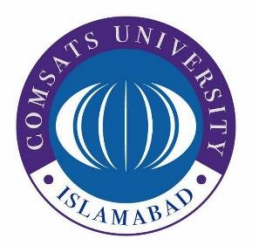


COMSATS University Islamabad (CUI)

Software Design Description
(SDS DOCUMENT)

for

Child Monitoring System (CMS)

Version 1.0

By

Saif ur Rehman **CIIT/FA20-BSE-092/ISB**

Ahmad Mahmood **CIIT/FA20-BSE-096/ISB**

Supervisor

Dr. Javed Iqbal

Bachelor of Science in Software Engineering (2020-2024)

Table of Contents

1. Introduction 3
2. Design Methodology and Software Process Model 3
2.1 Design Methodology 3
2.2 Software Process Model 4
3. System Overview 4
3.1 Architectural Design 4
3.1.1 Modular decomposition 4
3.1.2 System Architecture 6
4. Design Models 7
4.1 Activity Diagram 7
4.1.1 Login 7
4.1.2 Get the current location of the child 8
4.1.3 Listen to voice commands 9
4.1.4 Send SOS notification 10
4.1.5 Monitor child's phone activity 11
4.1.6 Generate reports 12
4.1.7 Send and receive alert messages 13
4.2 Class Diagram 14
4.3 Sequence Diagram 15
4.3.1 Get the child's current location 15
4.3.2 Set fenced zone 16
4.3.3 The child entered or exited the fenced zone 17
4.3.4 Send alert or SOS notification 18
4.3.5 The child speaks the "HELP" command 19
4.4 State Transition Diagram 20
4.4.1 SOS and alert notifier 20
4.4.2 The child says "HELP" 20
4.4.3 Low battery alert 21
4.4.4 Fenced zone entrance and exit 21
4.4.5 Monitoring reports requested 22
5. Data Design 23
5.1 Data Dictionary 23
5.1.1 Users 23

5.1.2 VisitedLocations 23
5.1.3 Calls 23
5.1.4 Messages 24
5.1.5 Contacts 24
6. Human Interface Design 25
6.1 Screen Images 25
6.1.1 Login 25
6.1.2 Set fences 26
6.1.3 See Current location 27
6.1.4 Send Alert Message 28
6.1.5 Set Profile 29
6.2 Screen Objects and Actions 30
6.2.1 Map view 30
6.2.2 Alert button 30
6.2.3 Report view 30
6.2.4 Set Area button 30
6.2.5 Login button 30

1. Introduction

Child Monitoring System (CMS) is an android application that lets the parents/guardians monitor their child without the physical presence of the parents. The CMS is a smart software solution that frequently updates the parents about their child and lets them take necessary action based on the situation the child currently is in.

The system is divided into two parts:

i. **Child Part** – The child part contains the functionalities to monitor the child's activities, and send the monitored data to the CMS app on the parent's phone.

ii. **Parent Part** – The parent part of the CMS is for the parent's phone, through which the parent could analyze the data that the child part of the CMS sends to the parent.

The CMS performs several functionalities, some of which are continuously being operated in the background of a child's phone e.g. location tracker, while some of which are executed only when required, e.g. notifying emergency contacts in case of an emergency. The main features or modules of the Child Management System are:

- User authorization
- GPS location tracker
- Voice command listener
- SOS notification sender
- Geo-fencing child's location
- Child's phone activity monitoring
- Maintaining activity history
- Report generation
- Sending and receiving alerts

2. Design Methodology and Software Process Model

2.1 Design Methodology

The design methodology used in this project is **Object Oriented Programming (OOP)**. The reason for choosing OOP is because it provides a more structured and modular approach to system development. As the system is comprised of several different modules, it is necessary to separate the functionality of each module from the others, and just provide an interface for their intercommunication.

An advantage of the modular approach of OOP is that it allows the developers to easily scale and extend the system in the future to add more functionality to the system. So in case, the parent decides to add another activity to be monitored, it could easily be incorporated into the existing software project. OOP also works very well with Graphical User Interface (GUI), as the elements being displayed on the screen are objects, that could invoke particular functionality when the user interacts with them. Hence OOP is more suitable for this project than procedural programming, as it could become very hectic to maintain a large project that has been implemented using procedural programming.

2.2 Software Process Model

There are several different requirements that parents would want a child monitoring system to perform, so therefore the developers must work on each functionality separately, and start the implementation of the next functionality after one has been completed. For this reason, the software process model for this project is the **Incremental model**.

The CMS is a modularized system, therefore the iterative approach of software development will make it easier to develop the system, by implementing each module with successive builds of the system. The first step in the incremental process model is to identify and prioritize the modules for development. Modules with higher priority, e.g. location tracking and emergency notification system will be implemented in the initial builds, while modules with lower priority e.g. report generation will be implemented in later builds. With each successive build of the system, the functionality could be presented to the parent along with a prototype of the latest build of the system.

3. System Overview

3.1 Architectural Design

3.1.1 Modular decomposition

After modular decomposition of the system, the following modules as listed in section 1 of the document are following:

- **User authentication** – For letting the users login into the system, and saving their respective data backups on the server
- **GPS location tracker** – For tracking the location of the child, and sending it to the parent
- **Voice command listener** – This module continuously listens to the microphone of the child's phone to send emergency alerts in case of the "HELP" command
- **SOS notification sender** – For sending emergency state alerts to parents, and other specified emergency contacts

- **Geo-fencing child's location** – This module continuously checks the child's location, and sends an alert to the parent in case the child enters or leaves a geofenced zone
- **Child's phone activity monitoring** – This module monitors a child's phone activities, including calls, text messages, contacts, and location history
- **Maintaining activity history** – This module saves the monitored data on the child's phones, and then sends them to the parent's phone
- **Report generation** – This module generates reports of a child's activity, which could then be viewed and downloaded by the parent
- **Sending and receiving alerts** – This module sends alerts to the parent that are less critical than SOS notifications. The alerts sent by this module include sending alerts for low battery, entering and leaving the geofenced zone, etc.

Among the above-listed modules, certain modules have some sort of relationship with other modules. The relationships are described below, along with graphical representation in a box and line diagram.

3.1.1.1 Overview of modules' collaboration:

The user authentication module authenticates and signs in the user into the system, after which the activity monitoring of the child's phone starts. The monitoring includes fetching the current location of the child through GPS, implementing geo-fencing alerts, and sending SOS notifications in case of an emergency, or through the voice command of "Help". The monitoring module also maintains the logs of the child's activities on the phone, including phone calls, messages, contacts, and location history. Reports of this data could then be viewed or downloaded by the parent.

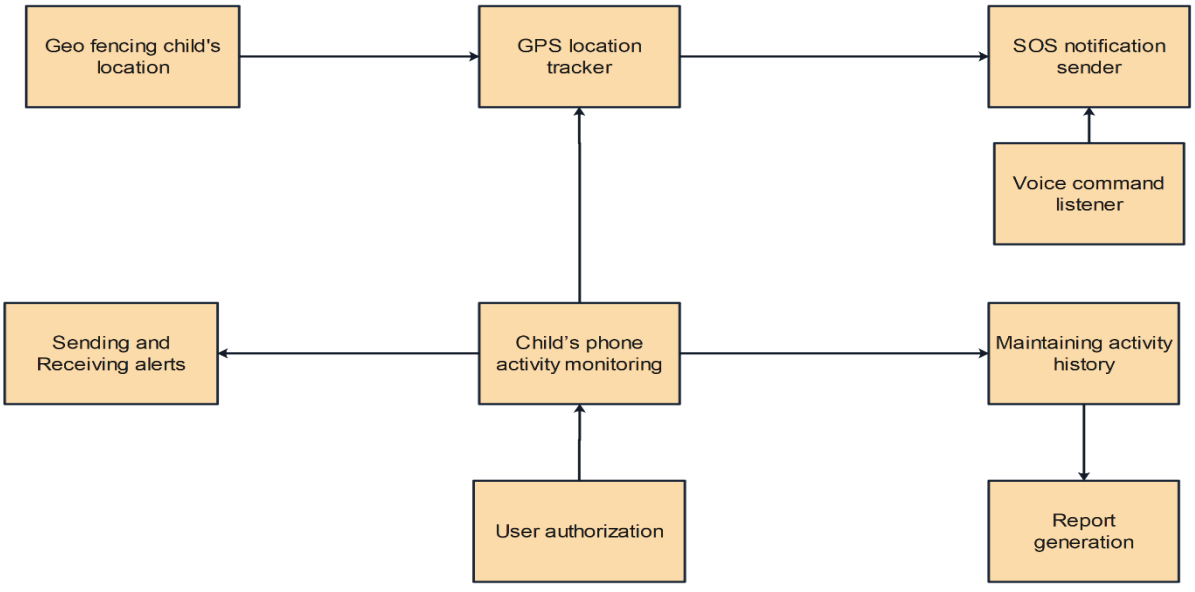


Figure 1 Box and Line diagram

3.1.2 System Architecture

The system will be developed based on the **Client-Server Architecture**. The primary reason for choosing client-server architecture is because there are two parts of the system, which need to communicate with each other and share data. The child part of the CMS will collect and upload data to a server, which will be downloaded into the parent's phone which could then be viewed by the parent.

The child and parent could not be connected all the time if they are not connected to the internet. Therefore, it is necessary for the child part of the system to first send the data to an online database server. Until the parent is not connected to the internet, that data will be safely stored in the database, and when the parent connects to the internet, the parent part of the CMS will look for available data in the server and will download it. Hence for this purpose, client-server is the most suitable architecture for the system.

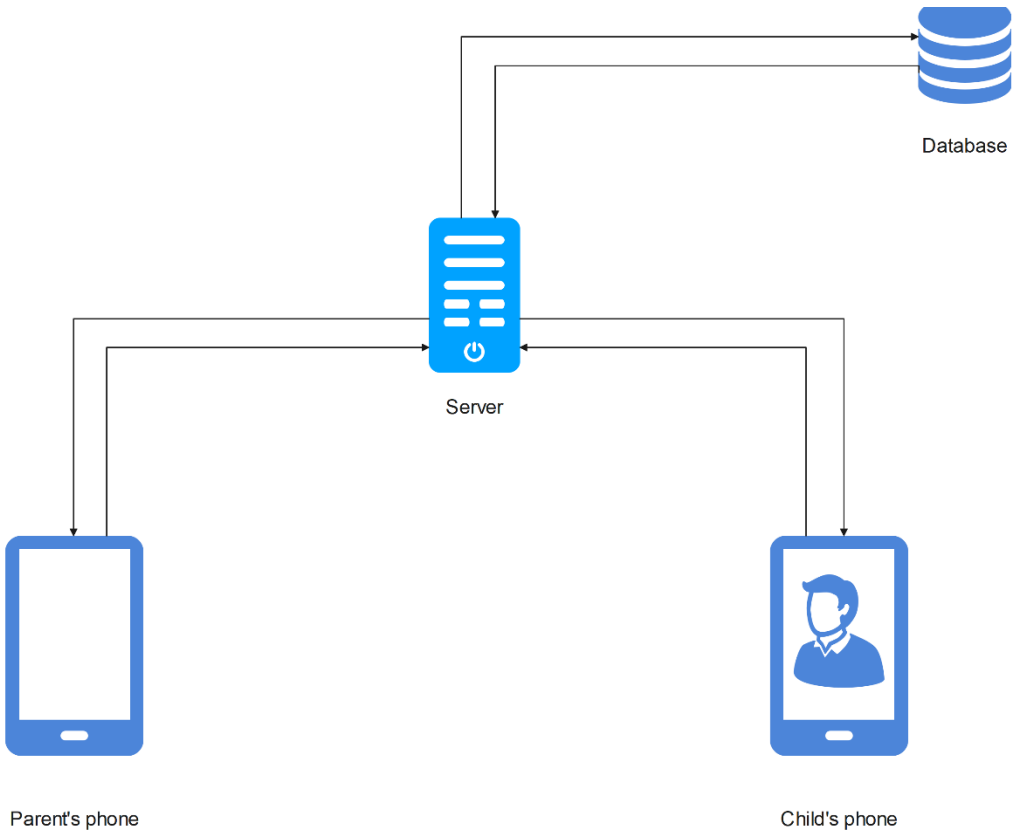
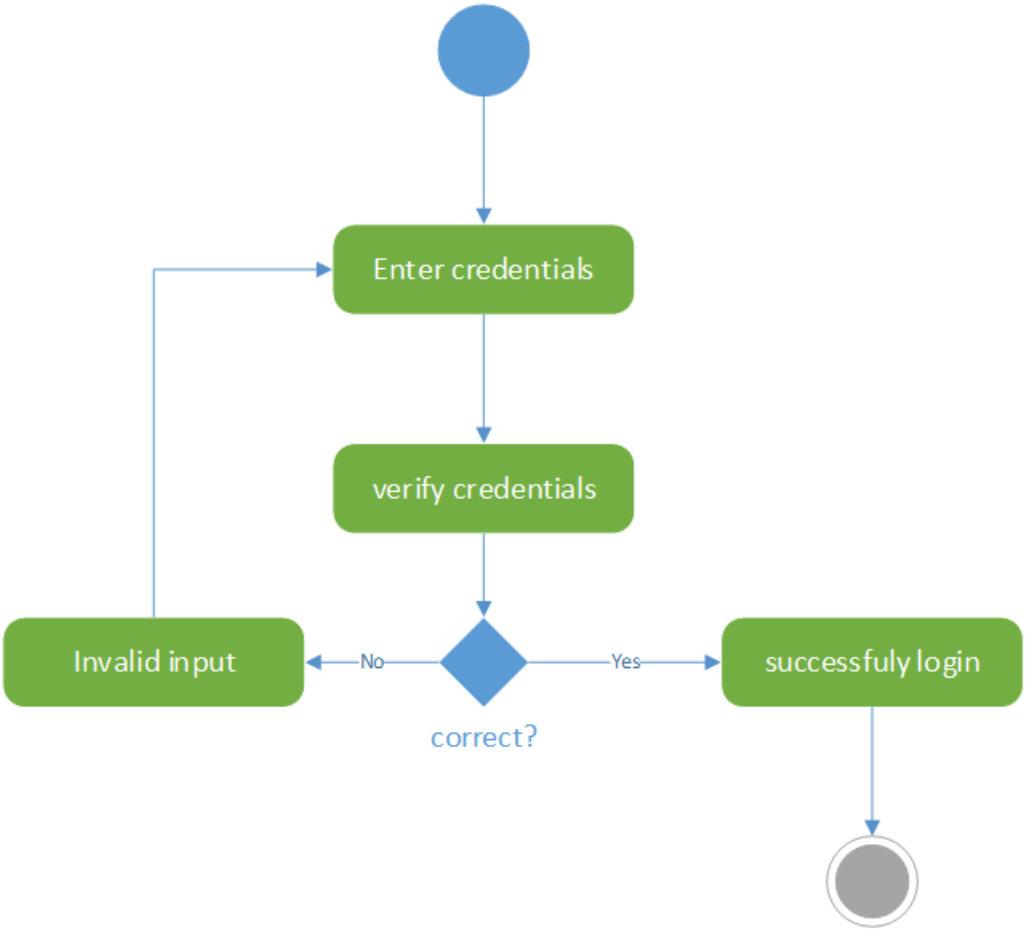


Figure 2 Architecture Diagram

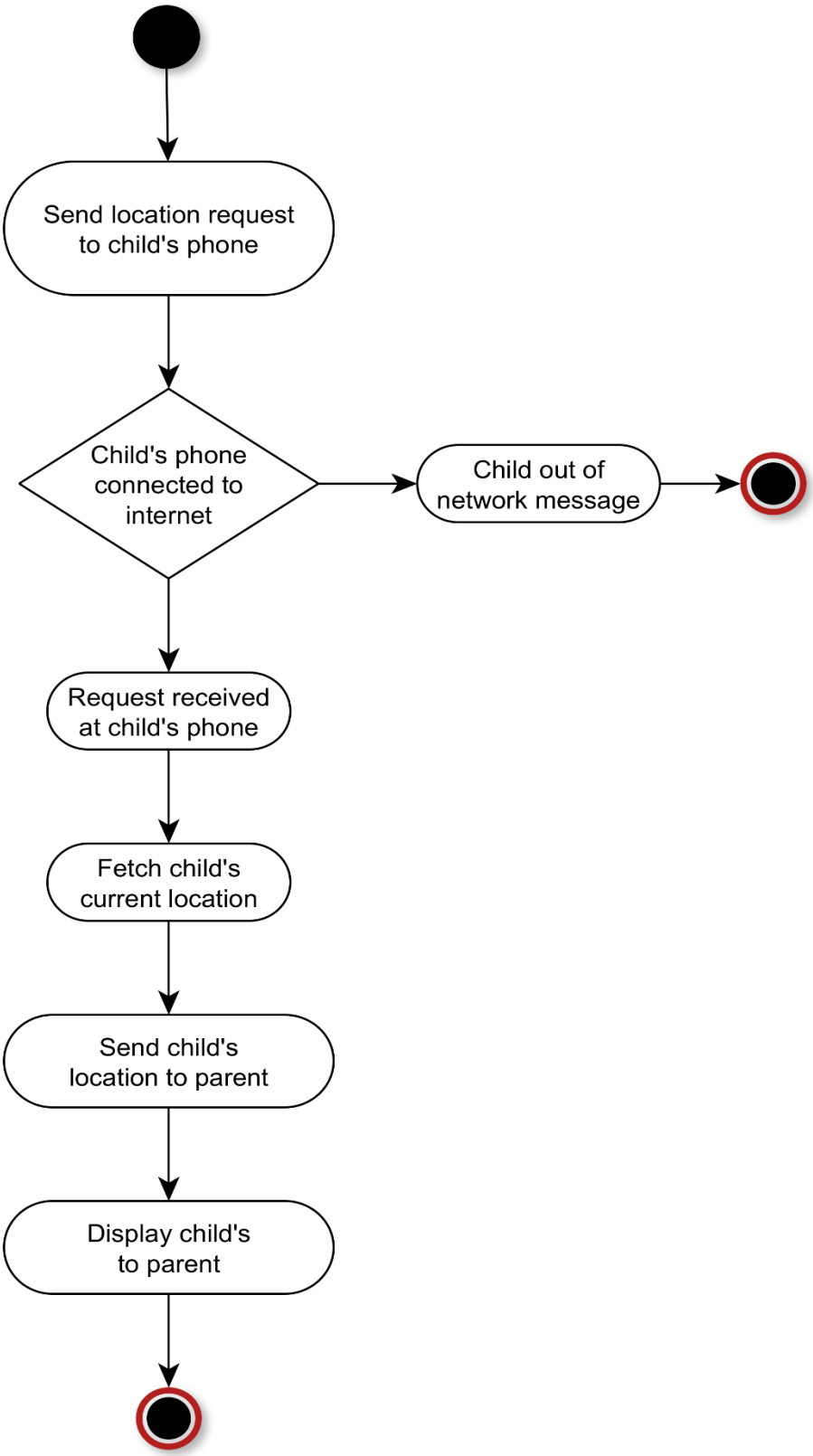
4. Design Models

4.1 Activity Diagram

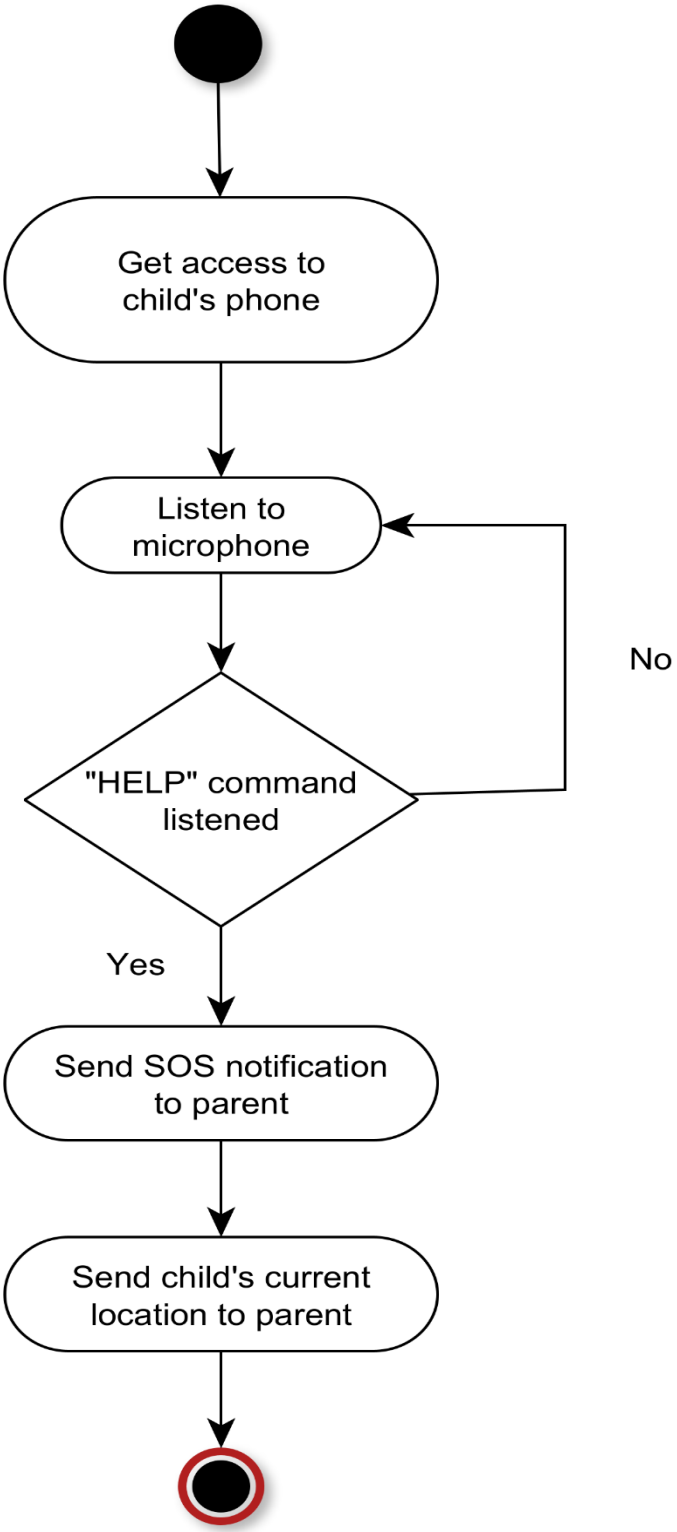
4.1.1 Login



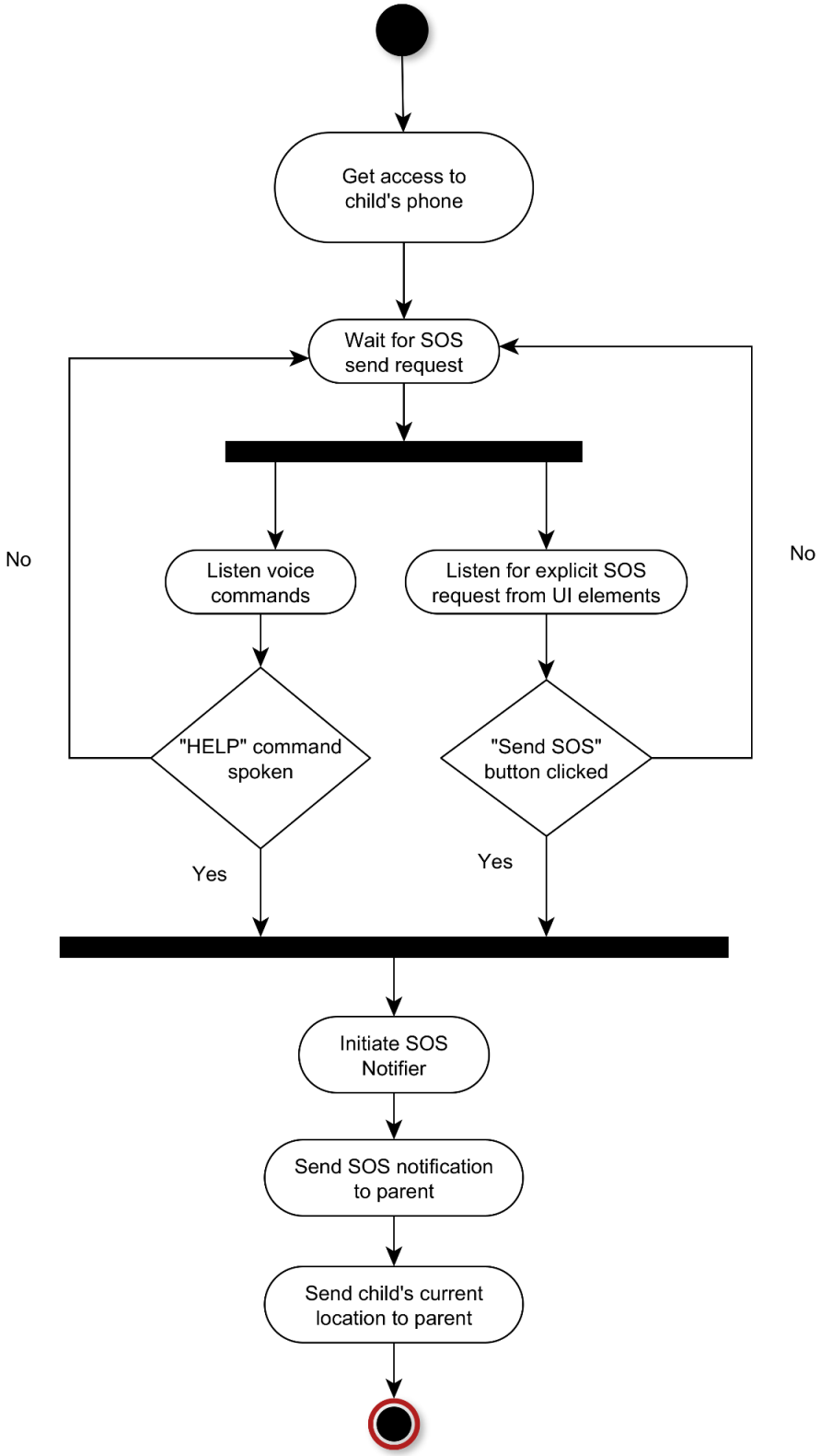
4.1.2 Get the current location of the child



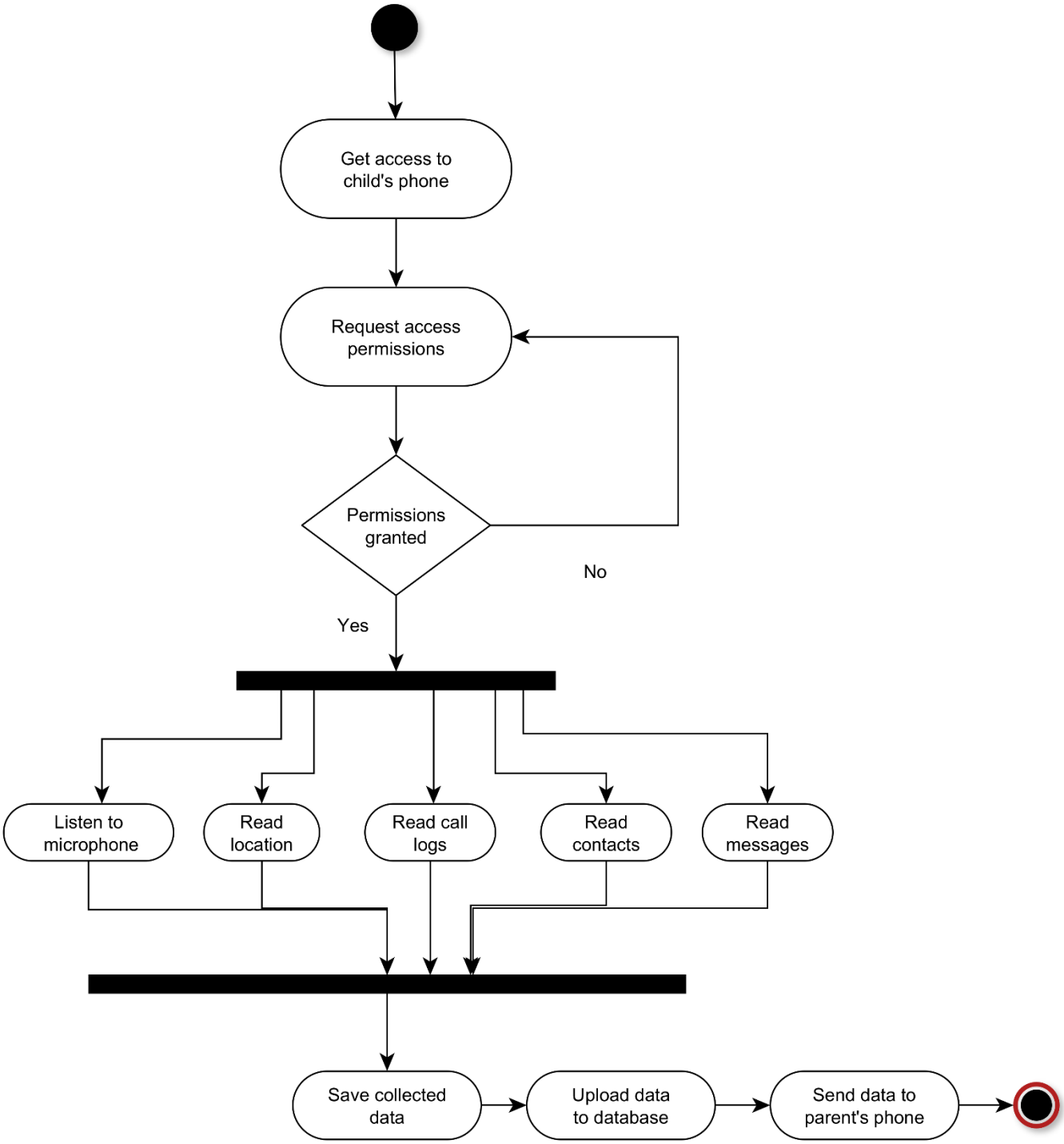
4.1.3 Listen to voice commands



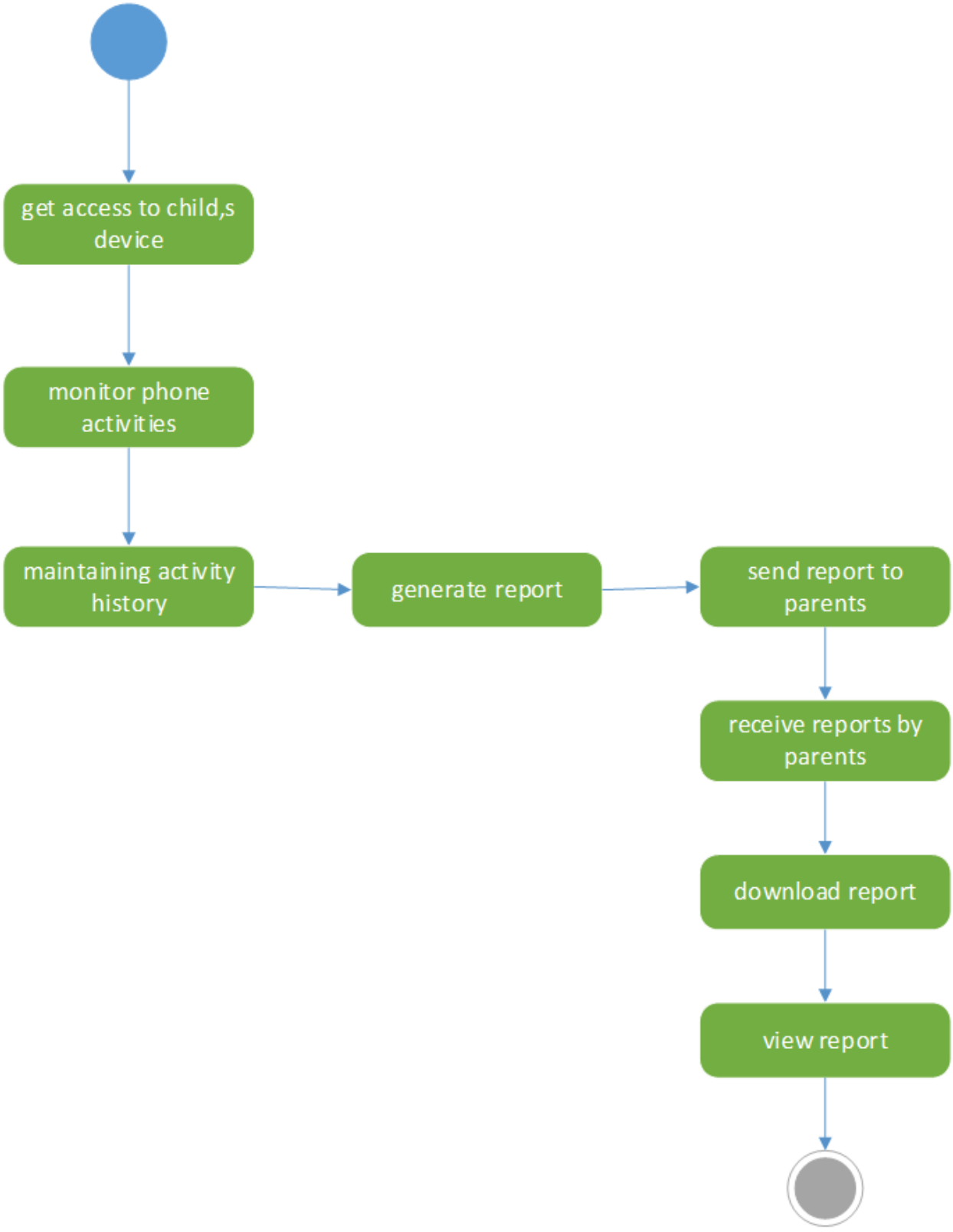
4.1.4 Send SOS notification



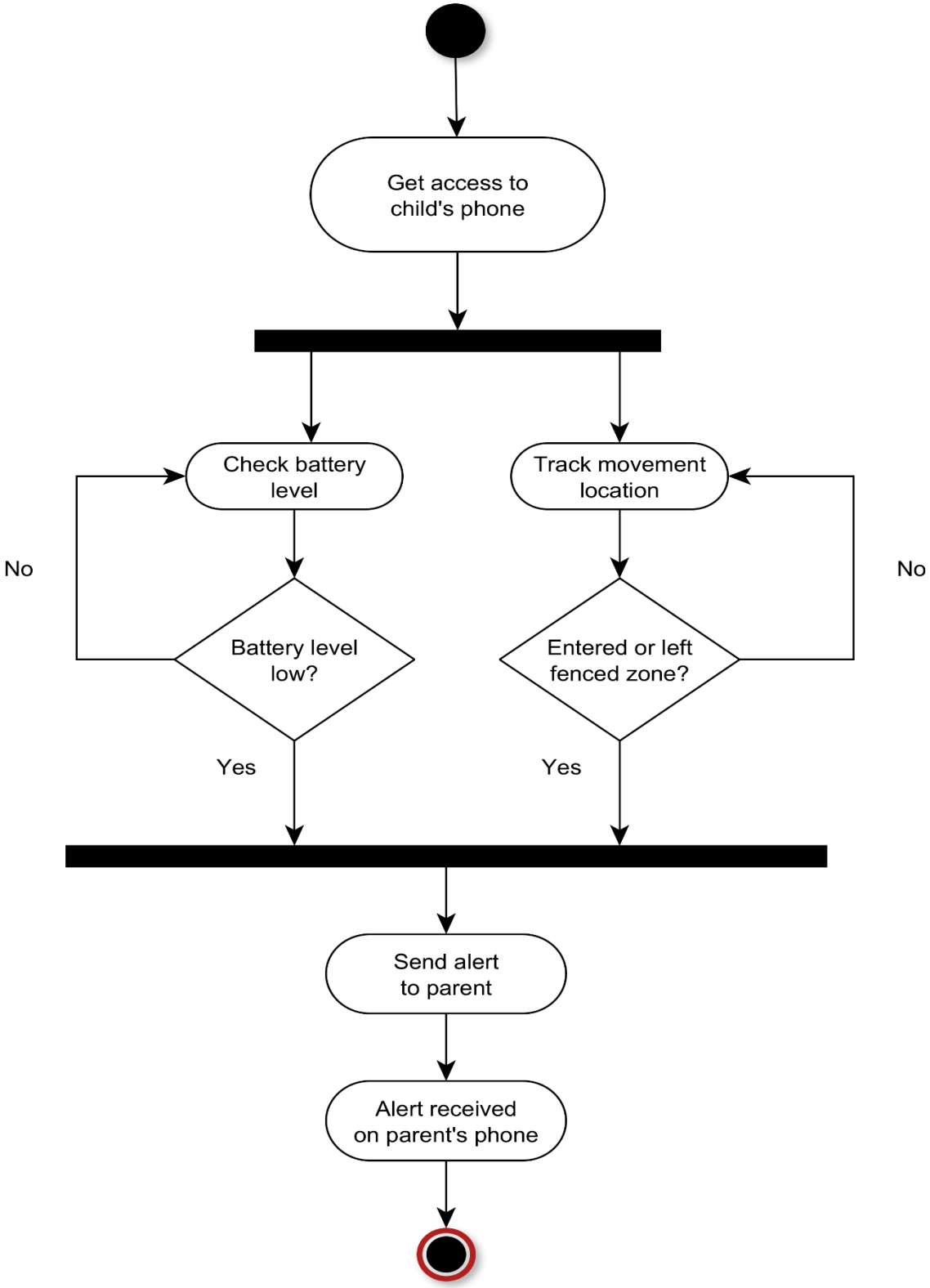
4.1.5 Monitor child's phone activity



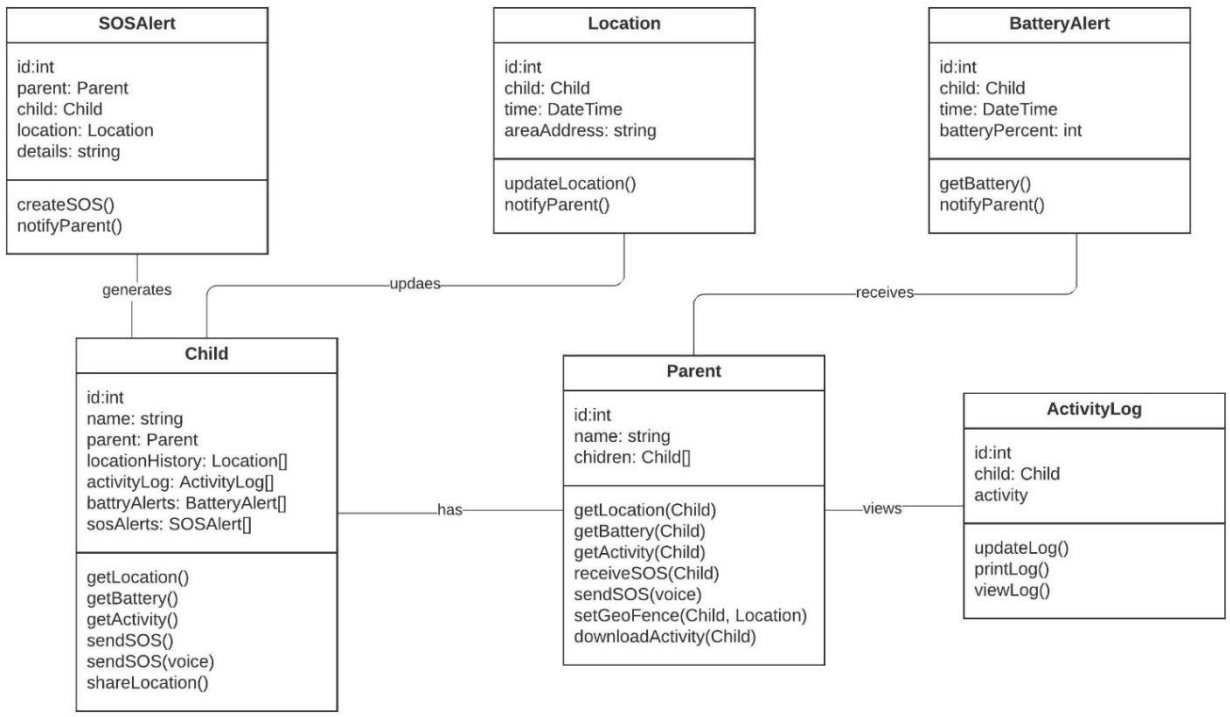
4.1.6 Generate reports



4.1.7 Send and receive alert messages

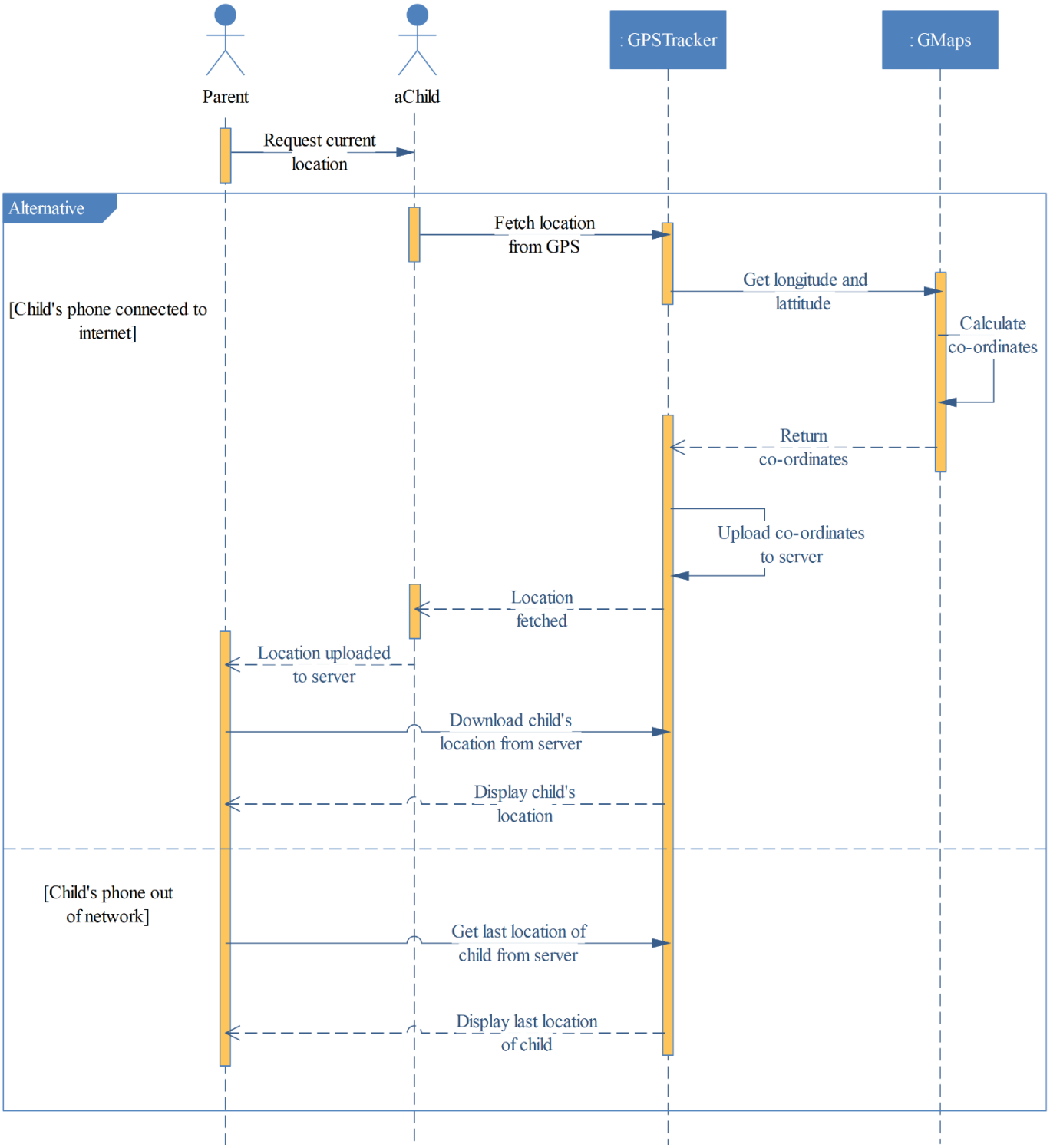


4.2 Class Diagram

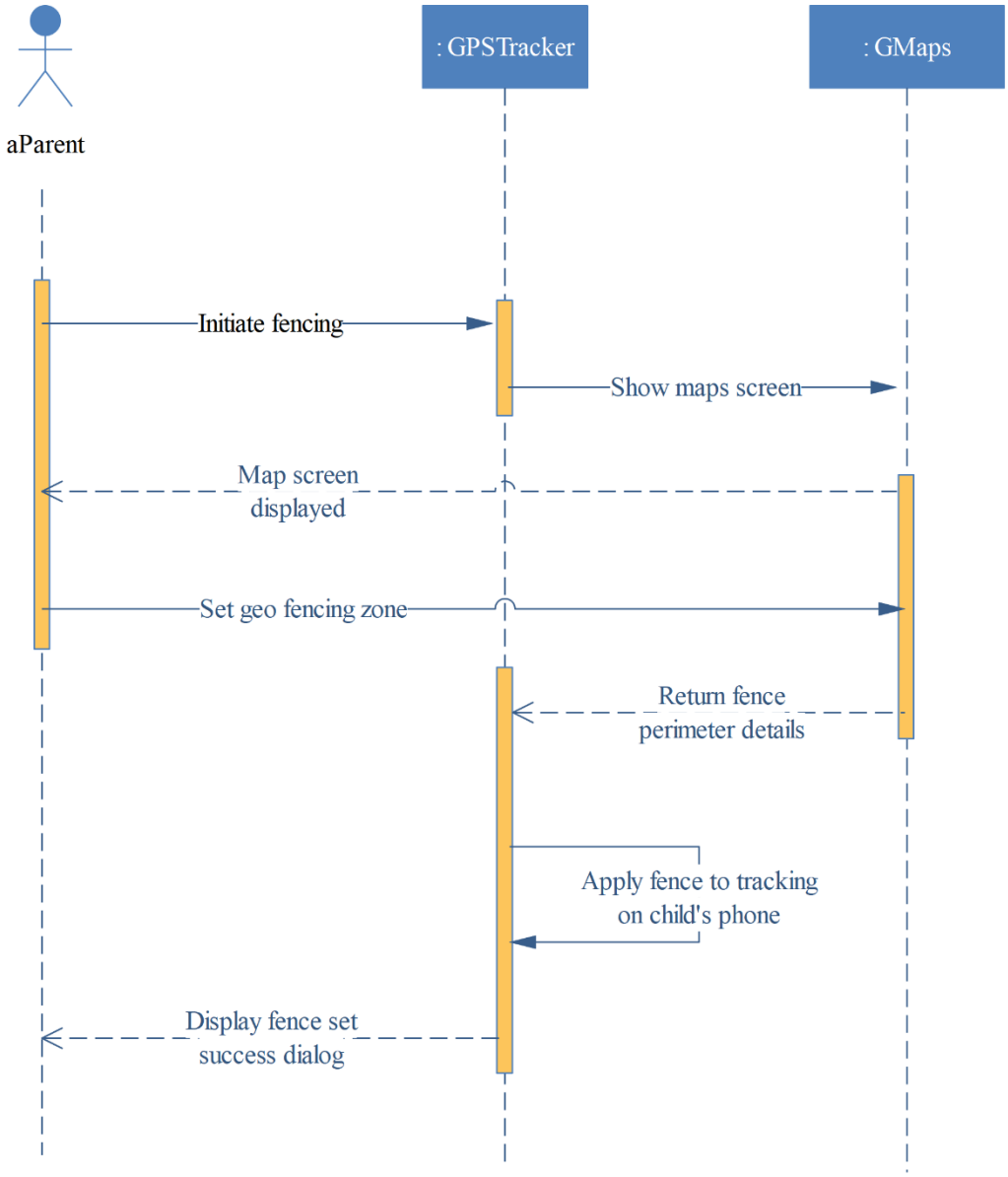


4.3 Sequence Diagram

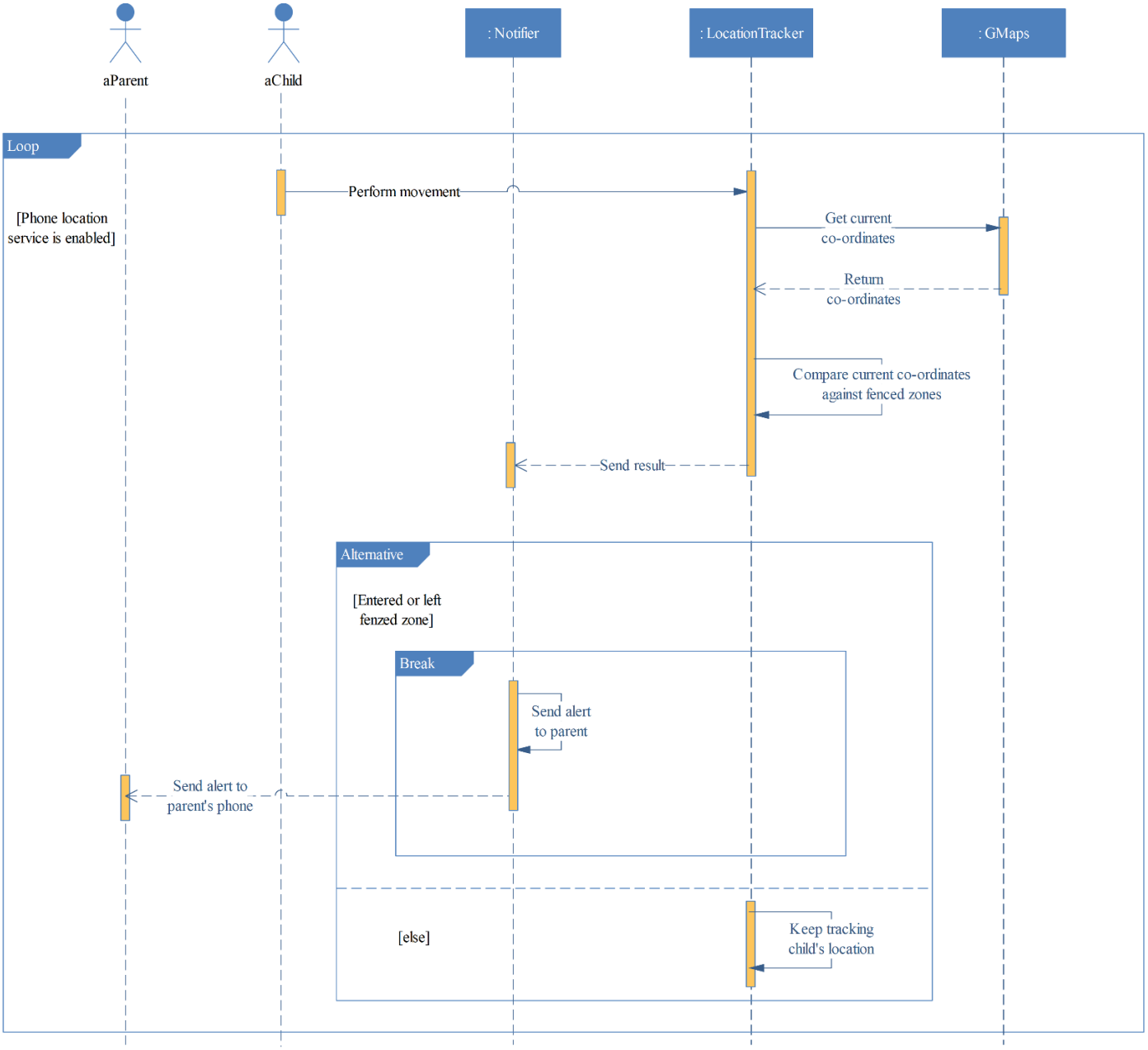
4.3.1 Get the child's current location



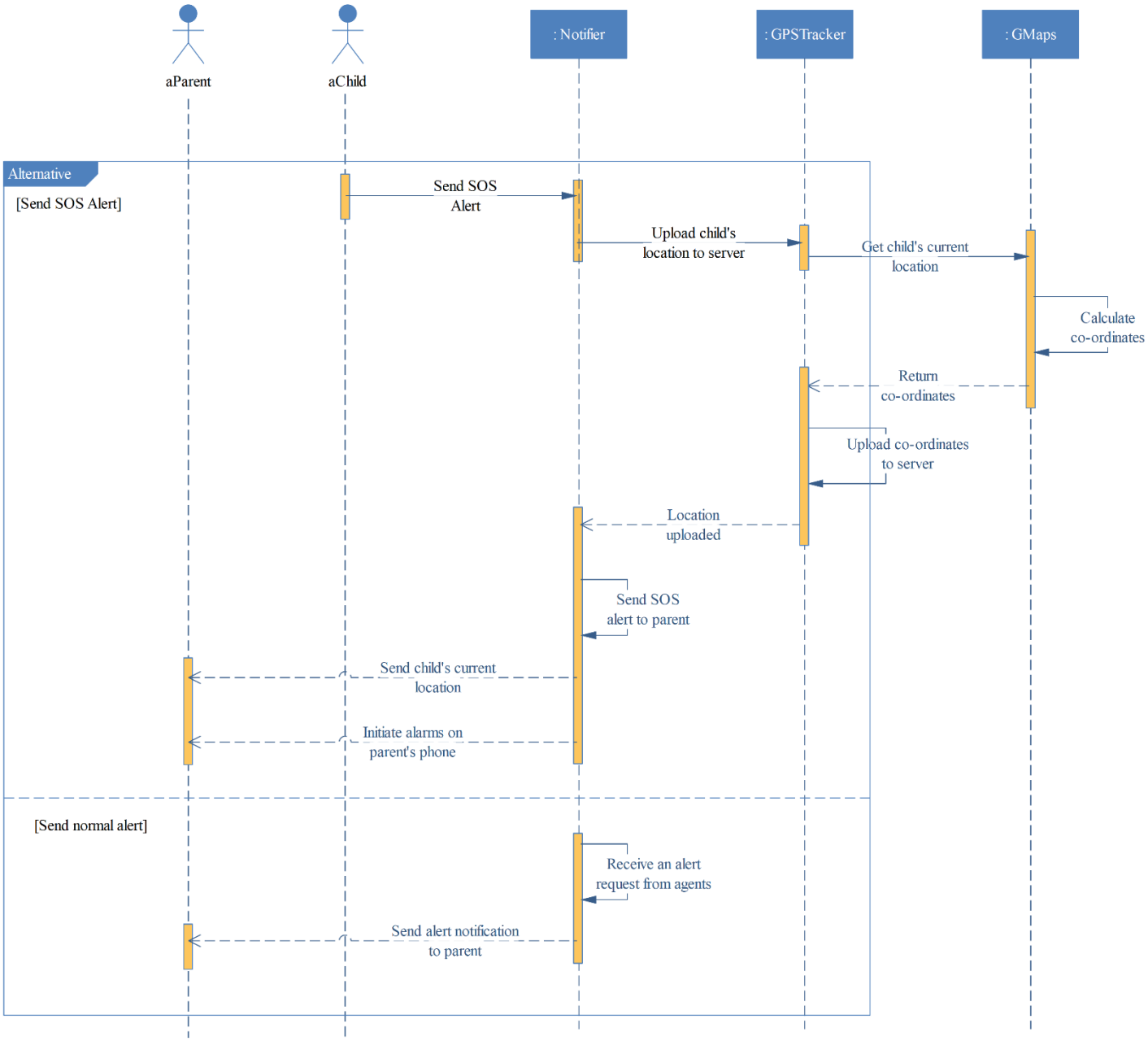
4.3.2 Set fenced zone



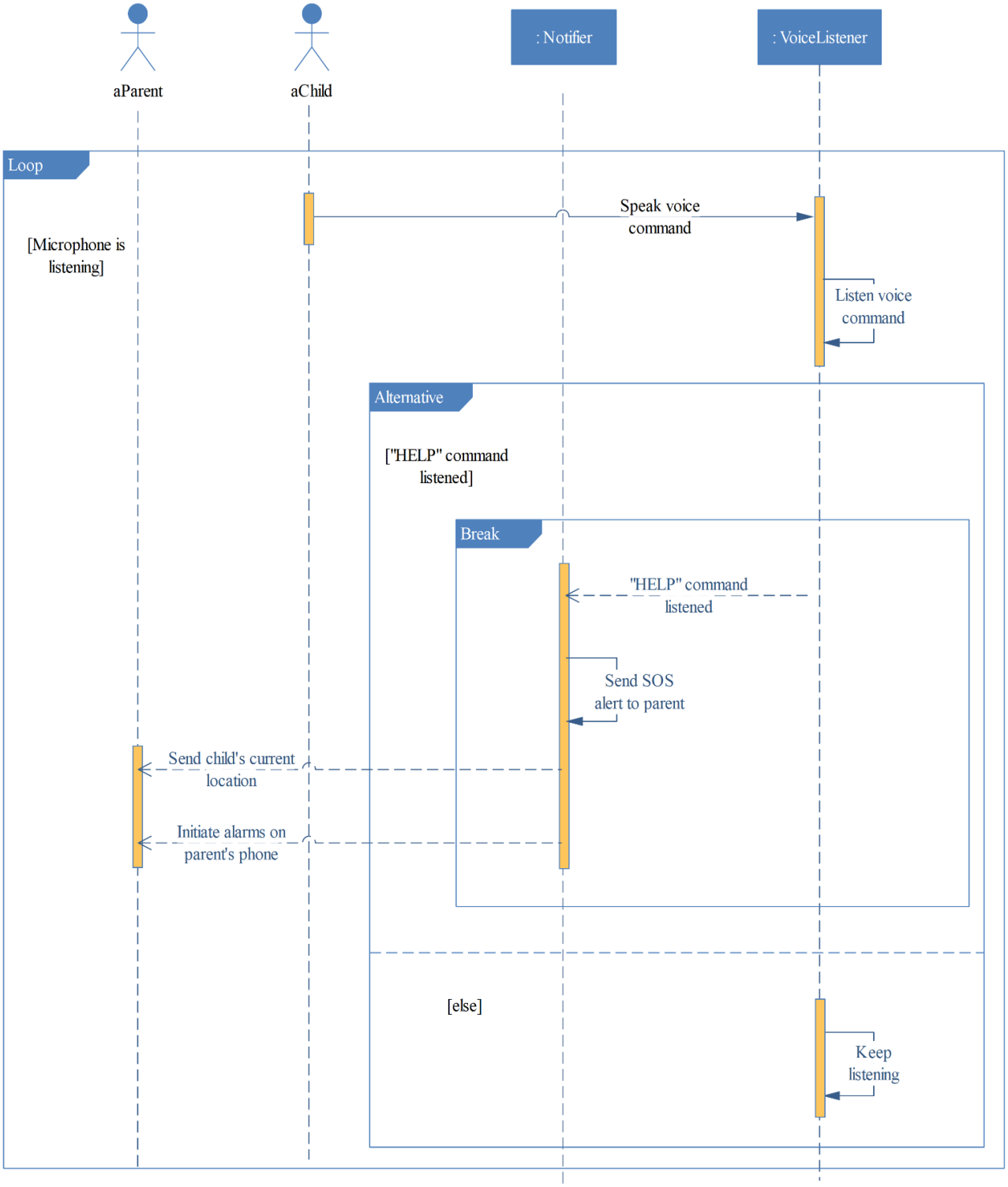
4.3.3 The child entered or exited the fenced zone



4.3.4 Send alert or SOS notification

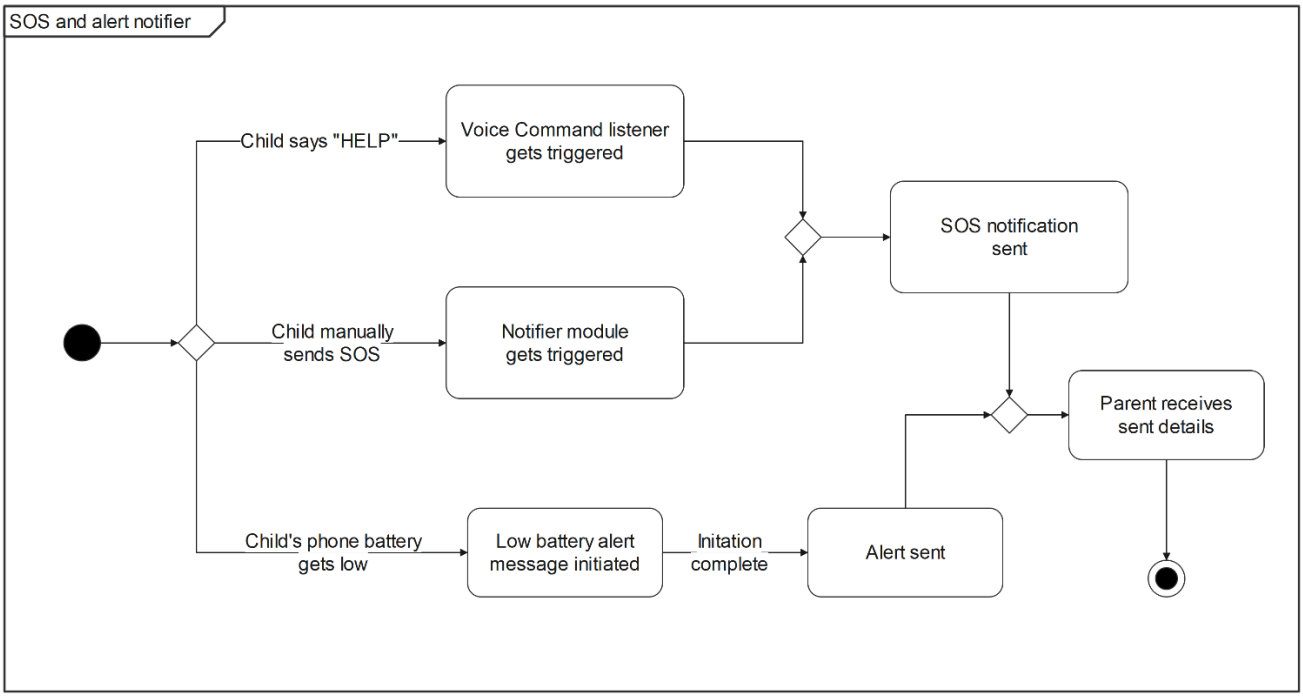


4.3.5 The child speaks the “HELP” command

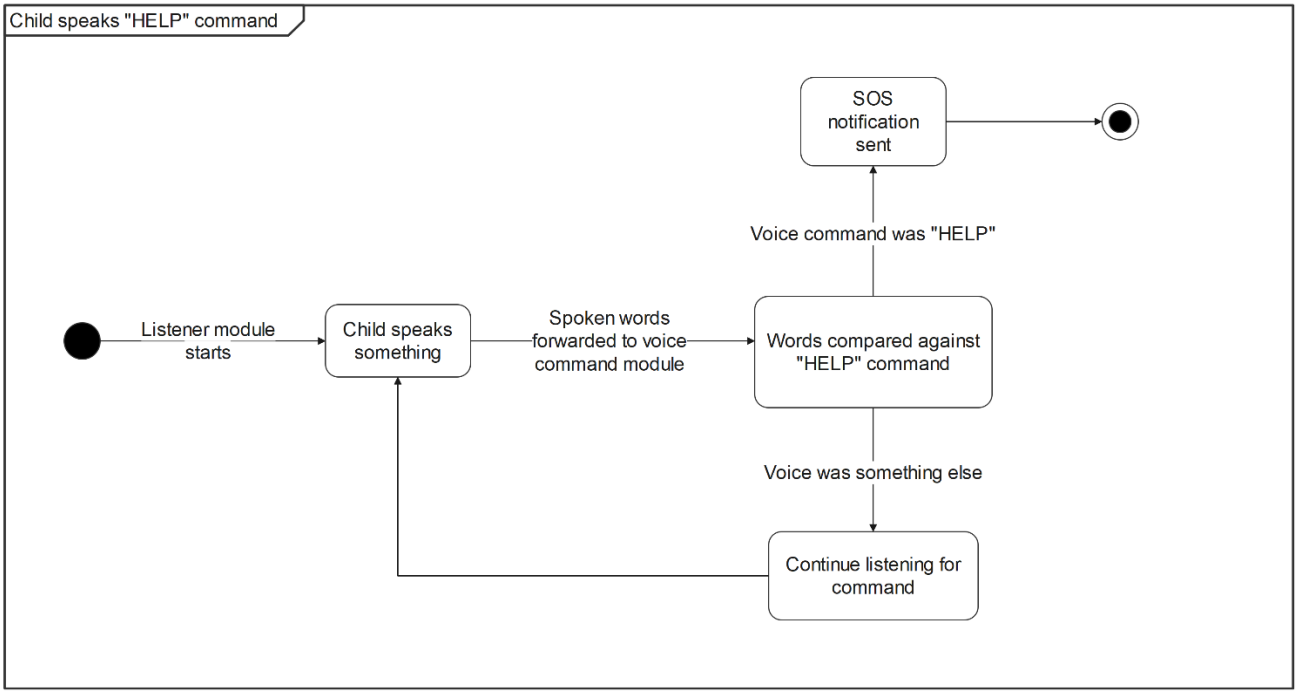


4.4 State Transition Diagram

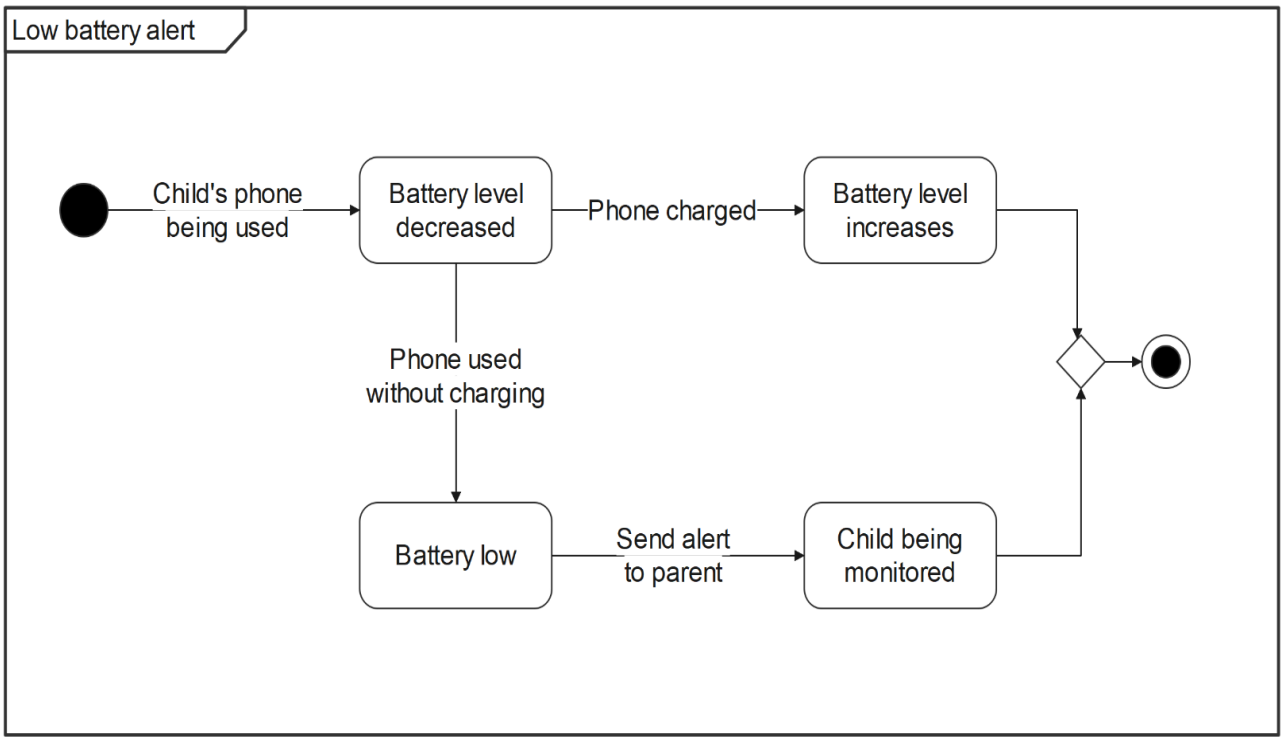
4.4.1 SOS and alert notifier



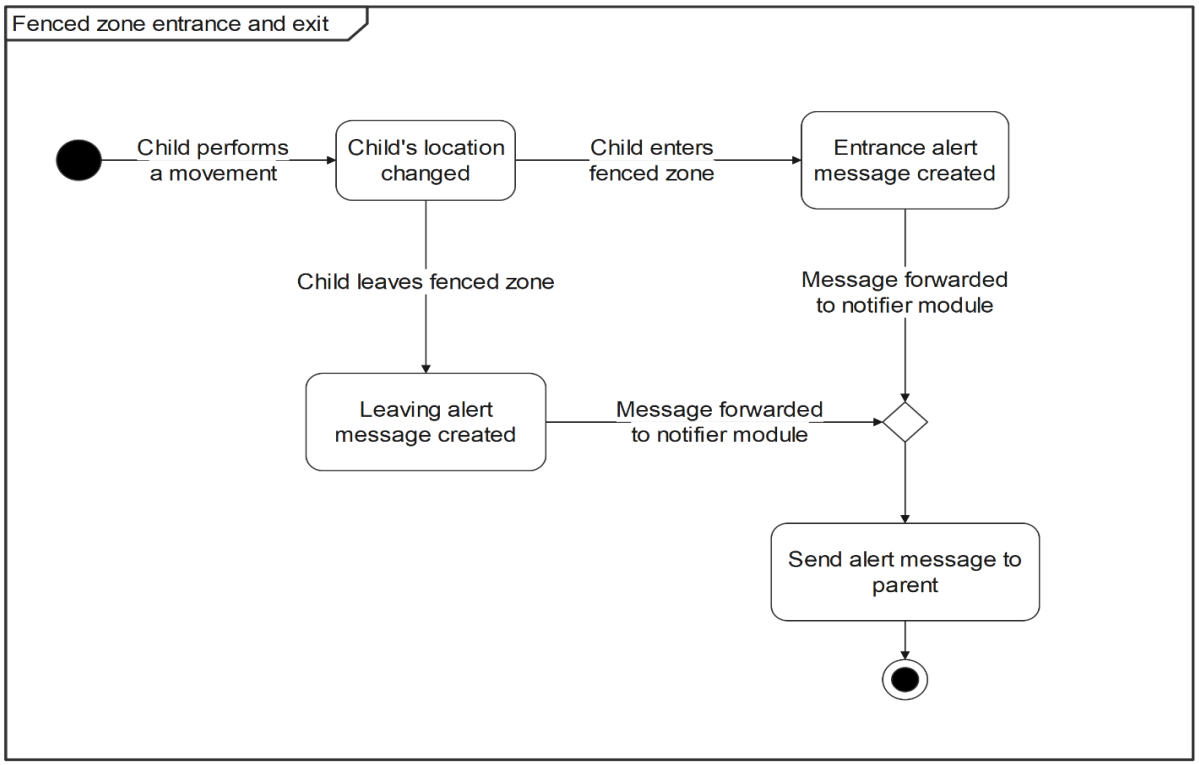
4.4.2 The child says “HELP”



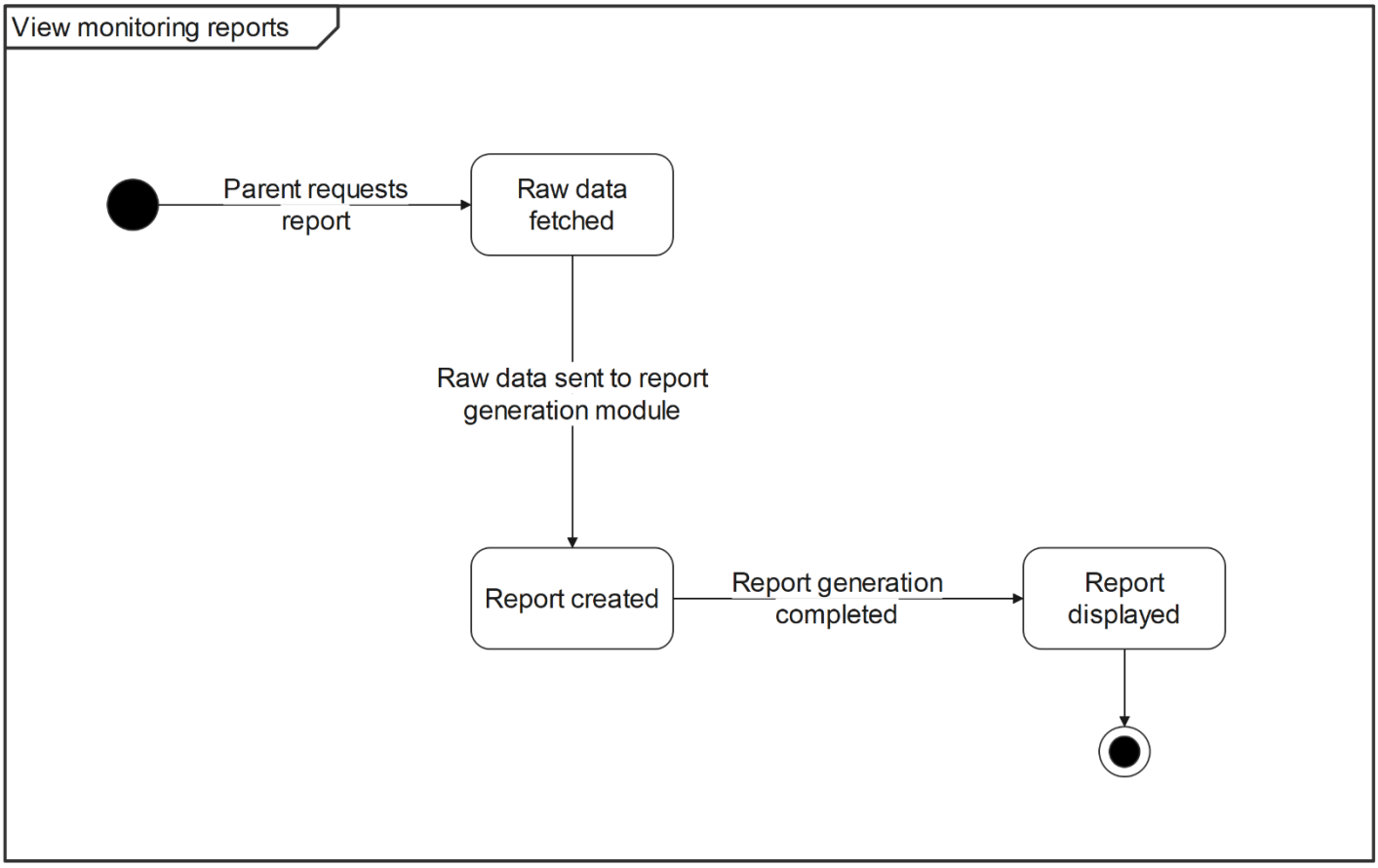
4.4.3 Low battery alert



4.4.4 Fenced zone entrance and exit



4.4.5 Monitoring reports requested



5. Data Design

5.1 Data Dictionary

5.1.1 Users

Key	Property name	Data type	Description
PK	user_id	int	To uniquely identify each user in the DB
	Name	varchar	To store the full name of the user
	PhoneNumber	varchar	To store the phone number of the user, used while sending emergency alerts.
	Email	varchar	Used by user for login
	Password	varchar	Used by user for login into the CMS system
	userType	int	To identify whether the user is a child or a parent. 0 represents the child, 1 represents the child

5.1.2 VisitedLocations

Key	Property name	Data type	Description
PK	record_id	int	To uniquely identify each location visit record
FK	user_id	int	To identify to which child this location visit record belongs, or to identify the particular child who visited this location
	visitTime	DateTime	The time at which the child visited this location
	longitude	double	The longitude of the location, for accurately identifying it
	latitude	double	The latitude of the location

5.1.3 Calls

Key	Property name	Data type	Description
PK	record_id	int	To uniquely identify each phone call record
FK	user_id	int	To identify to which child this call record belongs, or to identify the particular child who performed this phone call
	startTime	DateTime	The time at which the phone call started
	duration	float	The duration for which the call lasted
	callerName	varchar	The name of the person on the other side of the call
	callerPhoneNumber	varchar	The phone number of the person on the other side of the call

5.1.4 Messages

Key	Property name	Data type	Description
PK	record_id	int	To uniquely identify each message record
FK	user_id	int	To identify to which child this message record belongs to
	time	DateTime	The time at which the message was received
	body	varchar	The body of the text message
	senderName	varchar	The name of the person who sent the message
	senderPhoneNumber	varchar	The phone number of the person who sent the message

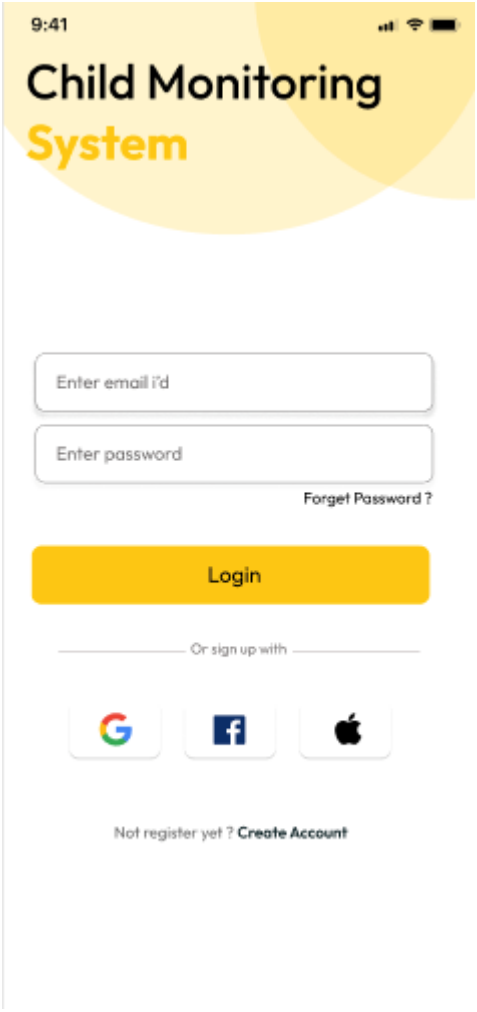
5.1.5 Contacts

Key	Property name	Data type	Description
PK	record_id	int	To uniquely identify each contact record
FK	user_id	int	To identify to which child's phone this contact is stored in
	contactName	varchar	Name of the person whose contact number this is
	phoneNumber	varchar	Phone number of the person whose contact number this is

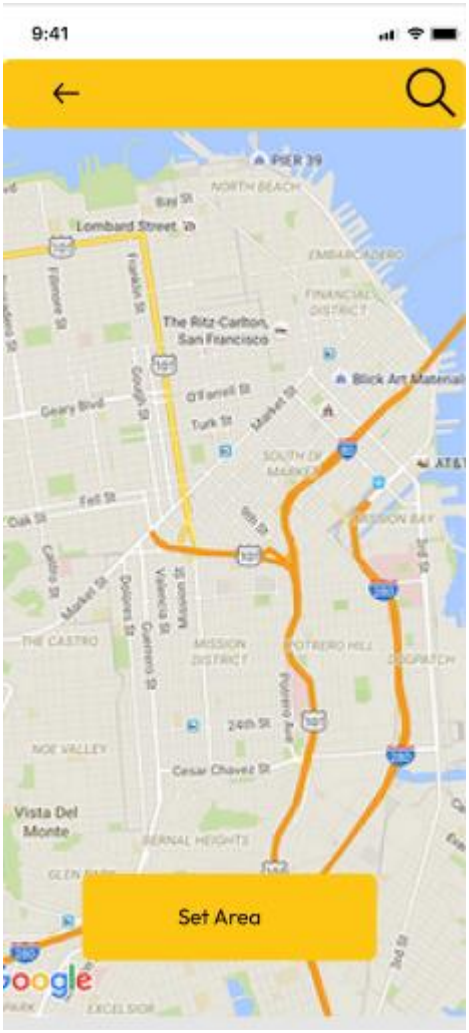
6. Human Interface Design

6.1 Screen Images

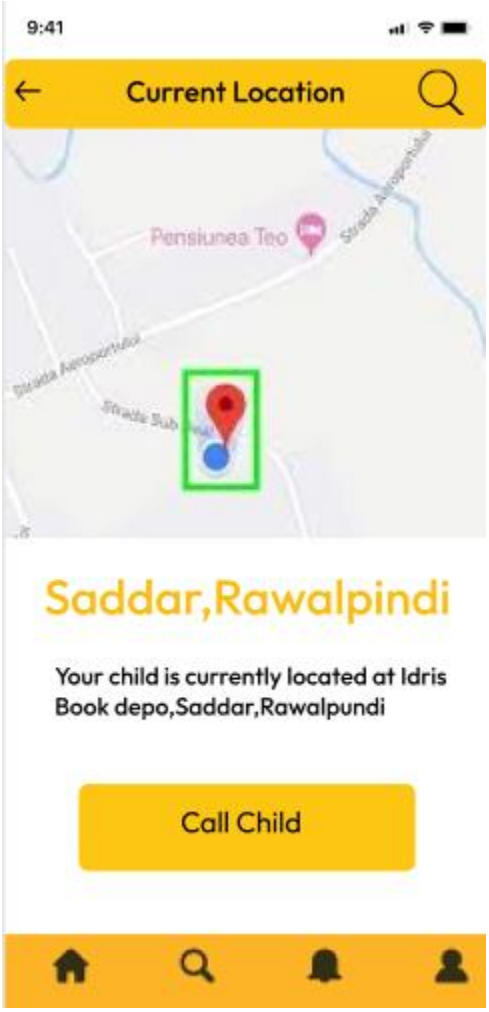
6.1.1 Login



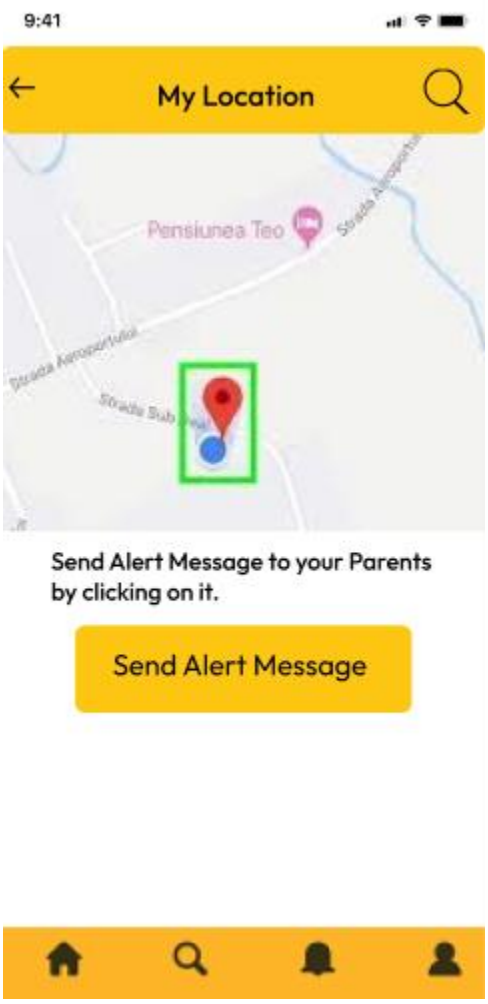
6.1.2 Set fences



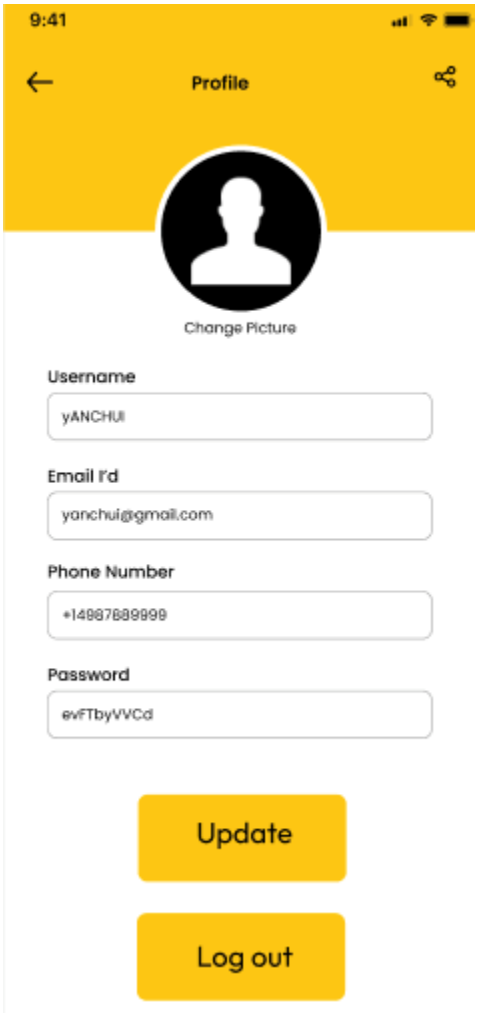
6.1.3 See Current location



6.1.4 Send Alert Message



6.1.5 Set Profile



6.2 Screen Objects and Actions

6.2.1 Map view

This view contains the map that is being displayed through Google Maps API. The child and parent could interact with this view to view locations, get and set fenced zones, etc.

6.2.2 Alert button

This button will trigger the SOS notifier and will send an emergency SOS notification to the parent, along with the current location of the child. The child could also send an SOS notification by speaking the voice command “HELP”.

6.2.3 Report view

This view will be viewable by a parent and contains reports of activities performed by the child, including SMS, calls, contacts, and movement history. The parent could also download these reports.

6.2.4 Set Area button

By specifying the fence zone on the map, the parent could then save the specified zone as a fenced zone and will get alerts whenever the child enters or leaves this zone.

6.2.5 Login button

This button will let the users of the app login into the app, and use its features by entering their login credentials.