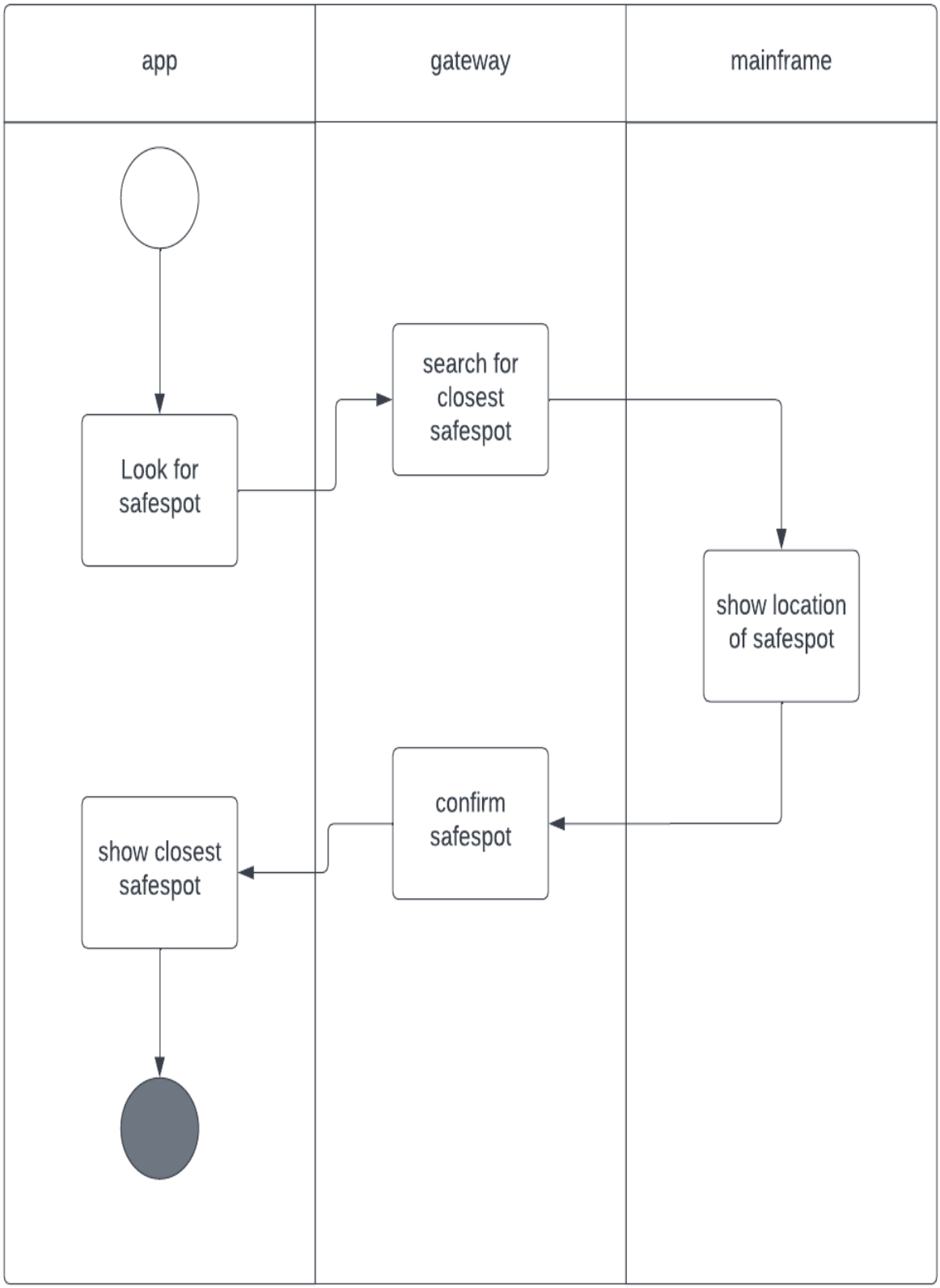
Physical Data Flow Diagram:

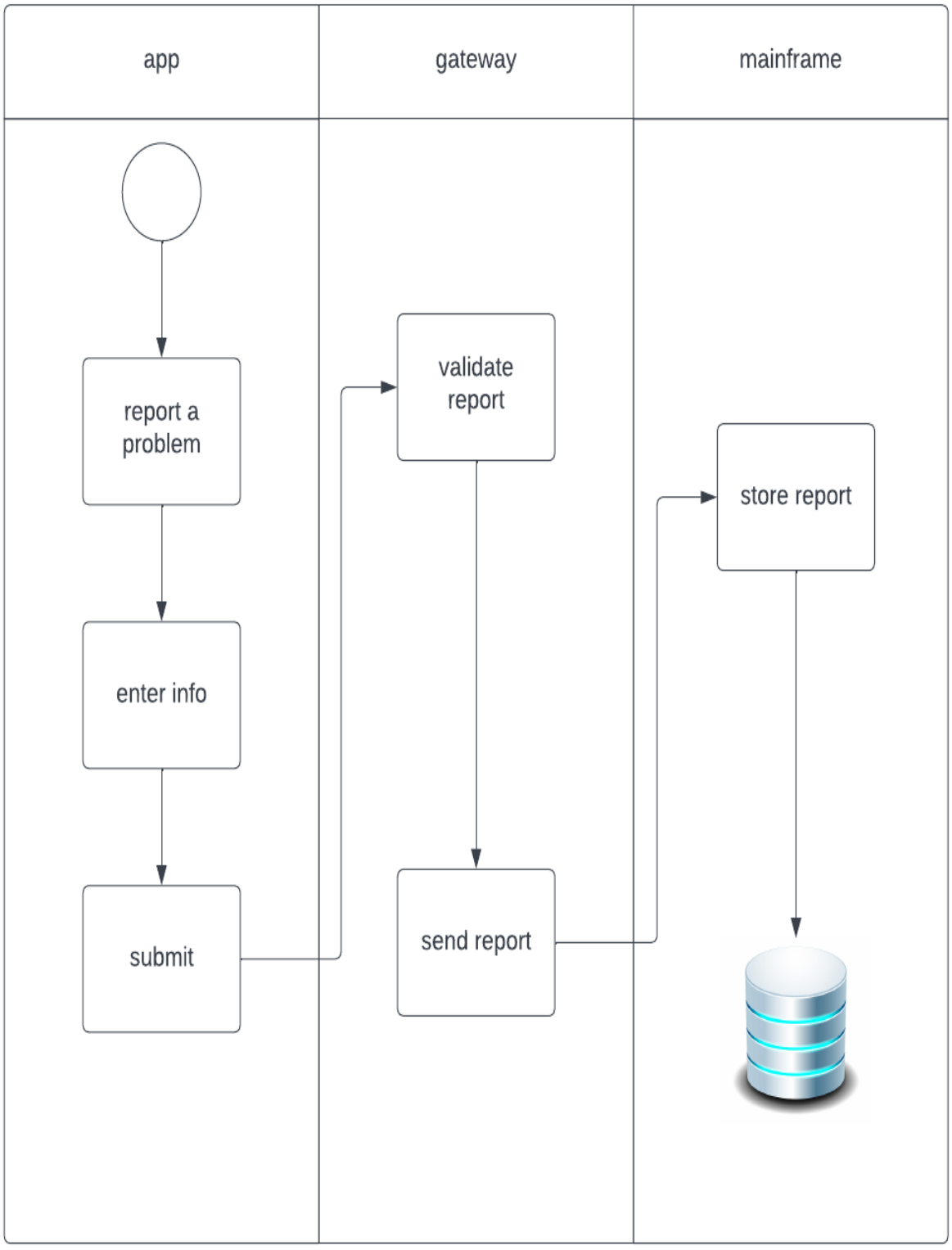


Activity Diagram:

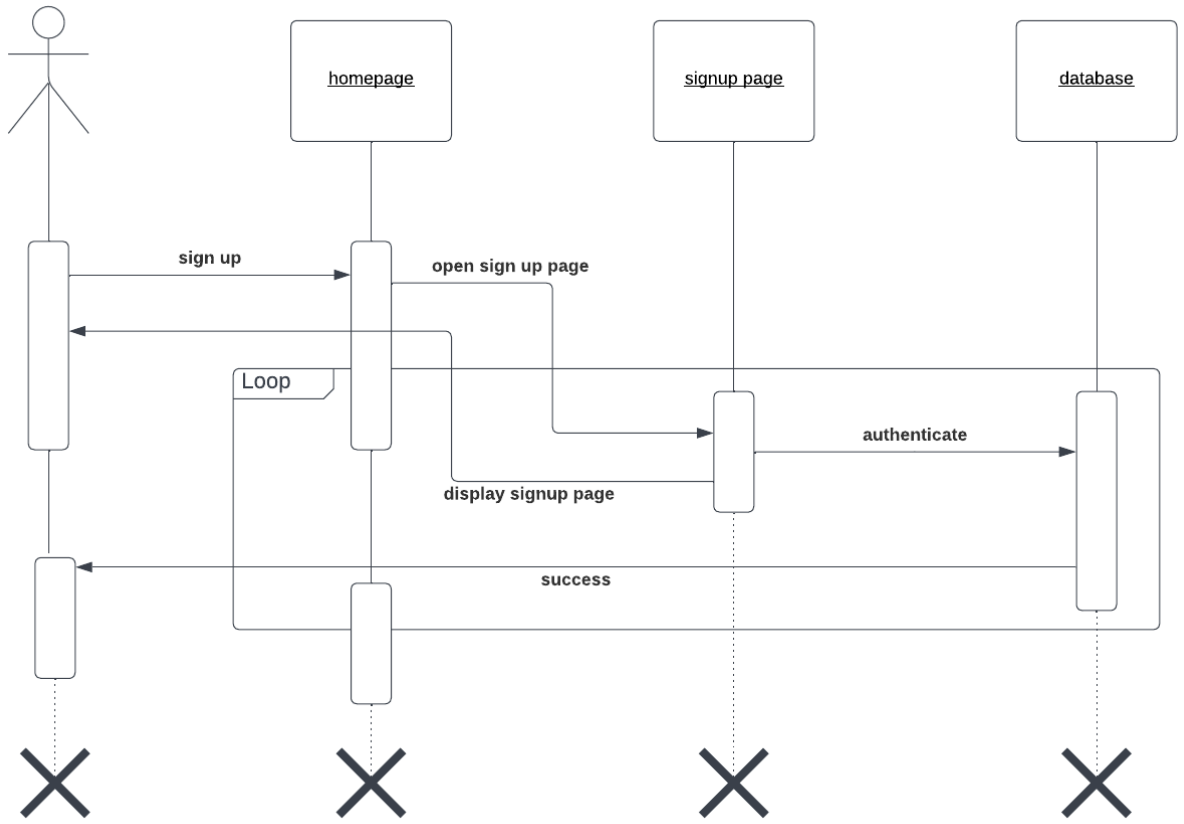


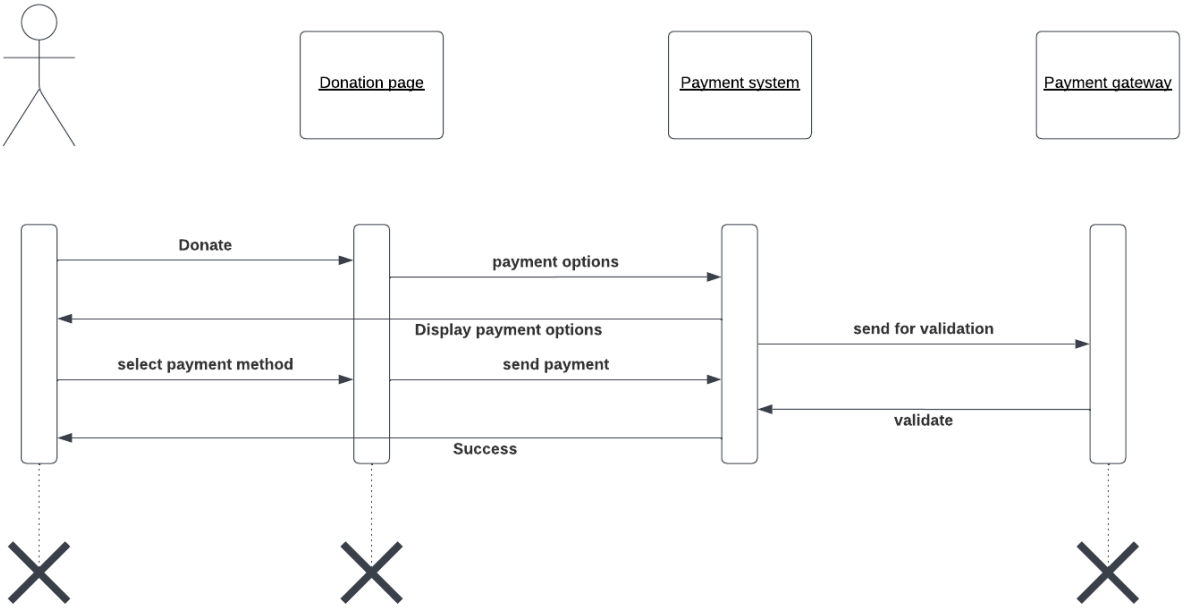

sign up page
homepage
mainframe
Enter information
Submit
Account created successfully
Login
View account info
Edit account
Edit account
Make changes


app
gateway
mainframe
Look for safespot
search for closest safespot
show location of safespot
confirm safespot
show closest safespot

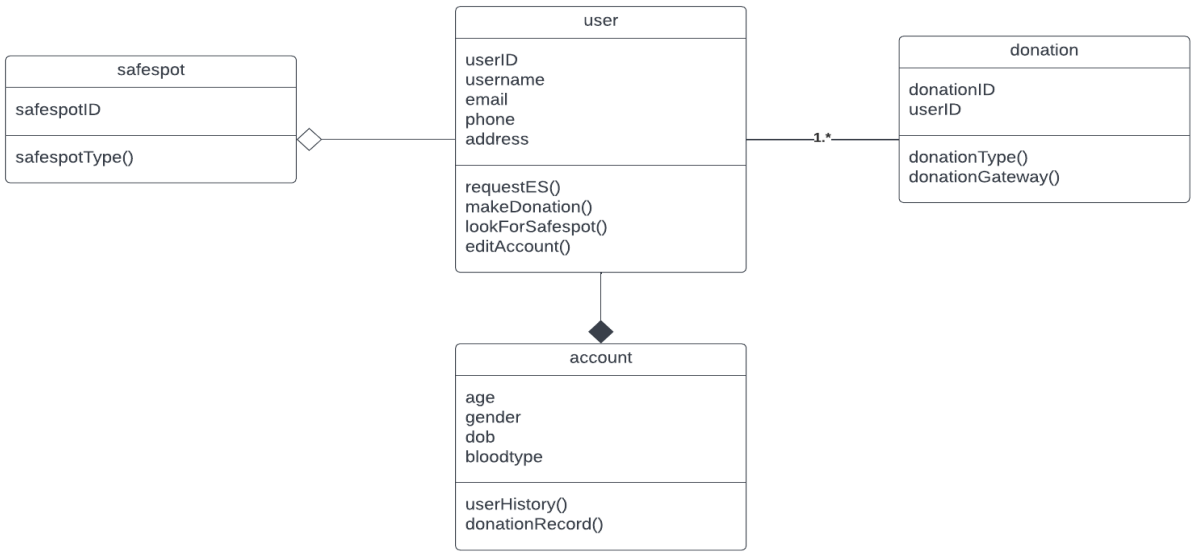

app
gateway
mainframe
report a problem
enter info
submit
validate report
send report
store report

Sequence Diagram:

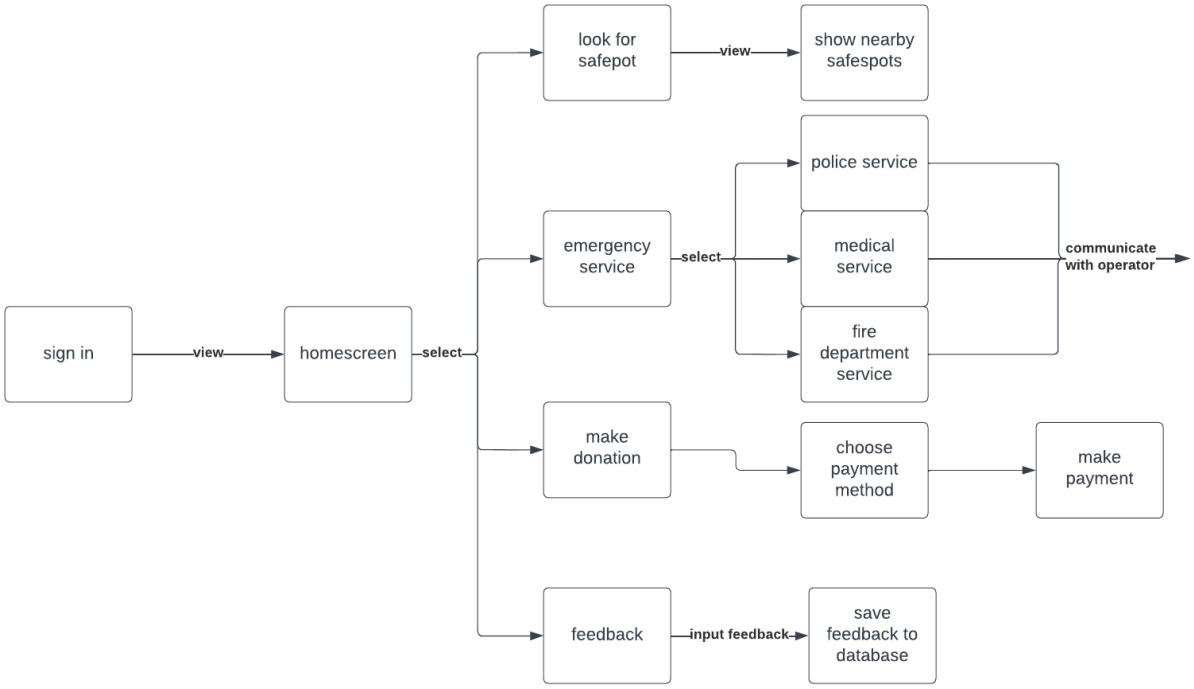




Class Diagram:



State-chart Diagram:



CRUD matrix:

Create Account:

Activity	User	Employee	Admin
input personal information	C		
check account	R	R	
confirm	C		

Report a problem:

Activity	User	Admin	Mainframe
Enter report	C		U
Check report	R	R	
Confirm	C		U

Section 5:

Structure English pseudo code for the system:

Login:
GET email
GET password

If(email and password == true) THEN
Login successful
Else
Show failed prompt

Create Account:
ENTER email
ENTER password
ENTER name
ENTER phone
ENTER location
SUBMIT details
ENTER
SHOW success prompt
ELSE
SHOW failed prompt

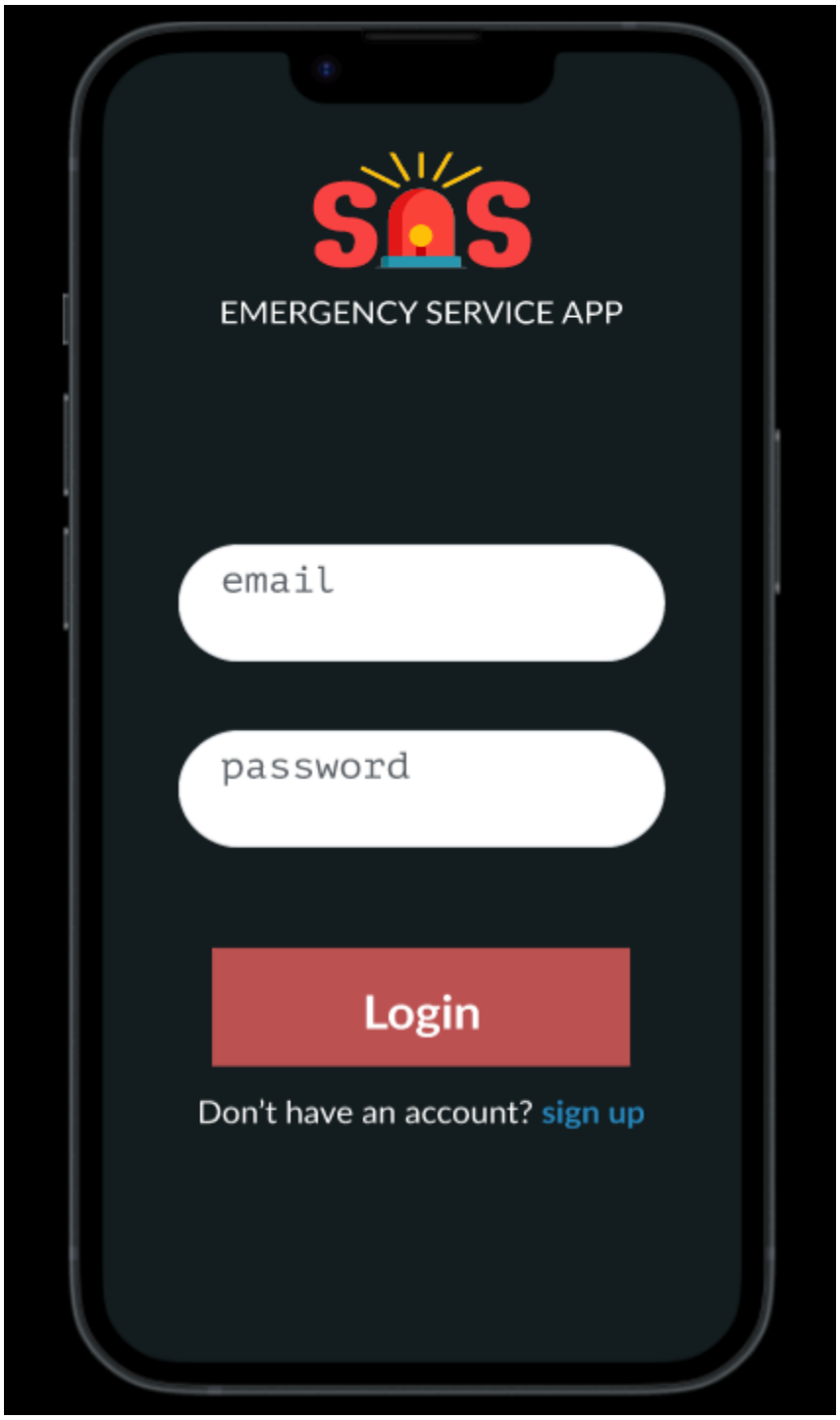
Report a problem:
ENTER report in textfield
GET textfield
STORE textfield contents
SHOW success prompt

Safespot:
GET userLocation
GET safespotLocation
RUN Dijkstra's algorithm between two userLocation and allSafespotLocation
SHOW shortest path as nearest safespot

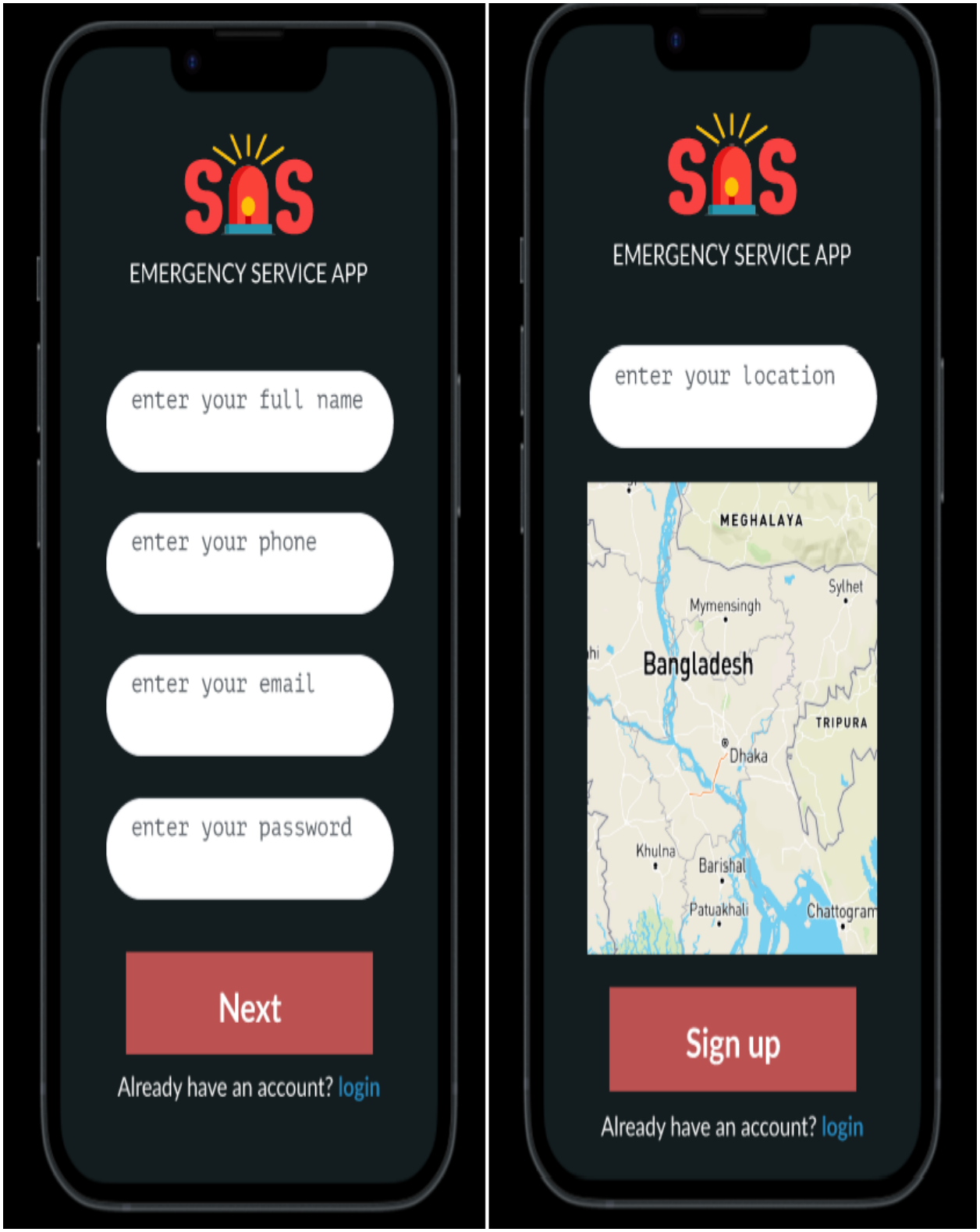
User Interface Prototype:

https://www.figma.com/file/9TdaFWZ8n5TshRDQxejcZ5/SOS?t=hbiMzlkuYK6f7RAm-1

Login screen:



Sign up screen:

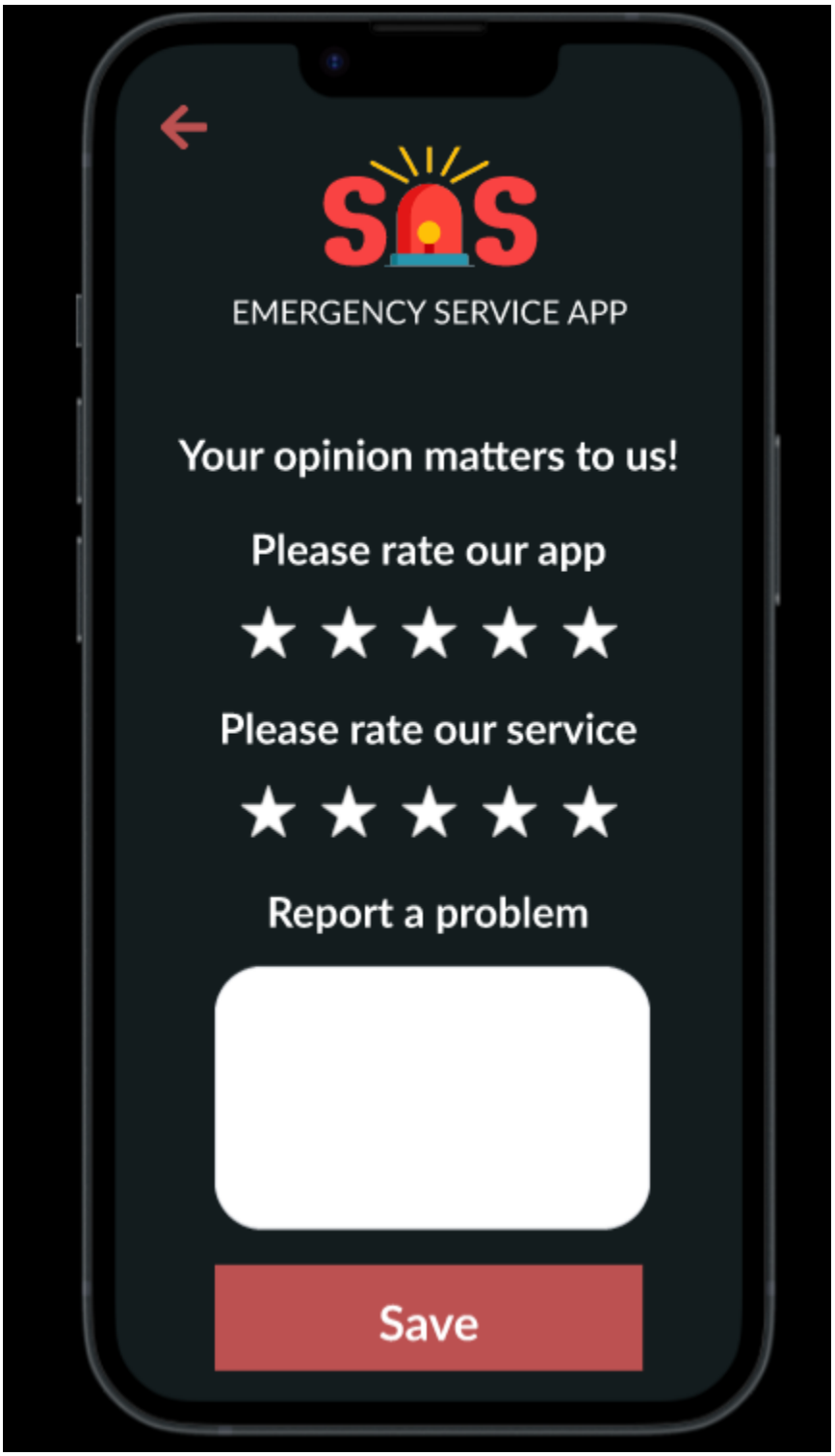


Home screen:


SOS
EMERGENCY SERVICE APP
Emergency services
Look for Safespot
Feedback
Log out

Safespot screen:

Feedback screen:



Emergency service screen:

Communication screen: