COMSATS University, Islamabad Pakistan

Child Monitoring System

By

Afshan Rehman **CIIT/SP18-BSE-006/ISB**

Hadia Khalid **CIIT/SP18-BSE-035/ISB**

Supervisor

Mr. Atique Ahmed Zafar

Bachelor of Science in Software Engineering (2018-2022)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

COMSATS University, Islamabad Pakistan

Child Monitoring System

A project presented to

COMSATS University, Islamabad

In partial fulfillment

of the requirement for the degree of

Bachelors of Science in Software Engineering (2018-2022)

By

Afshan Rehman	**CIIT/SP18-BSE-006/ISB**
Hadia Khalid	**CIIT/SP18-BSE-035/ISB**

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Afshan Rehman

Hadia Khalid

Afshan

Hadia

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (SE) “Child Monitoring System” was developed by **Afshan Rehman (CIIT/SP18-BSE-006)** and **Hadia Khalid (CIIT/SP18-BSE-035)** under the supervision of “Mr. Atique Ahmed Zafar” and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

Supervisor

External Examiner

Head of Department

(Department of Computer Science

Executive Summary

In today's busy life, parents are working continuously and can't watch over their children activities all the time. With the increase of technology advancement, most of the children are using mobile phones which may lead them being in vulnerable situations if proper monitoring and guidance is not provided to them. Parents not knowing whether their child are at some times or are in contact with strangers is a serious problem. Excessive use of mobile can be problematic and Life threatening games (like blue whale) can affect mental health and growth of children. Harmful mobile apps and social media installed on children smartphones not only affect the children mental health but is a major cause in spreading child abuse.

To prevent all these problems and facilitate parents, the CMS is developed. It is a monitoring system, which provides solution to prevent many of these problems. It is an android based smartphone application which is developed keeping the parent's worries in mind. It ensures the safety of children and help parents monitor their child activities.

Child Monitoring **S**ystem **(CMS)** is a monitoring android application that consists of two modes; Parent and Child. In Parent mode, the system can track child's real time live location and also maintain location history of each child. It also provides content monitoring of child device which includes monitoring of text messages, call history, contacts, installed apps. It gives a precise installed apps usage stats report to the parent so that they can keep track of what kind of app their child is in frequent use and how much time does he spend on it. It also provides parent an ability to set geo fences for their children. This will help parent to set geo fences around the places which they don't want their child to be at. So, that whenever their child enters/exits such place which is set as a geo fence region parent will receive an alert accordingly. In Child mode, it allows child to send SOS in any panic or emergency situation to their parent. Similarly, low battery alerts will be generated and send to parent as a helpful reminder.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor "Mr. Atique Ahmed Zafar". Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to him for his encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Afshan Rehman Hadia Khalid

Afshan Hadia

Abbreviations

SRS	Software Requirement Specification
SDD	Software Design Document
CMS	Child Monitoring System
UC	Use Case
FR	Functional Requirement
NFR	Non Functional Requirement
TC	Test Case
UT	Unit Test

Table of Contents

List of Figures

1 Introduction

The introduction section of our report gives brief overview of our project including the intent and signification of our product. It will also describe the similar existing system weaknesses and our project strengths. It will include tool and technology used in our project, project constraints and its deliverables while also describing our project relevance to different course modules.

1.1 Vision Statement

For parents/guardian who are concerned about their child safety and face trouble to keep an eye on their child mobile activities, **CMS** is an android application which helps to solve and prevent from any unwanted circumstances. The system will consist of two modes; Child (13-18 years old) and Parent. In child mode, the child will able to send SOS notifications and share their current location with the parents in any emergency situation. The child can also send SOS alert using voice command "HELP" whenever the app received this command from the child, the system will automatically share alert message and current location with parent. In parent mode, the system will allow parents to set geo fence zones for their children and will receive alerts whenever the child enter or exits those zones. The application will also allow the parents to monitor the phone activity of their child device (including call history, text messages history, contacts, installed apps, app usage statistics). Parents will also be able to view the current location of their child and receive an alert if the child mobile battery is low. The system will also generate emergency report where the parent will be able to view report of activity log which contains the call list, text message history, last known location and location history of child and will allow them to download this report. Unlike the existing system, our application will provide Geo fencing and will trigger an alert every time device enters or leaves a particular area. As outcome of our project parents find a platform through which they can ensure their child safety while child can also notify their parent in time of need.

1.2 Related System Analysis/Literature Review

As our application follows in the category of a monitoring system, we searched many applications which are related in regard to our project. Some of those products which are similar to our project CMS are Google Family Link which is a free parental control app by Google which covers tracking child’s location and content filtering but with some limitations. It is free of cost but provides limited feature such as device location but shows only most recently logged location on a map, and there’s no geo fencing support, no option to define zones and get alerts when your child arrives or leaves.

Similarly, Life360: Family Locator &GPS Tracker for Safety is a famous family social networking app. It provides location-based service designed primarily to enable friends and family members to share their location with each other. This application is premium based so to access all its features user has to pay for it.

Another application in this regard is B Safe - Never Walk Alone which is a personal safety mobile app. It is a location-based security service that notifies family members and friends in case the user of the app gets in danger. This application is subscription based so accessing all feature require users to pay certain amount. It also doesn’t provide many monitoring features as location history, content monitoring and geo fencing.

Further review of applications of same category also reveals that these apps can perform well on android as compared to iOS. The tendency of these applications is more towards android is due to tight restrictions by Apple Company. Following table shows the related system analysis with target project solution;

Table 1.1: Related System Analysis with Targeted Project Solution

Application Name	Weakness	Proposed Project Solution
• Google Family Link	• No geo fencing • No content monitoring • No live location tracking • No location history • No report creation	Our system will provide live location tracking, location history and content monitoring (which includes monitoring of call logs, text message history, contacts, installed apps, usage stats). It will also provide report creation of child activity log and alert generation whenever child enters or exits a geo fence region.

• Life360: Family Locator &GPS Tracker for Safety	• No content monitoring • No report creation • No location history • Expensive • Needs subscription to access all features.	Our System will provide content monitoring (which includes monitoring of call logs, text messages, contacts, installed apps). It will also update real time location accurately and will be able to create activity log report. It will be free of any cost.
• bSafe-Never Walk Alone	• No geo fencing • No content monitoring • No location history • No report creation • Needs subscription to access all features.	Our system will provide geo fencing and content monitoring (which includes monitoring of call logs, contacts, installed apps, usage stats). It will be free of any cost and will provide report creation of child activity log, live location tracking and location history.

1.3 Project Deliverables

Following are the required deliverables of the project;

- **Project Scope Document**

  In the start of the project we had to deliver the Scope document of our project with the problem statement and its proposed solution. The advantages the proposed solution would bring and a related system analysis of an existing system.

- **Software Requirement Specification Document (SRS)**

  As we moved forward with our project, we had to deliver the SRS Document which would include all the FRs and NFRs of the proposed solution.

- **SDS with 40% implementation**

  After delivering the SRS next we had to deliver the SDD which would include our system's design methodology and software process model. It would also include system overview, design models, data design and algorithm. We also had to deliver 40% implementation of the proposed system solution along with this document.

- **60% implementation with Half Final Project Report**

After delivering the SDD of the system we had to show substantial work done on the proposed solution. The implementation of the modules we showed will be 60%. Along this we also had to deliver the half final project report

- **100% implementation with Final Project Report**

  At this stage, we had to show the 100% implementation of our project along with full final project report.

1.4 System Limitations/Constraints

Following are constraints and limitation of our project;

LI-1: If the child mobile is discharged, system will not be able to give accurate responses but before discharging a low battery alert will be generated to the parent to know about battery percentage of their child.

LI-2: If mobile is lost or powered off, the system will be not give accurate location tracking but it will maintain location log from where the parent can view the last known location.

LI-3: The age group of child will be 13 to 18 years old.

LI-4: For accurate monitoring of child device all required permissions must be granted.

1.5 Tools and Technologies

Our application will be developed using Flutter platform. MS Teams will be used for conducting online meetings with supervisor and group members. MS Word will be used for documentation, MS Power Point to make presentations and MS Visio for Diagram creations. Some other tools and technology which will be used are Firebase for database management, Visual Studio code as IDE and Proto.io for mockups creation. Following table shows the tools and technologies for the targeted project;

Table 1.3: Tools and Technologies for the Targeted Project.

	Tools	Version	Rationale
	MS Teams	Office 365	Online Meetings

Tools And Technologies	Visual Studio Code	4.17.1	IDE
	MS Word	Office 365	Documentation
	MS Power Point	Office 365	Presentation
	Proto.io	6	Mockups Creation
	MS Visio	2016	Diagrams Creation
	Technology	**Version**	**Rationale**
	DART	2.10.0	Programming Language
	Firebase	4.4.0	DBMS
	Flutter	1.20.2	Framework

1.6 Relevance to Course Modules

Child Monitoring System is a related to several courses we have studied during our degree. Different courses and the explanation how they are related to our project is given below:

Table 1.4: Project Relevance to Courses

Course	Relevance to CMS
Introduction to Software Engineering	Helped in designing UML design models
Database System	Helped in storing, handling and querying data
Human-Computer Interaction	Helped in designing interfaces
Software Project Management	Helped in meeting deadlines and making Gant chart
Software Requirement Engineering	Helped in gathering requirement and writing functional / non- functional requirements
Mobile Application Development	Helped in android application development
Data Structure and Algorithms	Helped in creating complex logics

Software Testing	Helped in testing the system

2 Problem Definition

The problem definition section of our report describes the objectives of the project by going through the problem statement and solution in a systemic way. It will describe the main functionality and scope of our project.

2.1 Problem Statement

In today's busy life, parents are working continuously and can't watch over their children activities all the time. With the increase of technology advancement, most of the children are using mobile phones which may lead to them being in vulnerable situations if proper monitoring and guidance is not provided to them. Parents not knowing whether their child are at some times or are in contact with strangers is a serious problem. Excessive use of mobile can be problematic and Life threatening games (like blue whale) can affect mental health and growth of children. Harmful mobile apps and social media installed on children smartphones not only affect the children mental health but is a major cause in spreading child abuse and in such cases, the problem of failing to contact or inform your family in time can lead to serious consequences.

2.2 Problem Solution

Keeping in view the above stated problems about the worries of parent and the cons of technology advancement for child, we are developing CMS an android application for monitoring. The proposed application allows the children to share their location with parent through SOS caution in time of need. Parents will mainly use this system to track their children location, monitor their phone content including their call history, text message history, contacts, installed apps and receives alerts if their children have entered or exit a specific area. They will also be able to receive low battery alert of their children smartphone which will be a helpful reminder. Another aspect of the proposed solution that will help in easing parent worries is the ability of the system to provide location history log details of the children as well as activity log report that can be downloaded by parent. It will also provide parent a detailed install apps usage stats based on daily, weekly, monthly and yearly basis which help them to keep track of their child mobile application usage.

2.3 Objectives of the Proposed System

Following are the objectives of the CMS;

BO-1: To ensure parent can keep track of child live location at any time.

BO-2: To reduce rate of installing inappropriate mobile apps on child device by 90%.

BO-3: To ensure parent can keep an eye on child device content.

BO-4: To ensure parent receive timely alerts about their child to prevent emergency situations by 80%.

2.4 Scope

Our system is an android based application designed with child monitoring and child safety in mind where parent can keep track of their child location in real time and monitor their mobile phone activities. The platform will allow the child to send SOS Caution using voice command and share their current location with the parents in any emergency situation Proper account creation and management will be provided to parent and will allow the parents to register as many child as they want. The prominent feature of our application are content monitoring which is providing a way for parent to monitor their child text message history, call history, contact list, install apps. Other features include monitoring child live location in real time, location history, application usage statistics, geo fence region entry/exit alert (check in /check out), activity log report generation. Parents will have a strong check and balance on their child so that they can keep their child safe all the time.

2.5 Modules

2.5.1 Module 1: User Management

In order to access and use Child Monitoring System, the user must create an account and register themselves in our system. If the account already exists, then he/she just has to login. Our system will have two modes i.e. Parent and Child. All personal information will be stored in the database for later use. Following are the users for our system;

2.5.1.1 Parent

For Parent’s registration, the user need to select role as parent and enter email address, name, password, phone number. For Login, parents must enter the verified email address and password. It will also allow them set new password in case of forgotten password. After login, parent dashboard will be displayed. Parent can also update his profile details.

2.5.1.2 Child

For registration of child, the parent must add child by entering child email address, password, name and age of their child. For Login, the child will use the email address and password set by his/her parent. It will also allow them set new password in case of forgotten password. After login, child dashboard will be displayed.

2.5.2 Module 2: Location Tracking

2.5.2.1 Live Location

Parents will be able to track their child location any time. Real Time Live location will help the parents know about their child location address displayed on map and help them know where are their child at specific time. If something happens to their child, they will be able to quickly reach their current location.

2.5.2.2 Location History

Child location history will be maintained in the system. Parent can view their child location log any time. Through this they will be able to view all the places that their child has visited in the past. In case, if the child mobile location can’t be found the parent will be able to know about their previous visited places.

2.5.3 Module 3: Geo fencing

Parents will be updated on their children's whereabouts by receiving auto entry and exit point alerts. Parents will specify the location and mark the area they want to set as geo fence region. They will enter a specific address, name and set radius. They will receive alert when child enter or exit from the geo fence region. So, parent will be able to know when their child enters or leave those specified places. At any time, parent will also be able to delete the geo fence region.

2.5.4 Module 4: Content Monitoring

2.5.4.1 Monitor SMS History

Parents check on SMS communication including sent and received message of their child. Parent can check related details like name and number of sender and date & time stamps.

2.5.4.2 Monitor Contact

Parents can view all the saved contacts in their child’s device and analyze the unknown or harmful person in their contact list.

2.5.4.3 Monitor Call History

Parents can check total call history including contact numbers, call duration, call time and date stamps.

2.5.4.4 Monitor Installed App

Installed apps can be dangerous or harmful if contains malicious or unwanted content. So, Parents will be able to view the list of all the installed apps in child’s device.

2.5.5 Module 5: App Usage Statistics

Parents can view App Usage Statistics of their child’s device most used apps telling the time they spent on that app. They can view daily, weekly, monthly and yearly usage stats. This will help parents to know their child's app usage pattern closely.

2.5.6 Module 6: Emergency Notifications

2.5.6.1 SOS Caution

During any emergency situation, the child will be able to press an SOS button or give voice command “HELP” in the application which will immediately send an alert message with the child current location to the parent. Parent can then immediately contact their child and can look at its current location to help them in a timely manner.

2.5.6.2 Low Battery Alert

When the child battery percentage is low, the parent will receive a notification reminding them about the low battery percentage of their child. So, that they can remind their child to charge the phone.

2.5.7 Module 7: Reporting

Parent can generate pdf child activity log reports containing child location history, call history, text message history, app usage etc. They can download child activity log reports for later use.

3 Requirement Analysis

The requirement analysis section of our report will include detailed result of the requirement analysis phase of the development of the project. It will describe the selected requirement identifying techniques for our project. It will also include the functional and non-functional requirement while also describing the external interface requirement of our project.

3.1 User classes and characteristics

Following are the user classes and their pertinent characteristics that will use this product;

Table 3.1 User classes and characteristics of the Project

User Class	Description
Parent	A Parent is the user that wants to monitor his/her child location and device activities. Parent will register many children in the application. Parent will then ensure that child account is set up on child device. After setting up child account parent can then view child live location and related device content monitoring details and receive required alerts / notifications.

Child	A child is the user that is being monitored by their parent. Child can send SOS caution to parent any time in need. Through his account many notification will be sent to parent i.e. low battery, geo fence alert etc.

3.2 Requirement Identifying Technique

Keeping in mind the nature and type of our project the requirement identifying techniques selected are as below;

- Use Case (Use Case Diagram and Detailed Use Case)
- Event Response Table (Event List)

3.2.1 Use Case Diagram

Use case diagram will be split up into two different use case diagrams, for each actor i.e. Parent and Child. The first use case diagram will be for the Parent and second will be for Child User. The UML notation [6] is used for creating UCD.

3.2.1.1 Use Case Diagram for Parent

Parent is the actor in this use case diagram, who can perform use cases mentioned in diagram.

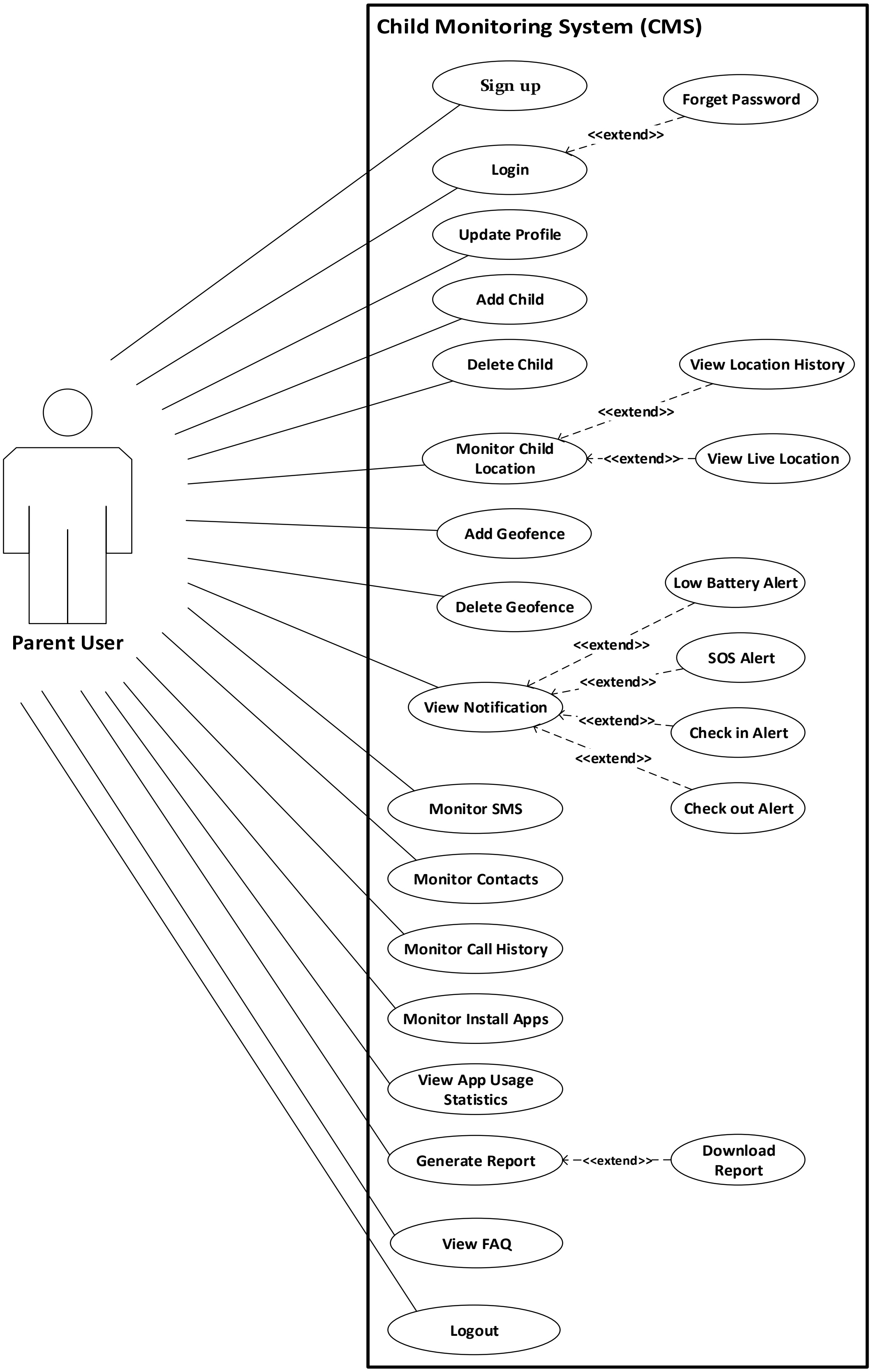


Figure 3.1: Use Case Diagram for Parent

3.2.1.2 Use Case Diagram for Child

Child is the actor in this use case diagram, who can perform use cases mentioned in diagram.

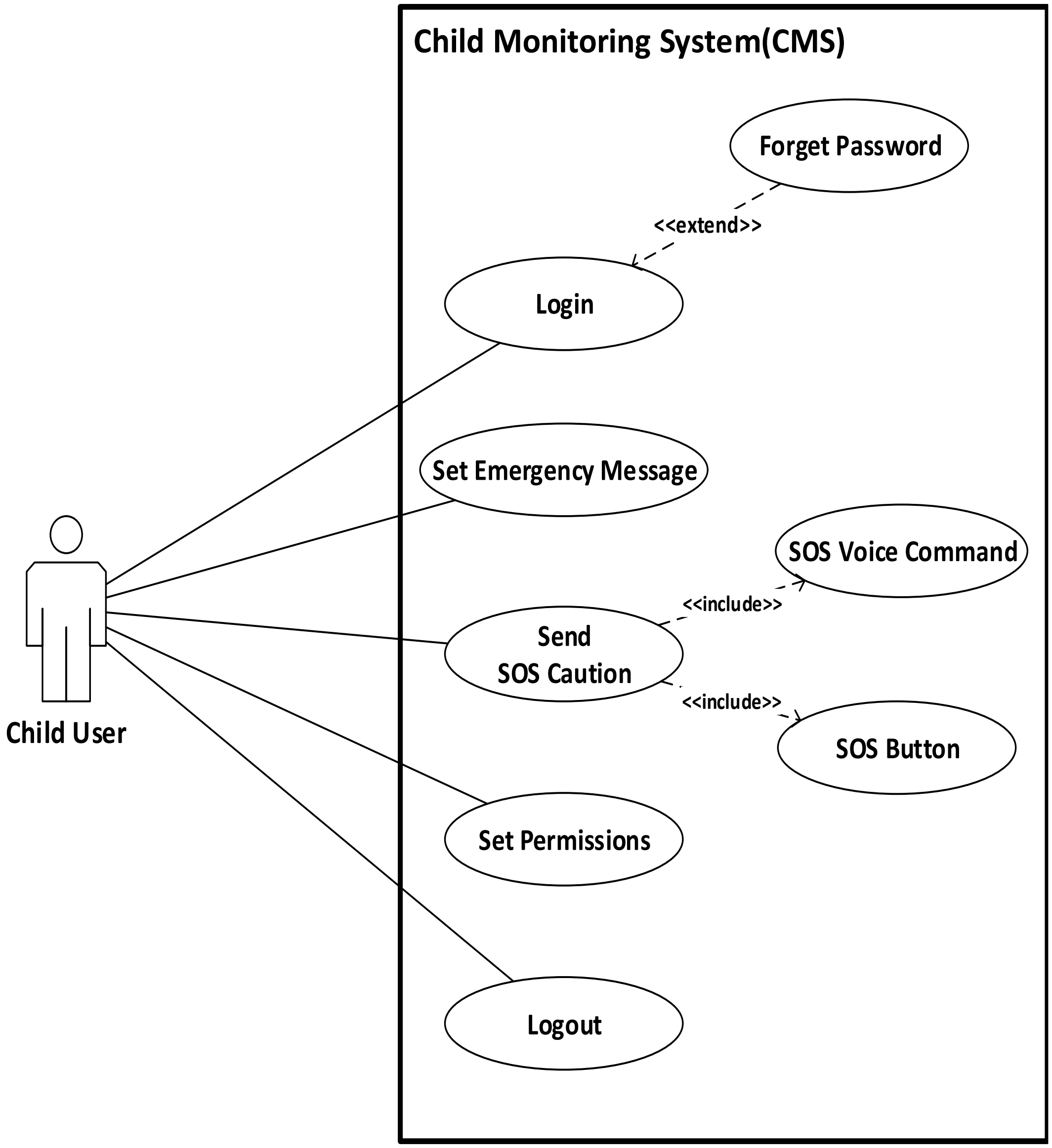


Figure 3.2: Use Case Diagram for Child

3.2.2 Detailed Use Case

3.2.2.1 Sign Up

Table 3.2: Textual Description of Sign Up

Use Case ID:	CMS-UC-01
Use Case Name:	Sign Up
Actors:	Primary Actor: Parent
Description:	This use case describes how users(Parent) gain access to the system through the sign up/registration (account creation) process.
Trigger:	Parents click signup button in order to create a unique account for themselves so that they can be identified by it
Preconditions:	PRE-1 User must be on the sign up screen PRE-2. User must have an email account through which he can sign up PRE-3. The email must not be registered on an another account
Postconditions:	POST-1. A user (parent) account is created successfully POST-2. The user(parent) can login to his/her account
Normal Flow:	1. The user (parent) opens the application and select its role as "Parent". 2. The user (parent) clicks on sign up button. 3. System displays the registration/sign up form to the user. 4. The user (parent) enters credentials such as user name, email, password, confirm password and phone number. 5. The system validates the user information. 6. The system generates and send a verification email link to user (parent). 7. The system displays Verify Email message to user (parent). 8. The user (parent) verify its email and press Done button. 9. The system shows successful account creation message to the user and redirects user to login screen.
Alternative Flows:	N/A

Exceptions:	1. If a user tries to submit an empty form or one of the required fields is missing the system gives an error message 'One or more required fields left empty' 2. If the email format is incorrect an 'Invalid Email' message will be shown 3. If the password and confirm passwords do not match a message will appear saying 'Passwords do not match' 4. If the Phone number format is not correct a message will appear saying "Phone number format is incorrect". 5. If an account already registered with same email address then a message will appear "An account already exists with the same email. Try entering another email".
Business Rules	BR-1: The system does not allow multiple accounts to be created on one single email address. The user can only create one account on one email address. If you try to create a new account, it will not get through and new account creation will be restricted. BR-2: Only registered user is allowed to access the system functionality.
Assumptions:	1. The user knows in general how to fill the form fields 2. Internet connection must be available

3.2.2.2 Login

Table 3.3: Textual Description of Login

Use Case ID:	CMS-UC-02
Use Case Name:	Login
Actors:	Primary Actor: Parent, Child
Description:	This use case describes how users logs into his/her account in order to access the functionality of the application.
Trigger:	The users clicks on Login/Sign in button
Preconditions:	PRE-1. Application must be running on mobile PRE-2. User must be on the login screen PRE-3. User must have created an account and done registration process.
Postconditions:	POST-1. A user is logged in to his/her account POST-2. The user has access to all functionalities provided by the app

Normal Flow:	1. The user clicks on the Login screen. 2. The user enters his/her required credentials i.e. email and password in order to log into the system. 3. The system validates the user information. 4. The system shows successful logged in message and redirects user to dashboard.
Alternative Flows:	In step 2 of the normal flow, if the user forgets his/her password then System shifts to Use Case CMS-UC-03.
Exceptions:	1. If a user tries to submit an empty form or one of the required fields is missing the system gives an error message 'One or more required fields left empty' 2. If a user tries to login to an unregistered account an error message appears saying 'Email not registered, please sign up' 3. If a user tries to login with an error unverified email account an error message appears saying 'account not verified'
Business Rules	BR-1: Only registered user is allowed to access the system functionality.
Assumptions:	Internet connection must be available The user is already registered to an account

3.2.2.3 Forget Password

Table 3.4: Textual Description of Forget Password

Use Case ID:	CMS-UC-03
Use Case Name:	Forget Password
Actors:	Primary Actor: Parent, Child
Description:	This use case describes how the users can reset password at any time with the help of recovery email sent to their email written in their profile.
Trigger:	The users clicks on forgot password button
Preconditions:	PRE-1. User must be on the login screen PRE-2. User must have created an account and done registration process
Postconditions:	POST-1. The user has successfully recovered/reset his/her password.

Normal Flow:	1. The user clicks on “Forgot Password”. 2. The system prompts the user to enter his/her email for reset password. 3. The user enters his email address. 4. The system sends a password recovery link to the user’s email. 5. The user clicks on the link and set a new password. 6. The system displays successful password reset message.
Alternative Flows:	N/A
Exceptions:	If a user tries to submit an empty email fields, the system gives an error message ‘please enter valid email address’.
Business Rules	BR-1: Only registered user is allowed to access the system functionality.
Assumptions:	Internet connection must be available The user is already registered to an account

3.2.2.4 Register/Add Child

Table 3.5: Textual Description of Add Child

Use Case ID:	CMS-UC-04
Use Case Name:	Add Child
Actors:	Primary Actor: Parent
Description:	This use case describes how users(Parent) register/add child in our system. Parent sets child profile and its required information to do registration of child. After that, the system stores all the information related to child in the database.
Trigger:	Parents clicks on add/register child button.
Preconditions:	PRE-1: User(Parent) must be logged in.
Postconditions:	POST-1: Child account is created successfully and he/she can log into the system. POST-2: The system stores the child information in the database successfully.

Normal Flow:	1. The user(parent) navigates to “My Child Tab” and clicks on add/register child button. 2. System displays the add child form to the parent to register their child. 3. The user(parent) enters credentials such as nickname, age, email, password. 4. The system validates the user information. 5. The system shows successful added child message to the user(parent)
Alternate Flows:	N/A
Exceptions:	1. If a user tries to submit an empty form or one of the required fields is missing the system gives an error message ‘One or more required fields left empty’ 2. If the email is invalid an ‘Invalid Email’ message will be shown 3. If a child account already registered with same email address then a message will appear “An account already exists with the same email. Try entering another email”.
Business Rules	BR-1: Only registered user is allowed to access the system functionality. BR-2: Before installing the application on child device, the parents and guardians must get consent of their child.
Assumptions:	Internet connection must be available The user knows in general how to fill the form fields

3.2.2.5 Delete Child

Table 3.6: Textual Description of Delete Child

Use Case ID:	CMS-UC-05
Use Case Name:	Delete Child
Actors:	Primary Actor: Parent
Description:	This use case describes how users(Parent) can delete their child account profile from the system.
Trigger:	Parents clicks on Delete Child button
Preconditions:	PRE-1: Parent must be logged in. PRE-2: Parent must have added a child.
Postconditions:	POST-1. A user (Parent) delete child account profile from the system successfully.

Normal Flow:	1. The user(parent) open My Child tab. 2. The system displays child list to user (parent). 3. The user (parent) presses a child from the list. 4. The system prompts the user a confirmation message asking “Are you sure you want to delete this child account”. 5. The user (parent) does confirmation for deletion. 6. The system deletes all the child information from the database 7. The system shows successful account deletion message to the user.
Alternative Flows:	N/A
Exceptions:	If the system found no child in the database then system will display No Child added yet message to the user(parent).
Business Rules	N/A
Assumptions:	Internet connection must be available.

3.2.2.6 Update Profile

Table 3.7: Textual Description of Update Profile

Use Case ID:	CMS-UC-06
Use Case Name:	Update Profile
Actors:	Primary Actor: Parent
Description:	This use case describes how users (parent) of our system can update and modify their profile details.
Trigger:	Parents clicks on Update Profile Button.
Preconditions:	PRE-1: User (parent) must be logged into the system
Postconditions:	POST-1: The user (parent) profile is successfully updated.
Normal Flow:	1. The user (parent) clicks on update profile button. 2. System displays account information and prompts the user (parent) to update their account details. 3. The user (parent) enter new account information into the system 4. The user (parent) presses update button.

	5. The system shows successful updated message to the user.
Alternative Flows:	N/A
Exceptions:	1. If a user tries to enter invalid new information, the system displays an appropriate error message.
Business Rules	N/A
Assumptions:	Internet connection must be available.

3.2.2.7 Add Geo fence

Table 3.8: Textual Description of Add Geo fence

Use Case ID:	CMS-UC-07
Use Case Name:	Add Geo Fence
Actors:	Primary Actor: Parent
Description:	This use case describes how users(Parent) can set/add geofence to get check in/ check out alert whenever the child enters and leave those specific geo fence areas.
Trigger:	Parents click Add Geo fence button
Preconditions:	PRE-1: The user(parent) must have added a child. PRE-2: Child must have install the application and logged into his account.
Postconditions:	POST-1. A user (Parent) has successfully created a geo fence area for his child.

Normal Flow:	1. The user (parent) open my child tab. 2. The system displays the child list to user (parent) 3. The user (parent) selects a child. 4. The system displays monitoring features to user (parent). 5. The user selects Geo fence tab and clicks on add geo fence button. 6. System displays geo fence form to user (parent) 7. The user set location address, set radius for the geo fence zone and set geo fence region name. 8. The user clicks Done button 9. System displays successfully created geo fence zone message and stores all the information to the database.
Alternative Flows:	N/A
Exceptions:	In step 2 of the normal flow, if the system found no child in the database Then, 1.System will display No Child added yet message to the user(parent). 2.Usecase terminates
Business Rules	BR-1: By using this system, the user consent to the collection and use of GPS location.
Assumptions:	Internet connection must be available The child account must be set up on child device.

3.2.2.8 Delete Geo fence

Table 3.9: Textual Description of Delete Geo fence

Use Case ID:	CMS-UC-08
Use Case Name:	Delete Geo fence
Actors:	Primary Actor: Parent
Description:	This use case describes how users(Parent) can delete a geo fence area they have set to receive check in and check out alert from those geofenced region.
Trigger:	Parents taps Delete Geo fence button.

Preconditions:	PRE-1 User(parent) must have added child account into the system PRE-2. User(parent) must have added some geo fence zones into the system
Postconditions:	A user (Parent) successfully deletes a geo fence area from the system.
Normal Flow:	1. The user (parent) open my child tab. 2. The system displays the child list to user (parent) 3. The user (parent) selects a child. 4. The system displays monitoring features to user (parent). 5. The user (parent) selects Geo fence tab. 6. System retrieves geo fence data from the database and displays the geo fence zones list to the user (parent). 7. The user (parent) select particular geo fence region and press Delete. 8. The system asks for user(parent) confirmation. 9. The user(parent) selects confirm button. 10. The system shows successful deletion message to user (parent).
Alternative Flows:	N/A
Exceptions:	If the system found no child in the database then system will display No Child added yet message to the user(parent) If the system found no geo fence data in the database then the system will display No geo fence added yet message to the user (parent).
Business Rules	N/A
Assumptions:	Internet connection must be available

3.2.2.9 View Location History

Table 3.10: Textual Description of Location History

Use Case ID:	CMS-UC-09
Use Case Name:	View Location History
Actors:	Primary Actor: Parent

Description:	This use case describes how users(Parent) can view their child location log any time. In case, if the child mobile location can’t be found the parent will be able to know about their previous visited places and last known location
Trigger:	The User(Parent) Select the “Location History” option
Preconditions:	PRE-1. User (Parent) must be logged in PRE-2. The user (parent) must have added a child.
Postconditions:	Parents will get the Location History of their child.
Normal Flow:	1. The user (parent) open my child tab. 2. The system displays the child list to user (parent) 3. The user (parent) selects a child. 4. The system displays monitoring features to user (parent). 5. The user (parent) press Location History tab. 6. System prompts the user (parent) to select date. 7. The (user) parent selects date. 8. The system displays the record observed from the child device location on that date to the user (parent).
Alternative Flows:	N/A
Exceptions:	1.If the system found no child in the database then system will display No Child added yet message to the user(parent). 2.If the user (parent) wants to see the location history of their child and no location history is found then in that case The system will display a message that no child location history is found.
Business Rules	BR-1: Child must grant GPS permission.
Assumptions:	Internet connection must be available Application must be running on both child and parent device

3.2.2.10 View App Usage Statistics

Table 3.11: Textual Description of App Usage Statistics

Use Case ID:	CMS-UC-10
Use Case Name:	View App Usage Statistics

Actors:	Primary Actor: Parent
Description:	This use case describes how users(Parent) will be able to view most used apps of child device daily and weekly.
Trigger:	User (Parent) Select “App usage Statistics” option
Preconditions:	PRE-1. User must be logged into the system PRE-2. User(parent) must have added child account into the system PRE-3. Application must be running on both child and parent device
Postconditions:	POST-1. A user (Parent) will be successfully view the app usage statistic of their child device.
Normal Flow:	1. The user (parent) open my child tab. 2. The system displays the child list to user (parent) 3. The user (parent) selects a child. 4. The system displays monitoring features to user (parent). 5. The user (parent) press App Usage Statistic tab. 6. System prompts displays Daily Weekly, Monthly and Yearly Usage options. 7. The user selects “Daily Usage” option. 8. The system receives the request and displays the record of daily usage stats observed from the child’s phone to the parent
Alternative Flows:	6a. In step 6 of the normal flow, if the user selects Weekly Usage option then, The system receives the request and displays the record of weekly usage stats observed from the child’s phone to the parent. 6b. In step 6 of the normal flow, if the user selects Monthly Usage option then, The system receives the request and displays the record of monthly usage stats observed from the child’s phone to the parent. 6c. In step 6 of the normal flow, if the user selects Yearly Usage option then, The system receives the request and displays the record of yearly usage stats observed from the child’s phone to the parent.
Exceptions:	If the user wants to see the usage stats of their child and no usage stats is found then in that case The system will display a message that no child usage statistic is found.
Business Rules	BR-1: By using this system, the user consent to the collection and use of install apps data.
Assumptions:	Internet connection must be available Connection with the database is established.

3.2.2.11 View Notification

Table 3.12: Textual Description of View Notification

Use Case ID:	CMS-UC-11
Use Case Name:	View Notification
Actors:	Primary Actor: Parent
Description:	This use case describes how the user (parent) will receive and view different notifications related to their child.
Trigger:	Parent clicks on Notices Tab.
Preconditions:	PRE-1. User must be logged into the system PRE-2. Parent must have added child into the system PRE-3. Connection with database is established
Postconditions:	Parent can view all the notification received successfully.
Normal Flow:	1. The user (parent) clicks on Notices Tab 2.The system checks for any notification received from the child device and retrieves all the notification which include child low battery notification, geo fence notification, SOS caution notifications. 3.The system displays all the notifications of child device to parent successfully.
Alternative Flows:	N/A
Exceptions:	If system does not find any notification related to child device then System display no notification received message to user (parent).
Business Rules	N/A
Assumptions:	Internet connection must be available

3.2.2.12 Set Emergency Message

Table 3.13: Textual Description of Set Emergency Message

Use Case ID:	CMS-UC-12
Use Case Name:	Set Emergency Message

Actors:	Primary Actor: Child
Description:	This use case will help the user(Child) to set an alert message which will used whenever they press SOS Caution to inform their parent.
Trigger:	Child taps on Set Alert Message option
Preconditions:	PRE-1. User (child) must be logged into the system PRE-2. Connection with database is established
Postconditions:	Child sets emergency alert message successfully.
Normal Flow:	1. The User (Child) selects set alert message button from the dashboard. 3.The system prompts the user(child) to enter an alert message. 4. The user (Child) enter an appropriate alert message and taps on Set button 5. The system stores the alert message in the database and shows a successfully set alert message to the user(child)
Alternative Flows:	N/A
Exceptions:	If a user (child) try to submit an empty alert message, the system will display an appropriate error message to the user to correctly enter a message.
Business Rules	N/A
Assumptions:	Internet connection must be available

3.2.2.13 Set Permissions

Table 3.14: Textual Description of Set Permissions

Use Case ID:	CMS-UC-13
Use Case Name:	Set Permissions
Actors:	Primary Actor: Child
Description:	This use case describes how users(child) gives consent and permission about accessing device location, contacts, text messages, calls.
Trigger:	User (child) press Grant Permission button
Preconditions:	PRE-1. User (child) must be logged in
Postconditions:	POST-1. All device permissions are successfully granted.
Normal Flow:	1. The user (child) logged into the system. 2. The system checks if all permissions are granted.

	3. The system prompts the user to grant all the device permissions required. 4. The user (child) press Grant option to accept all permissions.
Alternative Flows:	N/A
Exceptions:	If the user (child) doesn’t grant permission, the system will display an error message and request to allow the permission.
Business Rules	N/A
Assumptions:	Internet connection must be available.

3.2.2.14 Generate Report

Table 3.15: Textual Description of Generate Report

Use Case ID:	CMS-UC-14
Use Case Name:	Generate Report
Actors:	Primary Actor: Parent
Description:	This use case describes how users(Parent)can will generate the emergency report of their child phone activity log including (location history, sms history, calls history) and download the report.
Trigger:	User (Parent) Select “Generate Report” option
Preconditions:	PRE-1. User must be logged in PRE-2. The user (parent) has already added and set child device in the system.
Postconditions:	POST-1. A user (Parent) will successfully be able to view and download the child activity log report.
Normal Flow:	1. The user (parent) visits the Feature Tab and clicks on generated pdf report icon. 2. The system downloads the activity log report of child device. 3. The system opens the activity log report. 4. The user (parent) views the generated report successfully.

Alternative Flows:	N/A
Exceptions:	If there are no reports data available in the system. The system displays no report available message to the user.
Business Rules	N/A
Assumptions:	Internet connection must be available

3.2.2.15 Send SOS Caution

Table 3.16: Textual Description of SOS Caution

Use Case ID:	CMS-UC-15
Use Case Name:	Send SOS Caution
Actors:	Primary Actor: Child, Secondary Actor: Parent
Description:	This use case will help the user(Child) to generate SOS Alerts that will be sent to the Parents when user(Child) is in critical condition. So that the parent can help their children.
Trigger:	Child clicks SOS button or give voice command “help”
Preconditions:	PRE-1. User(Child) must be logged in PRE-2. Parent must have added child into the system PRE-3. Connection with database is established
Postconditions:	Child user sends the SOS alert to the parent successfully.
Normal Flow:	1. The user (child) open the dashboard. 2.The user(child) select the emergency SOS button. 3. The system sends an SOS alert to the parent. 4.Parent receives SOS notification of their child
Alternative Flows:	In step 2 of normal flow, if user(Child) will select the voice command SOS then, 1. The child presses the microphone and give voice command “Help” 2. The system validates the recorded voice audio 3. Use case resumes to step 3 of normal flow.
Exceptions:	If a user child gives voice command other than “Help”, a message will be displayed “say help to send sos”.

Business Rules	N/A
Assumptions:	Internet connection must be available The user has already set an emergency message.

3.2.2.16 View FAQ

Table 3.17: Textual Description of View FAQ

Use Case ID:	CMS-UC-16
Use Case Name:	View FAQ
Actors:	Primary Actor: Parent
Description:	This use case describes how the user (parent) will view different frequently asked question about the application if he needed any confusion regarding the application.
Trigger:	Parent clicks on FAQ button in side drawer menu.
Preconditions:	PRE-1. User must be logged into the system
Postconditions:	Parent can view all the frequently asked questions related to our application successfully.
Normal Flow:	1. Parent clicks on side menu icon on parent dashboard screen. 2. The system displays the side menu options. 3. Parent selects the FAQ option. 3.The system displays FAQ screen to parent successfully.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules	N/A
Assumptions:	N/A

3.2.3 Event List

As some of our functionality in the use case are actually business event. Business event are those event in which the use case is initiated by the user and the rest of the process is performed without any involvement of the user. Following are event list for such use case;

- Parent initiates a request for monitoring the child location.
- Parent places a request for child contact list to be shown.
- Parent places a request for child call logs to be shown.
- Parent places a request for child SMS history to be shown.
- Parent places a request for install apps on child device to be shown.
- Parent receives Low battery alert detected on child device.
- Parent receives enter/exit alert from geo fenced region for the child device.

3.2.4 Event Response Table

Following are the event response table for above event list;

3.2.4.1 Monitor Location

Parent initiates a request for monitoring the child location.

Table 3.18: Event Response Table for Monitor Location

ID	**Event**	**Data Element**	**System State**	**Exception Condition**	**System Response**		**System State**
					In case of Yes	In case of No	
1	This event occurs when parent requests the child location to be tracked.	Data elements that are needed to process the event are the child device GPS location (latitude and longitude) and maps data.	The current state of the system is normal and shows no child location.	1.If child turns off or disable permission for his device or system GPS location. 2.loss of connection with the database can result in no child information to be shown.	System display the child location update successfully on the map.	System shows a pop error message telling the parent that location couldn’t be found.	System displays child accurate location in the form of marker/ Pointer on maps.

3.2.4.2 Monitor Contact

Parent places a request for child contact list to be shown.

Table 3.19: Event Response Table for Monitor Contact

ID	**Event**	**Data Element**	**System State**	**Exception Condition**	**System Response**		**System State**
					In case of Yes	In case of No	
2	This event occurs when parent requests to monitor child contact list.	Data elements that are needed to process the event are the child device saved contact phone number, name.	The current state of the system is normal and shows no child contact list.	1. If child disables the contacts permission from his device. 2. loss of connection with the database can result in no child information to be shown.	System display the child device contact list successfully to the parent.	System shows a pop error message telling the parent that contacts couldn't be found.	System displays child mobile's contacts including phone number and name to the parent.

3.2.4.3 Monitor Call History

Parent places a request for child call logs to be shown.

Table 3.20: Event Response Table for Monitor Call History

ID	**Event**	**Data Element**	**System State**	**Exception Condition**	**System Response**		**System State**
					In case of Yes	In case of No	
3	This event occurs when parent requests to monitor the child call logs.	Data elements that are needed to process the event are the child's call date, call duration, caller number, missed, incoming and outgoing calls.	The current state of the system is normal and shows no child call logs updates to parent.	1.If child disables the call logs permission from his device. 2.loss of connection with the database can result in no call information to be shown.	System displays the child call logs successfully to the parent.	System shows a pop error message telling the parent that call logs couldn't be found.	System displays to the parent about all the call log history of their child including incoming, outgoing, missed calls and duration of each call.

3.2.4.4 Monitor SMS History

Parent places a request for child SMS history to be shown.

Table 3.21: Event Response Table for Monitor SMS History

ID	**Event**	**Data Element**	**System State**	**Exception Condition**	**System Response**		**System State**
					In case of Yes	In case of No	
4	This event occurs when parent requests to monitor the child mobile's SMS communication.	Data elements that are needed to process the event are the name and number of sender and date & time stamps.	The current state of the system is normal and shows no child SMS information is shown.	1. If child disables the SMS permission from his device. 2. loss of connection with the database can result in no child information to be shown.	System display the child's text messages (SMS) on the parent screen successfully.	System shows a pop error message telling the parent that the notable to access the child's mobile sms.	System displays child SMS to parent including received, sent SMS's date and time stamps.

3.2.4.5 Monitor Install Apps

Parent places a request for install apps on child device to be shown.

Table 3.22: Event Response Table for Monitor Install Apps

ID	**Event**	**Data Element**	**System State**	**Exception Condition**	**System Response**		**System State**
					In case of Yes	In case of No	
5	This event occurs when parent requests to monitor install apps of the child device.	Data elements that are needed to process the event are the list of install apps on child device.	The current state of the system is normal and shows no list of child installed apps.	1. If child disable the permission to access install apps. 2.loss of connection with the database can result in no child information to be shown.	System display the child's install application list to the parent successfully.	System shows a pop up error message telling the parent that no install apps were found.	System displays list of install apps on the child device to the parent.

3.2.4.6 Low Battery Alert

Parent receives Low battery alert detected on child device.

Table 3.23: Event Response Table for Low Battery Alert

ID	**Event**	**Data Element**	**System State**	**Exception Condition**	**System Response**		**System State**
					In case of Yes	In case of No	
6	This event occurs when the system receives data from the child device about the low battery percentage.	Data elements that are needed to process the event are the child device battery data.	The current state of the system is normal and shows no Low battery alert.	1. loss of connection with the database can result in no child information to be accessed.	Notify the parent about low battery percentage of child device.	System displays no low battery alert notification.	System displays child device low battery percentage notification to the parents.

3.2.4.7 Entry/Exit Alert

Parent receives enter/exit alert from geo fenced region for the child device.

Table 3.24: Event Response Table for Entry/Exit Alert

ID	**Event**	**Data Element**	**System State**	**Exception Condition**	**System Response**		**System State**
					In case of Yes	In case of No	

7	This event occurs when the child device enters and exits from the geo fence zone.	Data elements that are needed to process the event are the geo fence area location latitude and longitude, Radius and child current location data.	The current state of the system is normal and shows no entry/exit alert to parent.	1. If child turn off his device Location. 2. loss of connection with the database can result in no child information to be shown.	System displays entry/exit alert from the geo fence area to the parent successfully.	System shows a pop up error message telling the parent that	System displays entry and exit alert notification to the parent.

3.3 Functional Requirements

3.3.1 Sign up

Table 3.25: Description of FR-01

Identifier	FR-01
Title	Sign-up
Requirement	Parent shall be able to create his/her account by clicking on Sign-up option.
Source	Supervisor
Rationale	To access the system user should be registered.
Restrictions and Risk	User shall install the application.
Dependencies	None
Priority	High

3.3.2 Enter Email Address

Table 3.26: Description of FR-02

Identifier	FR-02

Title	Enter Email Address
Requirement	The user shall be able to enter his/her email address.
Source	Supervisor
Rationale	To get valid email of user and to uniquely identify user
Restrictions and Risk	1. Email is required and cannot be empty. 2. Email address that can only contain lower case alphabets, upper case alphabets, numeric values and only special that can occur are @ .
Dependencies	FR-1
Priority	High

3.3.3 Display message for Invalid Email

Table 3.27: Description of FR-03

Identifier	FR-03
Title	Display message for invalid email
Requirement	The system shall display "Invalid Email" pop up message for incorrect email.
Source	Supervisor
Rationale	To provide feedback to the user that email is not matched with the pattern.
Restrictions and Risk	Email is required and cannot be empty. Users can only enter the specified characters.
Dependencies	FR-1, FR-2
Priority	High

3.3.4 Enter Password

Table 3.28: Description of FR-04

Identifier	FR-04
Title	Enter Password
Requirement	The user shall be able to enter his/her password.

Source	Supervisor
Rationale	Users will be able to provide password, which they will use to login later.
Restrictions and Risk	Password length can only be in the specified range. Password is required and cannot be empty.
Dependencies	FR-1
Priority	High

3.3.5 Display Message for Invalid Password

Table 3.29: Description of FR-05

Identifier	FR-05
Title	Display message for invalid password
Requirement	The system shall display “Invalid Password” pop up message for incorrect password.
Source	Supervisor
Rationale	To provide feedback to the user that password is not matched with the pattern.
Restrictions and Risk	Password must contain a digit. Minimum length of password is 8 characters.
Dependencies	FR-1, FR-4
Priority	High

3.3.6 Select Role

Table 3.30: Description of FR-06

Identifier	FR-06
Title	Select Role
Requirement	User shall be able to select role for device as Parent or Child.
Source	Supervisor
Rationale	1. If user wants to register device as Parent. 2. If user wants to register device as Child.
Restrictions and Risk	User must select one of the Role.
Dependencies	None
Priority	High

3.3.7 Email Unavailability

Table 3.31: Description of FR-07

Identifier	FR-07
Title	Email Unavailability
Requirement	The system shall generate an error message notifying the user that the entered email address already exists in the database.
Source	Supervisor
Rationale	To get valid email of user and to uniquely identify user
Restrictions and Risk	Each user must have unique email
Dependencies	FR-1 , FR-2
Priority	High

3.3.8 Enter Phone Number

Table 3.32: Description of FR-08

Identifier	FR-08
Title	Enter Phone Number
Requirement	The user shall be able to enter his/her phone number.
Source	Supervisor
Rationale	To get valid phone number of the user.
Restrictions and Risk	Phone Number is required and can't be empty. Phone Number length must be 11 digits.
Dependencies	FR-1
Priority	High

3.3.9 Display Message for Invalid Phone Number

Table 3.33: Description of FR-09

Identifier	FR-09
Title	Display message for invalid phone number
Requirement	The system shall display "Invalid Phone Number" pop up message for incorrect phone number.
Source	Supervisor

Rationale	To provide feedback to the user that phone number is not matched with the pattern.
Restrictions and Risk	Maximum length of phone number is 11 digits.
Dependencies	FR-1, FR-8
Priority	High

3.3.10 Send Email Verification Link

Table 3.34: Description of FR-10

Identifier	FR-10
Title	Send Email Verification link
Requirement	The system shall send a verification link to the email that user provided while signing up.
Source	Supervisor
Rationale	To verify the email provided by the user.
Restrictions and Risk	User cannot use all the functionality without verifying their email.
Dependencies	FR-1, FR-2, FR-6, FR-7
Priority	High

3.3.11 Verification Success

Table 3.31: Description of FR-11

Identifier	FR-11
Title	Verification Success
Requirement	The system shall display “Email Verified” message after the link is opened by the user.
Source	Supervisor
Rationale	To provide user with proper feedback.
Restrictions and Risk	N/A
Dependencies	FR-1, FR-2, FR-6, FR-7,FR-10
Priority	High

3.3.12 Login

Table 3.32: Description of FR-12

Identifier	FR-12
Title	Login
Requirement	Registered user shall be able to login by entering his/her correct email and password in the login screen.
Source	Supervisor
Rationale	To access his/her registered account user will sign-in.
Restrictions and Risk	User shall be registered and have valid account.
Dependencies	FR-11
Priority	High

3.3.13 Validate Login Credentials

Table 3.33: Description of FR-13

Identifier	FR-13
Title	Validate Login Credentials
Requirement	The system shall be able to validate user's login credentials by verifying the account information of the user from the database.
Source	Supervisor
Rationale	To verify user login credentials from the database.
Restrictions and Risk	Connection to Database must be Established
Dependencies	FR-12
Priority	High

3.3.14 Forgot Password

Table 3.34: Description of FR-14

Identifier	FR-14
Title	Forgot Password

Requirement	The user shall be able to reset his/her password by clicking on Forgot Password option.
Source	Supervisor
Rationale	To recover and change password for a user.
Restrictions and Risk	N/A
Dependencies	FR-13
Priority	High

3.3.15 Edit Profile

Table 3.35: Description of FR-15

Identifier	FR-15
Title	Edit Profile
Requirement	User shall be able to edit his/her account by changing and saving the required information.
Source	Supervisor
Rationale	To update account information users, require the edit functionality.
Restrictions and Risk	Connection to the database is established. System will overwrite old data when user enters new data in editable fields.
Dependencies	FR-12
Priority	Medium

3.3.16 Add Child

Table 3.36: Description of FR-16

Identifier	FR-16
Title	Add Child
Requirement	Parent shall be able to add his/her child in the system by entering and saving the required information.
Source	Supervisor
Rationale	To register child in the system
Restrictions and Risk	Connection to database must be established
Dependencies	FR-12
Priority	High

3.3.17 Display Child List

Table 3.37: Description of FR-17

Identifier	FR-17
Title	Display Child List
Requirement	Parent shall be able to view all child by clicking on My child tab.
Source	Supervisor
Rationale	To display all child added to the parent.
Restrictions and Risk	Parent must set up child profile. Connection to database must be established
Dependencies	FR-16
Priority	High

3.3.18 Delete Child

Table 3.38: Description of FR-18

Identifier	FR-18
Title	Delete Child
Requirement	Parent shall be able to delete his/her child account by clicking on “Delete Child Account” button.
Source	Supervisor
Rationale	To delete child account information from the database user requires the delete functionality.
Restrictions and Risk	Parent must have added a child in their child list. Database connection must be Established
Dependencies	FR-16,FR-17
Priority	Medium

3.3.19 Logout

Table 3.39: Description of FR-19

Identifier	FR-19

Title	Logout
Requirement	User shall be able to logout from the system by selecting on the “logout” option
Source	Supervisor
Rationale	To enable user to logout from the system
Restrictions and Risk	N/A
Dependencies	FR-12
Priority	High

3.3.20 Display All Child Location

Table 3.40: Description of FR-20

Identifier	FR-20
Title	Display All Child Location
Requirement	Parent shall be able to view all child location by clicking on All Child tab.
Source	Supervisor
Rationale	To have access and monitor current live location of child device.
Restrictions and Risk	Child account must be set by parent and device location permission must be granted.
Dependencies	FR-16,FR-17
Priority	Medium

3.3.21 Display Selected Child Location

Table 3.41: Description of FR-21

Identifier	FR-21
Title	Display Selected Child Location
Requirement	Parent shall be able to select a particular child to view his/her location after the map is loaded.
Source	Supervisor
Rationale	To have monitor current live location of child device.

Restrictions and Risk	Child account must be set by parent and device location permission must be granted.
Dependencies	FR-16,FR-17
Priority	High

3.3.22 Unable to load location

Table 3.42: Description of FR-22

Identifier	FR-22
Title	Unable to load Location
Requirement	The system display “No location found” message to the parent if child location is not found.
Source	Supervisor
Rationale	To give proper feedback to user.
Restrictions and Risk	N/A
Dependencies	FR-16,FR-17
Priority	Low

3.3.23 Display Location History

Table 3.43: Description of FR-23

Identifier	FR-23
Title	Display Location History
Requirement	The Parent shall be able to view the location history of child device by selecting the date on location history screen.
Source	Supervisor
Rationale	To view the location history of child.
Restrictions and Risk	Child account must be set by parent and device location permission must be granted.
Dependencies	FR-21
Priority	High

3.3.24 Unable to load location history

Table 3.44: Description of FR-24

Identifier	FR-24
Title	Unable to load location history
Requirement	The system display “No location history” message to the parent if child location history is not found.
Source	Supervisor
Rationale	To give proper feedback to user.
Restrictions and Risk	N/A
Dependencies	FR-16,FR-17
Priority	Low

3.3.25 Add Geo fence

Table 3.45: Description of FR-25

Identifier	FR-25
Title	Add Geo fence
Requirement	Parent shall be able to add geo fence for their by providing the radius and location address for the geo fence region.
Source	Supervisor
Rationale	To create geo fences for child device.
Restrictions and Risk	Child account must be set by parent and device location permission must be granted.
Dependencies	FR-16,FR-17
Priority	High

3.3.26 Display Geo Fence

Table 3.46: Description of FR-26

Identifier	FR-26
Title	Display Geo fence
Requirement	Parent shall be able to view all Geo fences set for a particular child by clicking on Monitor Geo fence Tab.
Source	Supervisor

Rationale	To view geo fences for child device.
Restrictions and Risk	N/A
Dependencies	FR-25
Priority	High

3.3.27 Generate Geo Fence Alert

Table 3.47: Description of FR-27

Identifier	FR-27
Title	Generate Geo fence Alert
Requirement	The system shall be able to notify parent whenever the child enters/exits a geo fence region.
Source	Supervisor
Rationale	To monitor and notify parent about child device location.
Restrictions and Risk	N/A
Dependencies	FR-25
Priority	High

3.3.28 Delete Geo Fence

Table 3.48: Description of FR-28

Identifier	FR-28
Title	Delete Geo Fence
Requirement	Parent shall be able to delete a geo fence region set for a child by click on delete geo fence option
Source	Supervisor
Rationale	To remove geo fences for child device.
Restrictions and Risk	N/A
Dependencies	FR-25
Priority	Medium

3.3.29 Set Emergency Message

Table 3.49: Description of FR-29

Identifier	FR-29
Title	Set Emergency Message
Requirement	Child shall be able to set emergency message by clicking on Set Alert option.
Source	Supervisor
Rationale	To allow child to create own emergency message.
Restrictions and Risk	N/A
Dependencies	FR-12,FR-16
Priority	Low

3.3.30 Send SOS Caution through SOS button

Table 3.50: Description of FR-30

Identifier	FR-30
Title	Send SOS Caution through SOS Button
Requirement	Child shall be able to send SOS caution to parent by clicking on SOS option.
Source	Supervisor
Rationale	To allow child to notify his/her parent.
Restrictions and Risk	N/A
Dependencies	FR-12,FR-16,FR-29
Priority	High

3.3.31 Send SOS Caution through Voice Command

Table 3.51: Description of FR-31

Identifier	FR-31
Title	Send SOS Caution through Voice Command
Requirement	Child shall be able to send SOS caution to parent by giving Voice Command “Help” to the system.
Source	Supervisor

Rationale	To allow child to notify his/her parent.
Restrictions and Risk	N/A
Dependencies	FR-12,FR-16,FR-29
Priority	High

3.3.32 Low Battery Notification

Table 3.52: Description of FR-32

Identifier	FR-32
Title	Low Battery Notification
Requirement	The system shall be able to notify the parent if child device battery percentage is low.
Source	Supervisor
Rationale	To notify parent about child low device battery.
Restrictions and Risk	N/A
Dependencies	FR-12,FR-16
Priority	High

3.3.33 View Notification

Table 3.53: Description of FR-33

Identifier	FR-33
Title	View Notification
Requirement	Parent shall be able to view all notification by clicking on Notices Tab.
Source	Supervisor
Rationale	To view all notification received about child device.
Restrictions and Risk	N/A
Dependencies	FR-12,FR-16
Priority	High

3.3.34 Display SMS History

Table 3.54: Description of FR-34

Identifier	FR-34
Title	Display SMS History
Requirement	Parent shall be able to view text messages history i.e. sent, received of their child device by clicking on SMS history Tab.
Source	Supervisor
Rationale	To monitor all text messages in child device.
Restrictions and Risk	Child account must be set by parent and SMS permission must be granted.
Dependencies	FR-12,FR-16
Priority	High

3.3.35 Display Call History

Table 3.55: Description of FR-35

Identifier	FR-35
Title	Display Call History
Requirement	Parent shall be able to view call history i.e. incoming, outgoing of their child device by clicking on Call history Tab.
Source	Supervisor
Rationale	To monitor call logs in child device.
Restrictions and Risk	Child account must be set by parent and access calls permission must be granted.
Dependencies	FR-12,FR-16
Priority	High

3.3.36 Display Contact

Table 3.56: Description of FR-36

Identifier	FR-36
Title	Display Contacts

Requirement	Parent shall be able to view all contact list of their child device by clicking on Monitor Contact Tab.
Source	Supervisor
Rationale	To view all contact list in child device.
Restrictions and Risk	Child account must be set by parent and contact permission must be granted.
Dependencies	FR-12,FR-16
Priority	High

3.3.37 Display Install Apps

Table 3.57: Description of FR-37

Identifier	FR-37
Title	Display Install Apps
Requirement	Parent shall be able to view all device apps of their child device by clicking on Monitor Apps Tab.
Source	Supervisor
Rationale	To monitor all the install apps in child device.
Restrictions and Risk	Child account must be set by parent and access apps permission must be granted.
Dependencies	FR-12,FR-16
Priority	High

3.3.38 Unable to load content

Table 3.58: Description of FR-38

Identifier	FR-38
Title	Unable to load content
Requirement	The system shall display “ no content found” message if there is no content of child stored in database.
Source	Supervisor
Rationale	To give proper feedback to user.
Restrictions and Risk	N/A

Dependencies	FR-16
Priority	Low

3.3.39 View App Usage Stats

Table 3.59: Description of FR-39

Identifier	FR-39
Title	View App Usage Stats
Requirement	Parent shall be able to view most used app usage stats i.e. usage time duration of child device by clicking on weekly and daily stats option.
Source	Supervisor
Rationale	To inform parent about the most used apps on child device.
Restrictions and Risk	Child account must be set by parent and access apps usage permission must be granted.
Dependencies	FR-16, FR-37
Priority	High

3.3.40 Unable to load Stats

Table 3.60: Description of FR-40

Identifier	FR-40
Title	Unable to load Stats
Requirement	The system shall display “ no usage stats” message to user if there is no usage duration of apps stored in database.
Source	Supervisor
Rationale	To give proper feedback to user.
Restrictions and Risk	N/A
Dependencies	FR-16
Priority	Low

3.3.41 Generate Report

Table 3.61: Description of FR-41

Identifier	FR-41
Title	Generate Report
Requirement	Parent shall be able generate different pdf report about their child activity log by clicking on PDF option.
Source	Supervisor
Rationale	To provide activity reports to the parent
Restrictions and Risk	N/A
Dependencies	FR-16, FR-30, FR-31, FR-34, FR-35, FR-36
Priority	High

3.3.42 Download Report

Table 3.62: Description of FR-42

Identifier	FR-42
Title	Download Report
Requirement	Parent shall be able download the report of their child by clicking on PDF option.
Source	Supervisor
Rationale	To allow parent to download report for later use.
Restrictions and Risk	N/A
Dependencies	FR-41
Priority	Medium

3.4 Non-Functional Requirements

3.4.1 Usability

USE-1: The user shall be able to use 95% of the features after 10 minutes of using the application for the first time.

USE-2*:* Intended users of CMS are Parents, Guardians and children.

USE-3*:* The system shall be user friendly by having an interface with proper buttons, images, error, and confirmation messages

USE-4: CMS shall have FAQ button on the screen to help user understand the working of the system.

USE-5: CMS shall produce user friendly messages in case of system delay.

3.4.2 Reliability

REL-1: User shall be able to rely on the system for his every request. If the user uses the system for tracking child live location, then system shall be 90% sure that the system will display him about the accurate location results.

REL-2: CMS will have a very low frequency of failure i.e. 5% and the system shall be able to recover its original state and functionality in chance of a failure.

3.4.3 Performance

PER-1: CMS shall respond within 4 seconds for more than 90% of operations.

PER-2: More than 100 concurrent users shall be handled by CMS with no lag in performance

3.4.4 Security

SE-1: Only authenticated user's data must be stored on the databases. Only authorized user will be given access to the application functionality and all its personal information will be kept in a secure manner.

3.5 External Interface Requirements

3.5.1 User Interfaces Requirements

UI-1: The color scheme that will be used is based on pastel colors so that they appear soft to the eyes of the users and does not trigger any photo-sensitive person

UI-2: A menu icon (three stack bars) will be shown at top-left of the header on dashboard to enable users the access to different screens.

UI-3: Bottom and Top slider bar navigation will be used instead of buttons to increase user interface attractiveness.

UI-4: A notification bell-icon will also be displayed where the user will receive all the related notifications

UI-5: Back button will be displayed on every screen the application.

UI-6: Different material icons will be used to make the user interface interactive.

UI-7: FAQ option will be given to brief about overall functionality to the user.

3.5.2 Software interfaces

SI-01*:* Visual Studio Code will be used for the development of the mobile application.

SI-02*:* Firebase Database will be used for saving and maintaining the records of the users.

SI-03: Firebase Cloud Messaging will be used to send all the notifications to the user.

SI-04: CMS shall use Google Maps to display the current child location to parent.

SI-05; CMS shall use geo fence service library (3.3.0) to enable geo fences for child device.

SI-06: CMS shall use speech to text library (5.2.0) to convert child voice command to string.

SI-07: CMS shall calendar strip package to display calendar dates to the parent.

3.5.3 Hardware interfaces

HI-1: Our mobile application will run on android based smartphones having android version 4.7 or higher. We will also utilize mobile GPS sensors, microphone, storage, Wi-Fi for smooth monitoring purpose.

3.5.4 Communications interfaces

CI-1: CMS shall use the communication protocols like HTTP (POST) to send notifications to respective user.

CI-2: CMS shall send verification email link to the parent’s email to verify its email to enable usage of its account.

CI-3: CMS shall send forget password link to the user’s email to reset its password.

4 Design and Architecture

This section of our report will describe the complete design and architecture of our project. It will include different design models, data design and describe the interface of the system from user's perspectives.

4.1 Architectural Design

Mobile applications should be robust and flexible in nature due to its computational ability. For mobile application of our system, we will follow "MVVM Architecture". This architecture will help us reduce code complexity and will help us maintain a clean and reusable structure of our code. MVVM is useful to move business logic from view to ViewModel and Model. ViewModel is the mediator between View and Model which carry all user events and return back the result. MVVM is preferable for our project because it supports an event-driven approach, which is quite suitable with flutter components which are performed based on events.

MVVM has three major parts which are as follows:

- **Model:** The Model in the MVVM design pattern represents the actual data(real-time) which will be used in application development.

- **View Model:** ViewModel is the mediator between View and Model, which accept all the user events and request that to Model for data. Once the Model has data then it returns to ViewModel and then ViewModel notify that data to View. ViewModel can be used by multiple views, which means a single ViewModel can provide data to more than one View.

- **View:** The view is where the user is interacting with Widgets that are shown on the screen. These user events request some actions which navigate to ViewModel, and the rest of ViewModel does the job. Once ViewModel has the required data then it updates View.

Following is the system architecture for CMS:

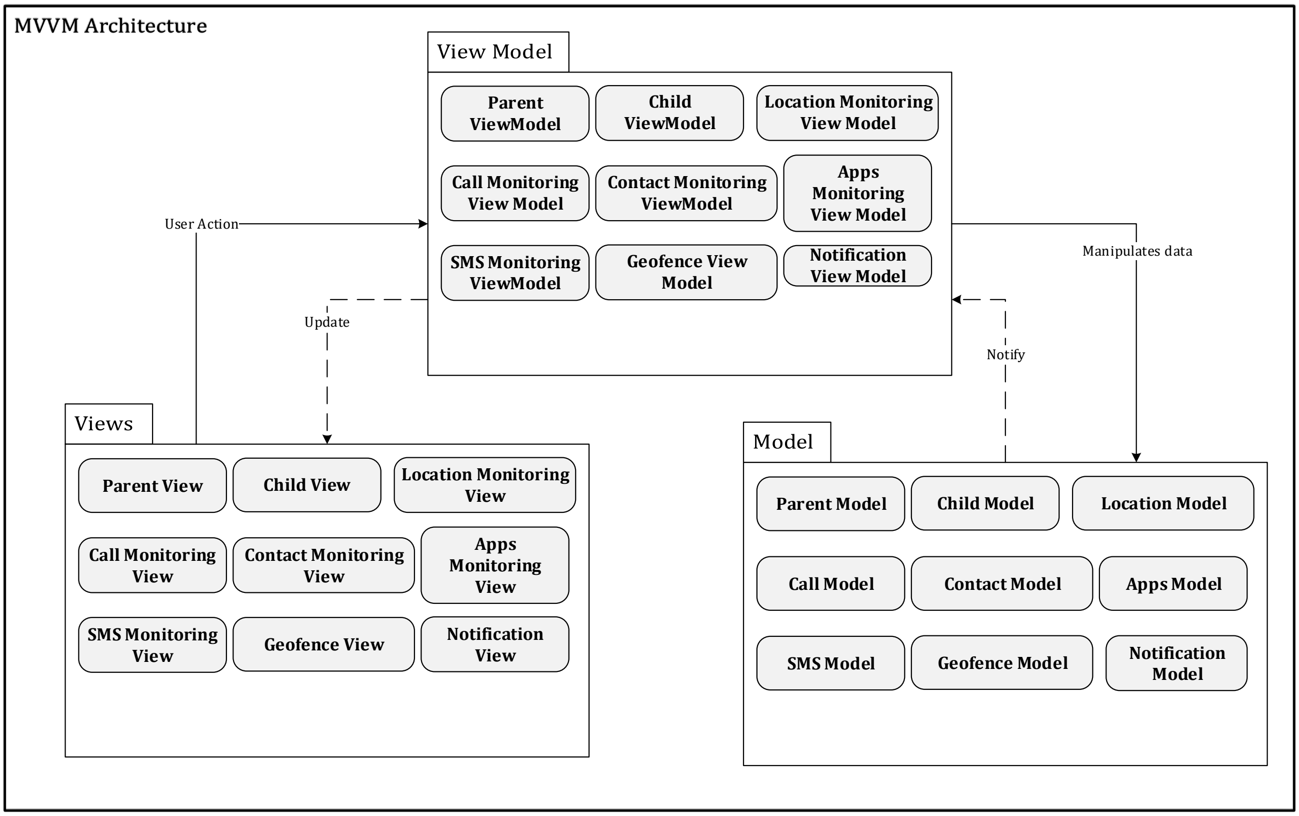


Figure 4.1 Architecture Diagram for CMS

4.2 Design Models

The applicable design models for our project includes:

- Activity Diagram
- Class Diagram
- Sequence Diagram
- State Transition Diagram

4.2.1 Activity Diagram

4.2.1.1 Login

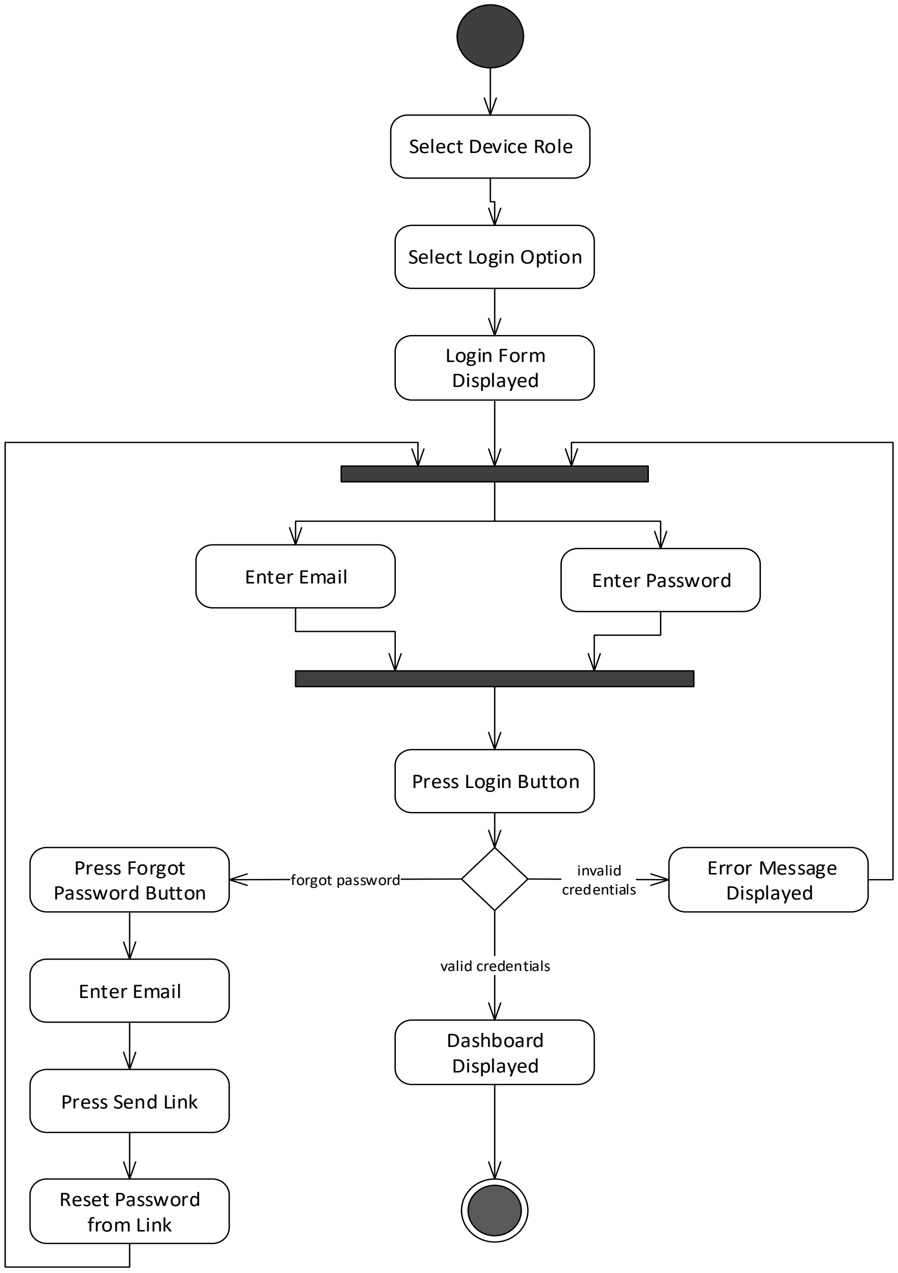


Figure 4.2 Activity Diagram of Login

4.2.1.2 Add Child

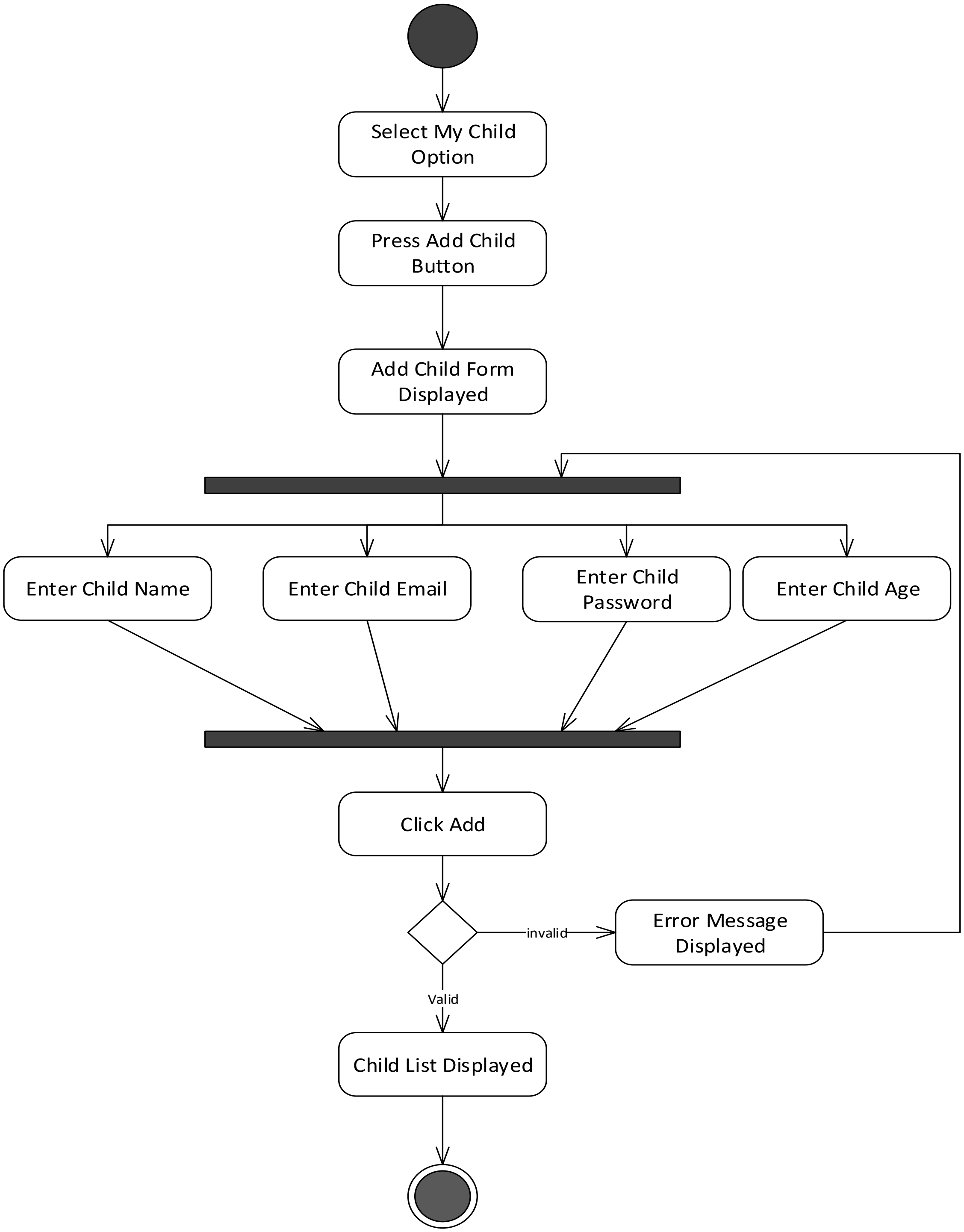


Figure 4.3 Activity Diagram of Add Child

4.2.1.3 Delete Child

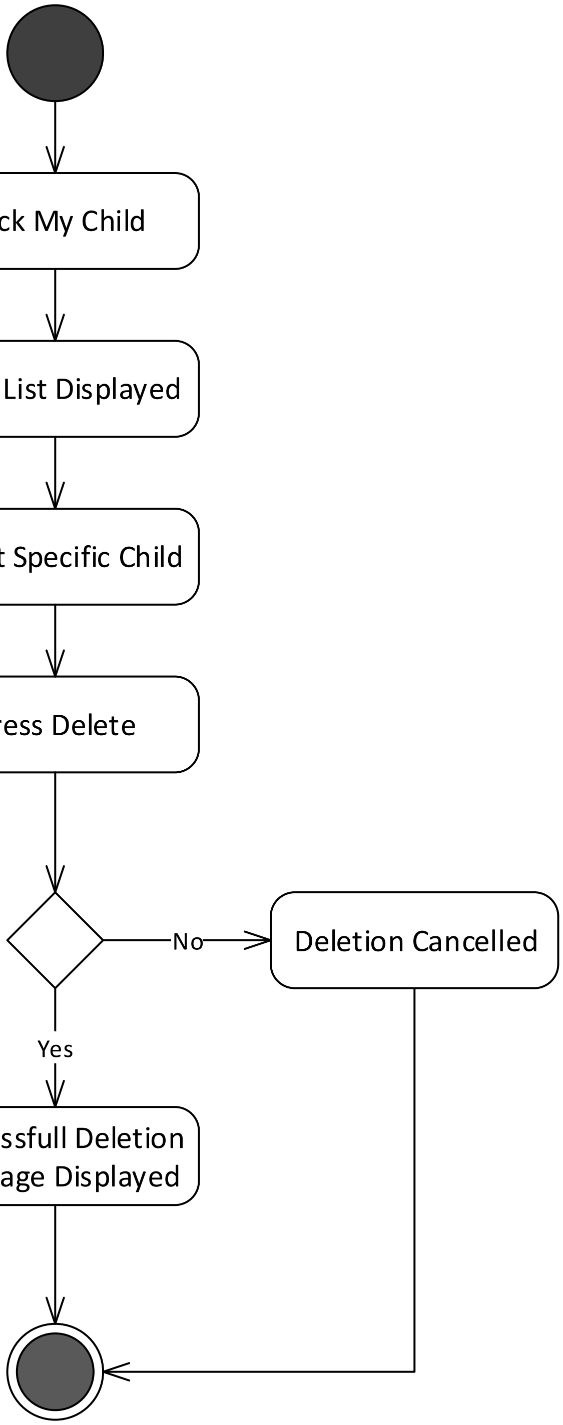


Figure 4.4 Activity Diagram of Delete Child

4.2.1.4 Add Geofence

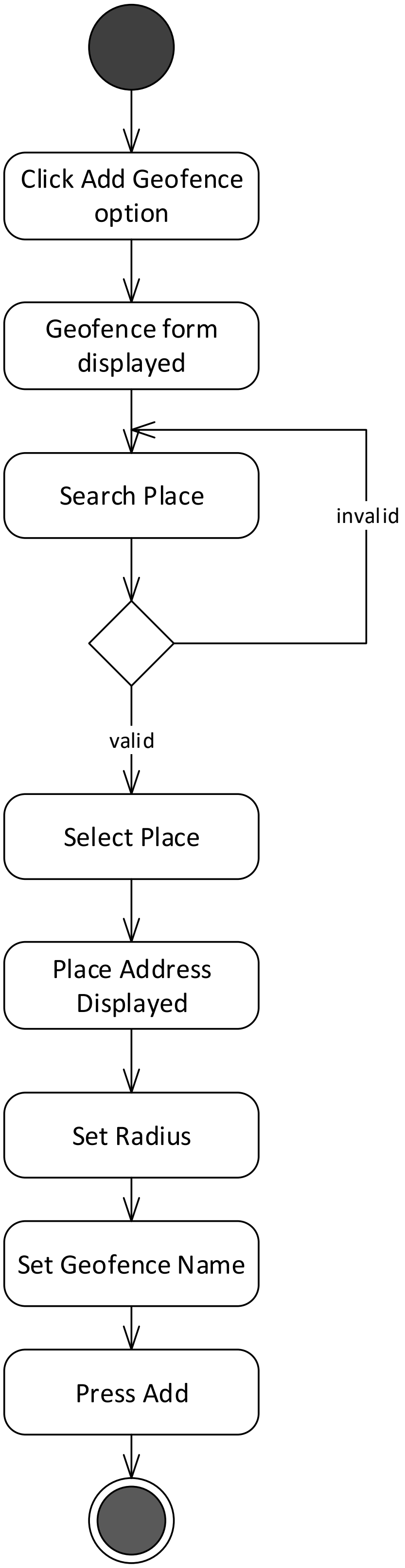


Figure 4.5 Activity Diagram of Add Geofence

4.2.1.5 Update Profile

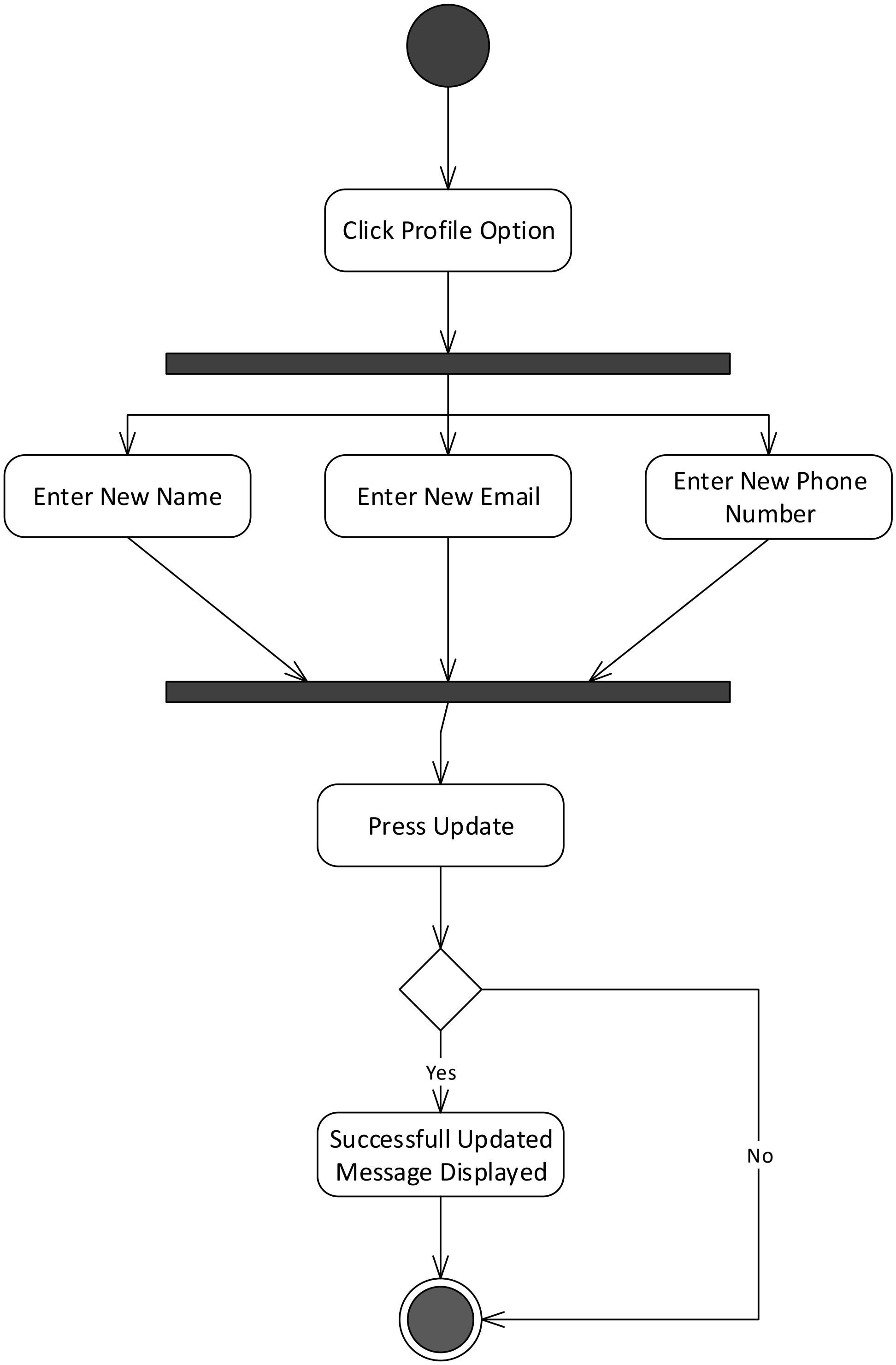


Figure 4.6 Activity Diagram of Update Profile

4.2.1.6 Delete Geo fence

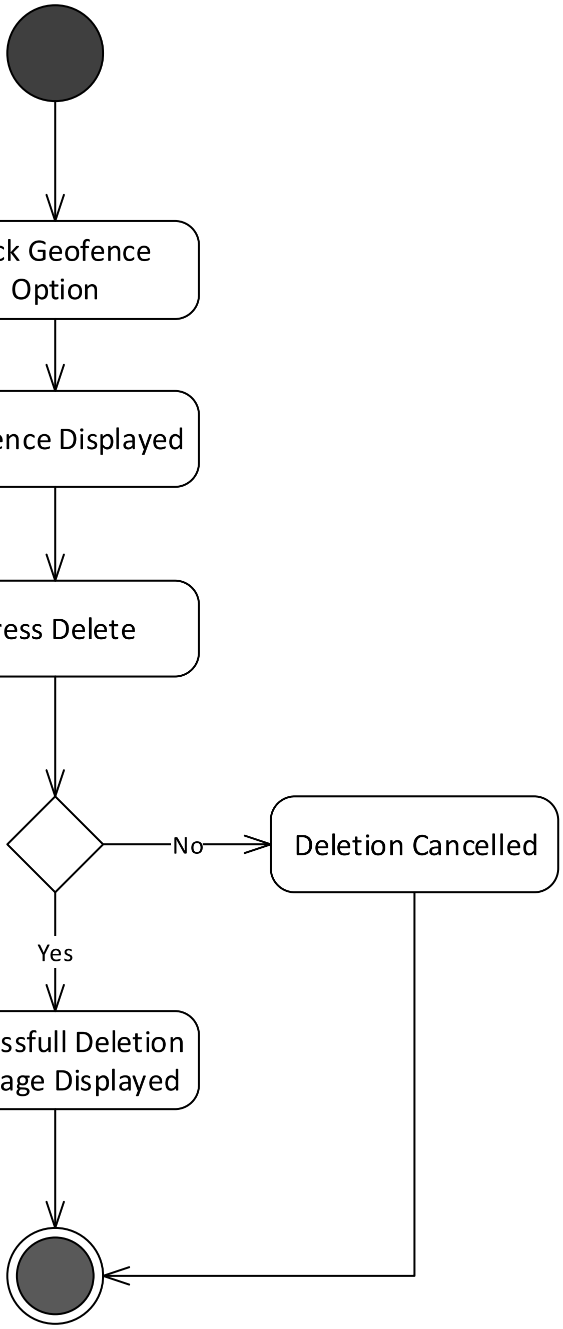


Figure 4.7 Activity Diagram of Delete Geo fence

4.2.1.7 SOS Caution

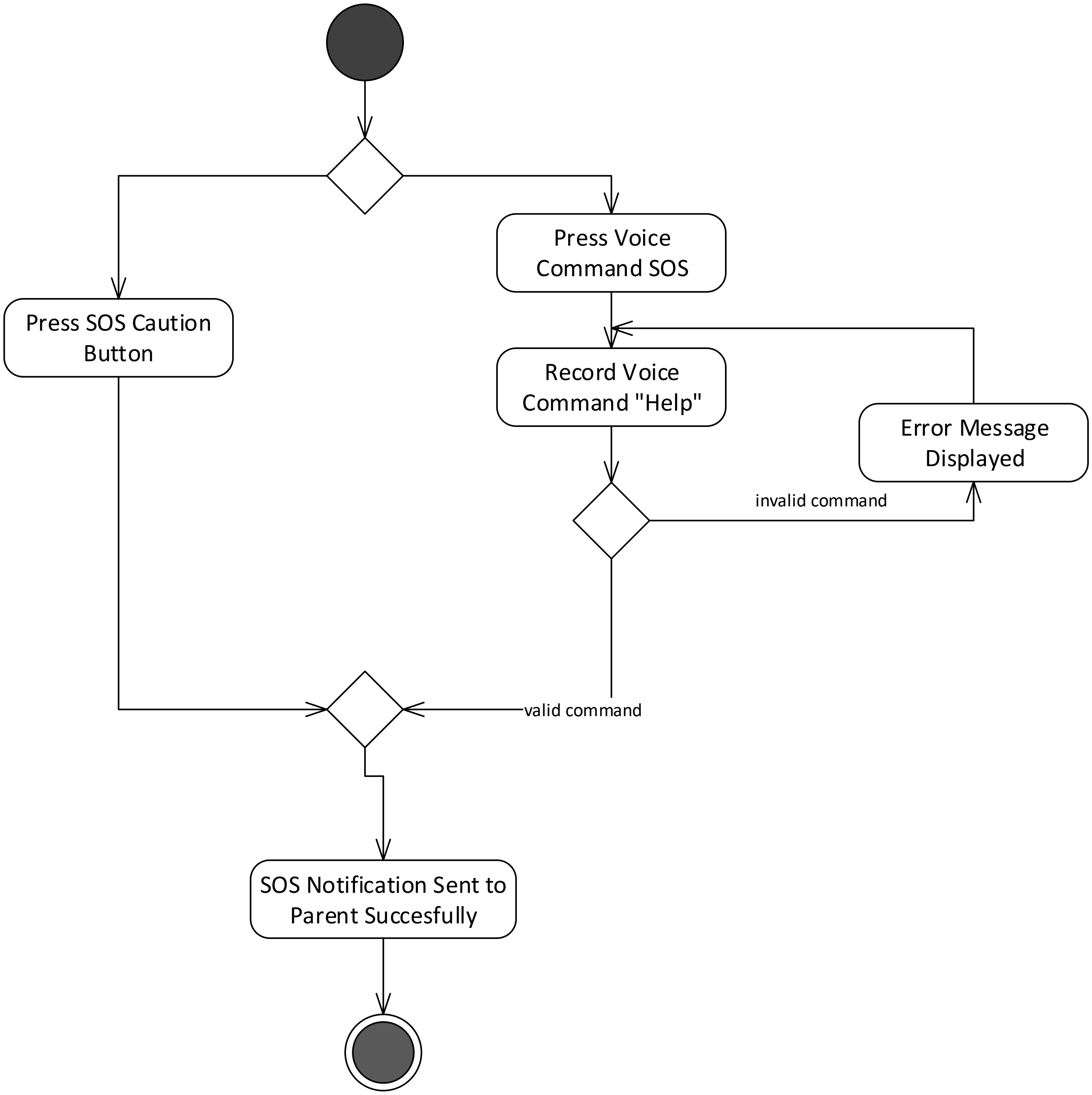


Figure 4.8 Activity Diagram of SOS Caution

4.2.1.8 View App Usage Stats

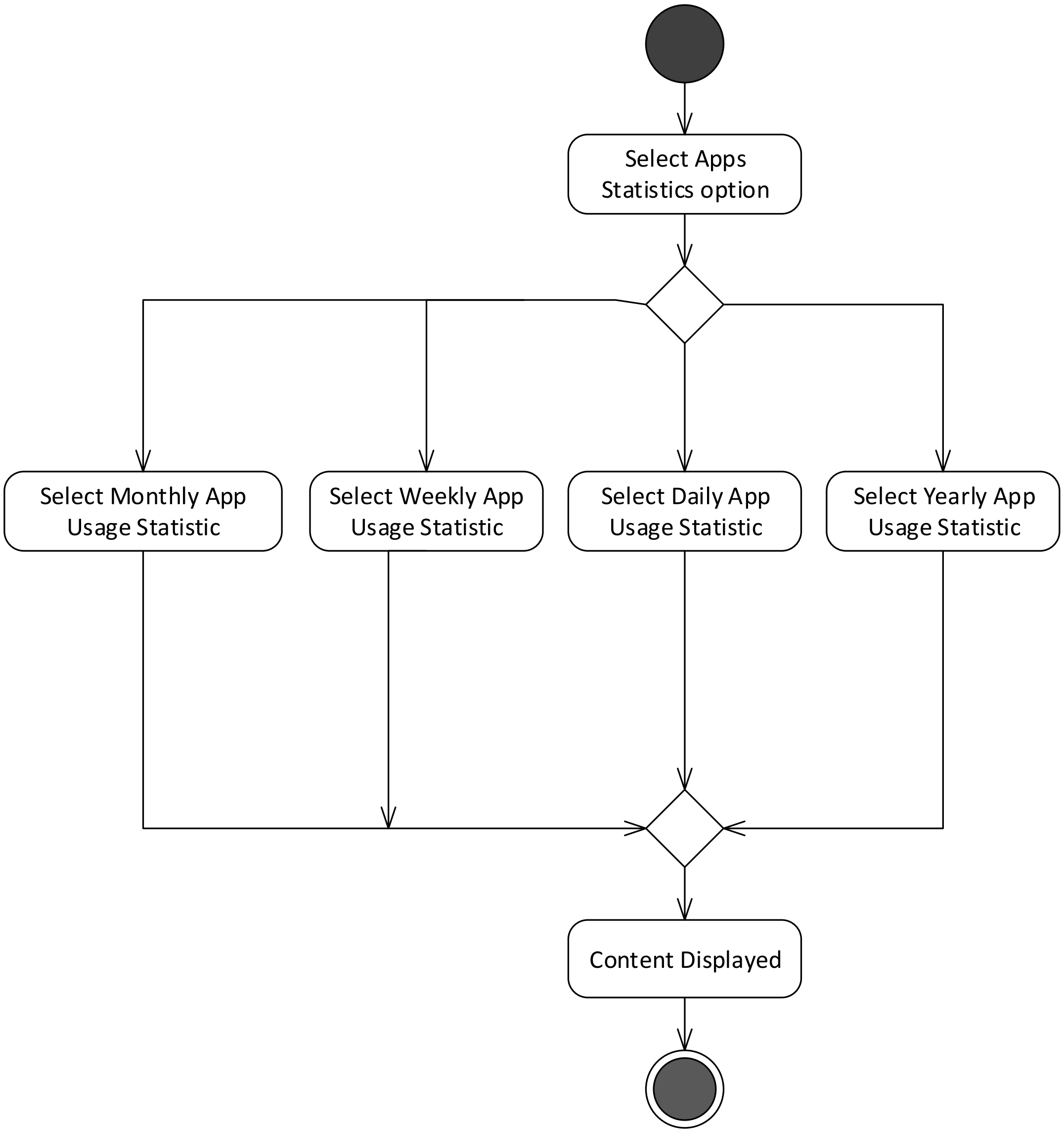

Select Apps Statistics option
Select Monthly App Usage Statistic
Select Weekly App Usage Statistic
Select Daily App Usage Statistic
Select Yearly App Usage Statistic
Content Displayed

Figure 4.9 Activity Diagram of View Usage Stats

4.2.2 Class Diagram

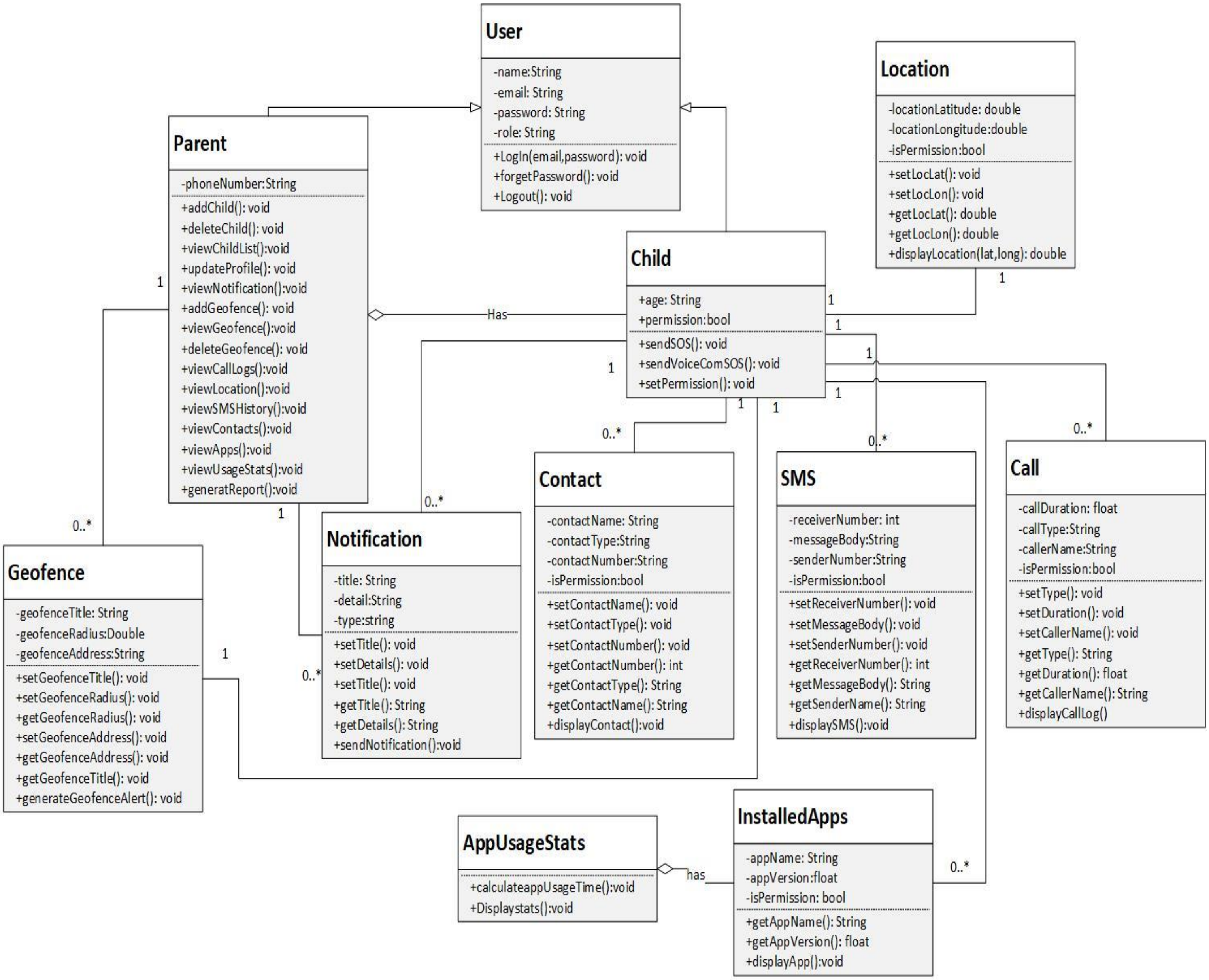


Figure 4.10 Class Diagram of CMS

4.2.3 State Transition Diagram

4.2.3.1 Monitor Location

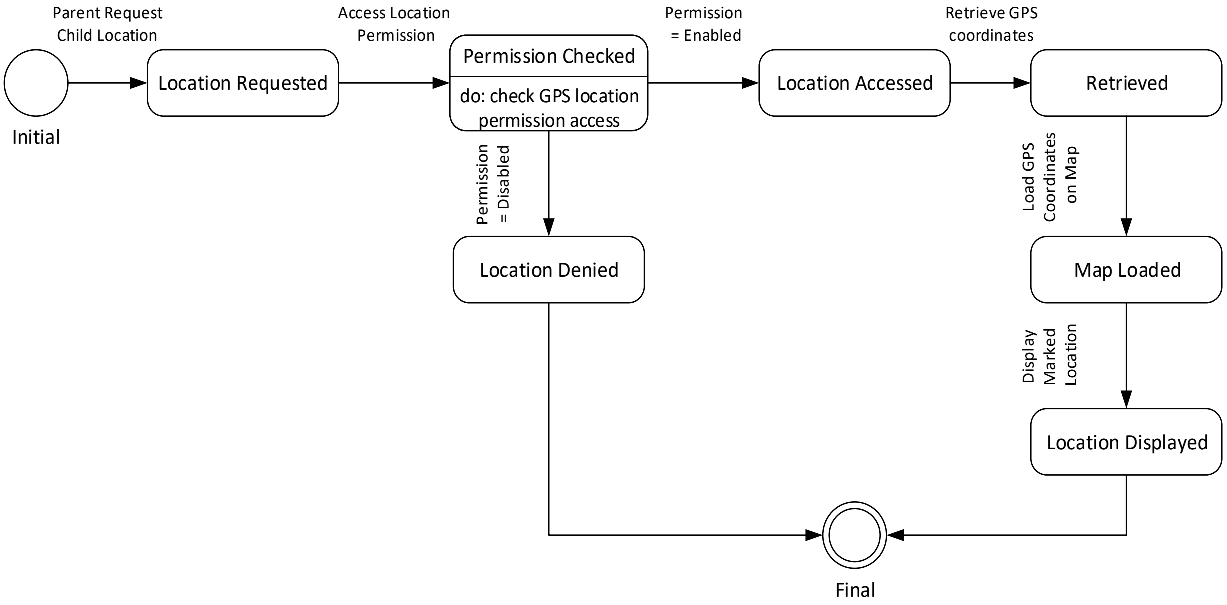


Figure 4.11 State Transition Diagram of Monitor Location

4.2.3.2 Monitor Call

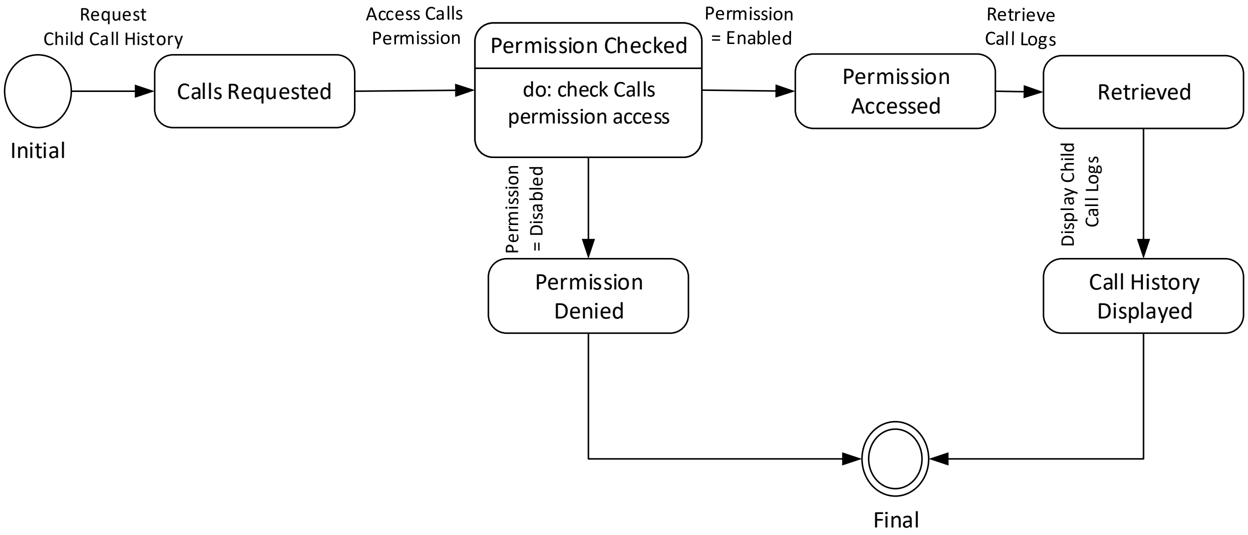


Figure 4.12 State Transition Diagram of Monitor Call

4.2.3.3 Monitor Contact

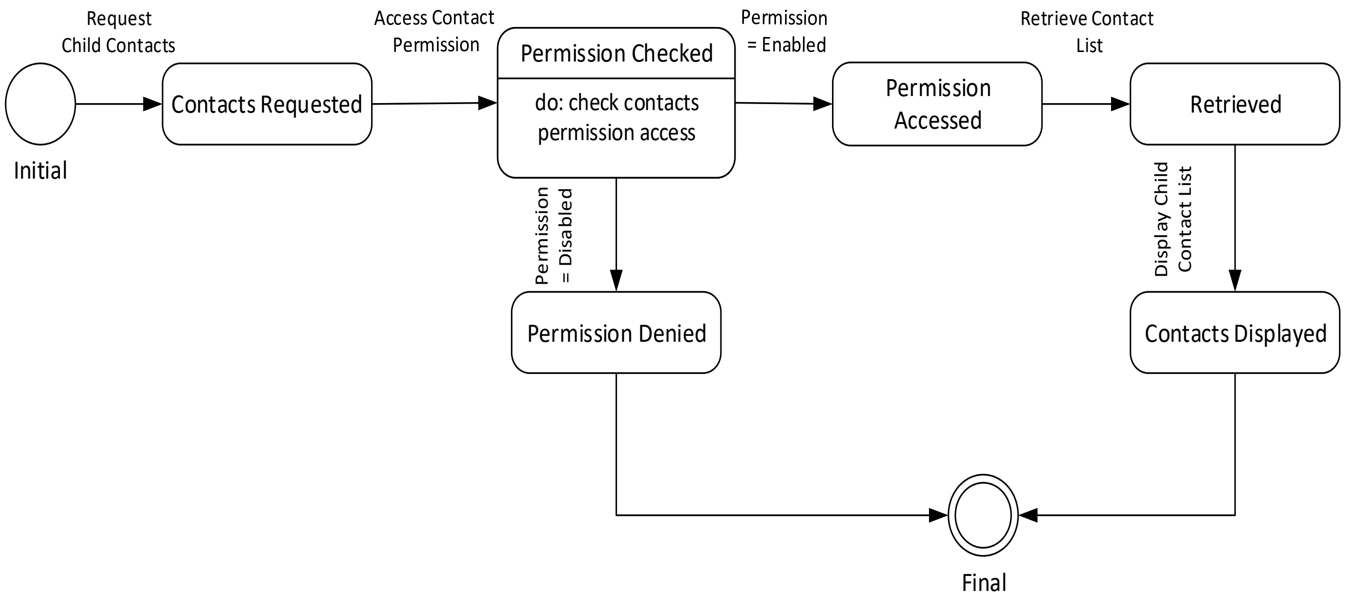


Figure 4.13 State Transition Diagram of Monitor Contact

4.2.3.4 Monitor SMS

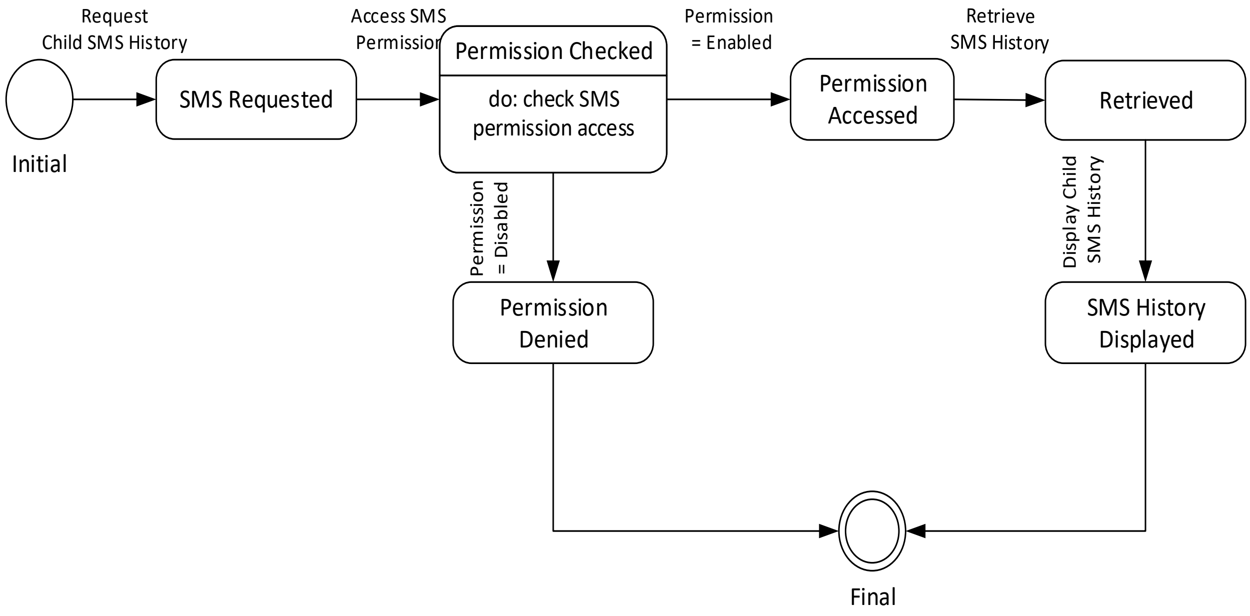


Figure 4.14 State Transition Diagram of Monitor SMS

4.2.3.5 Monitor Install App

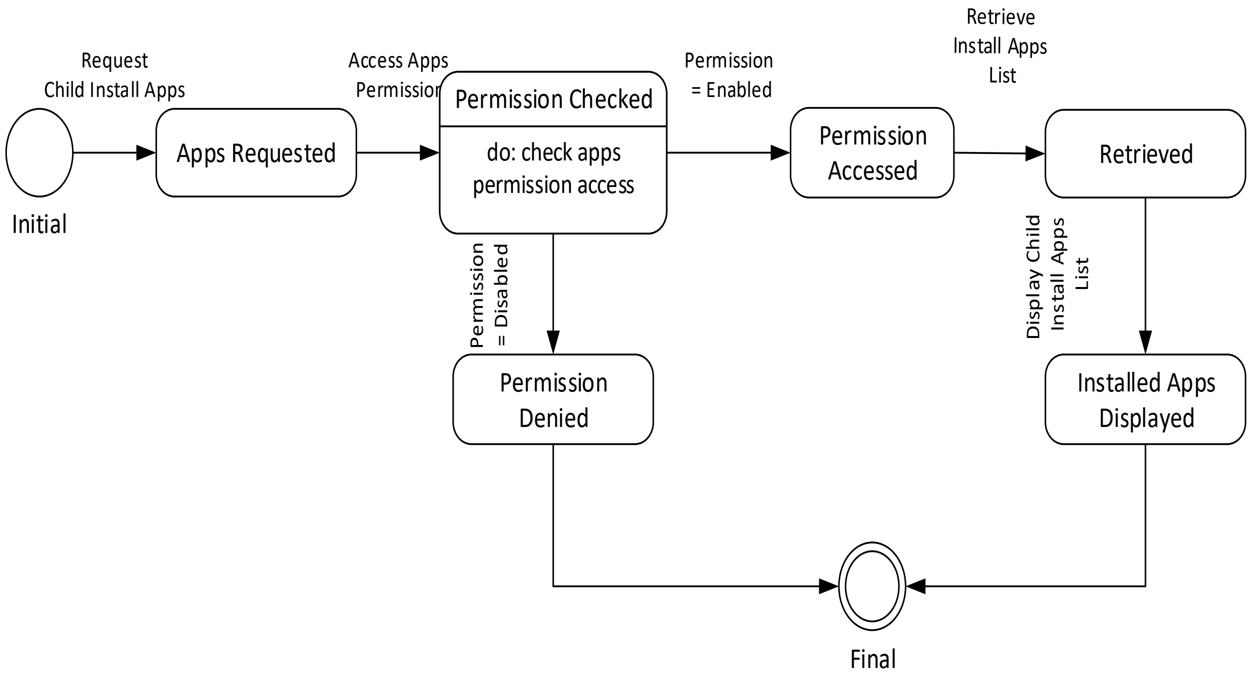


Figure 4.15 State Transition Diagram of Monitor Install App

4.2.3.6 Low Battery Alert

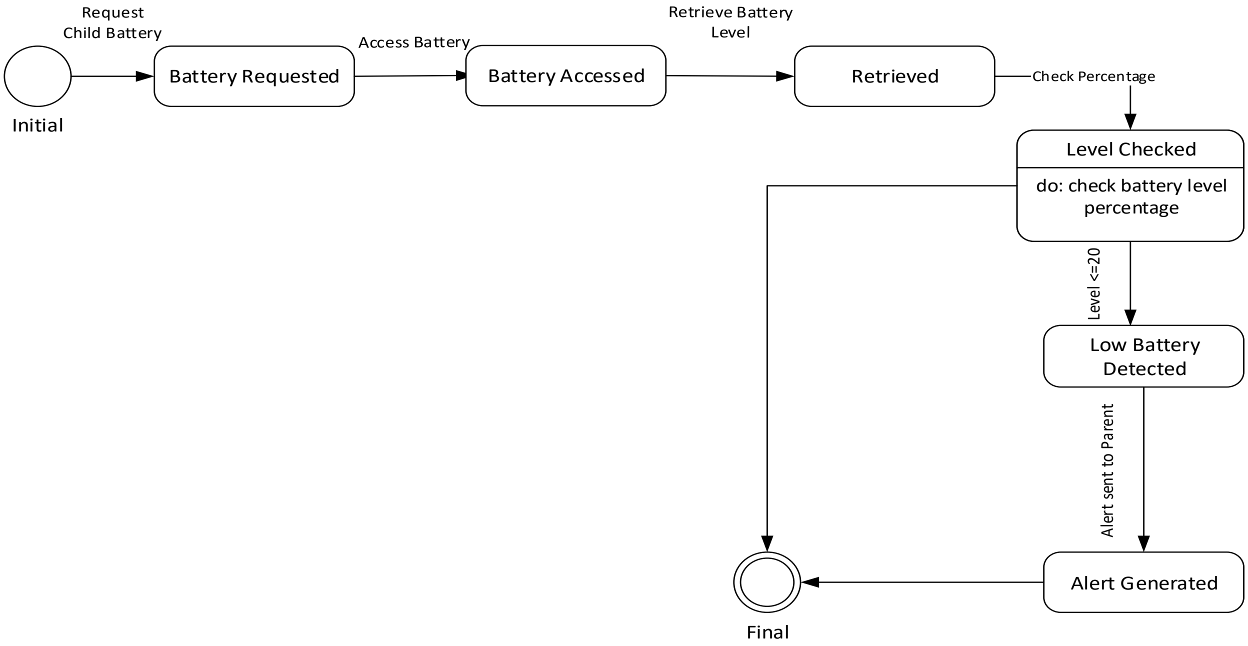


Figure 4.16 State Transition Diagram of Low Battery Alert

4.2.4 Sequence Diagram

4.2.4.1 Add child

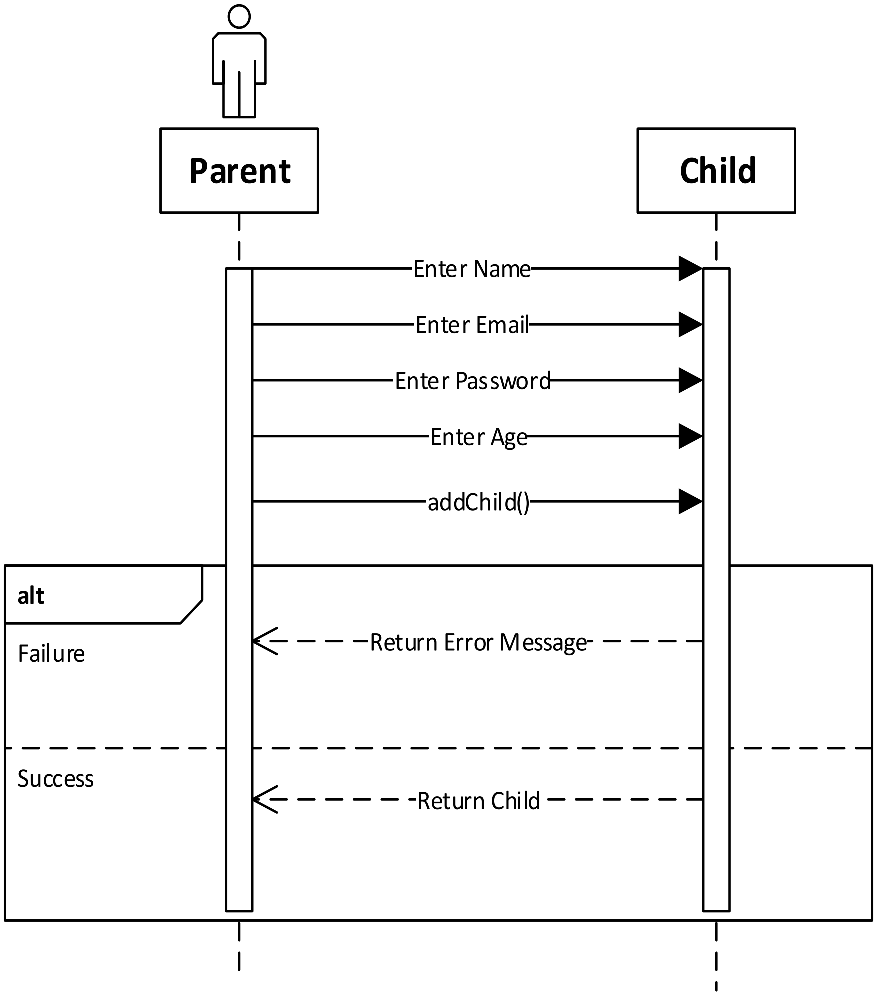


Figure 4.17 Sequence Diagram of Add Child

4.2.4.2 Delete Child

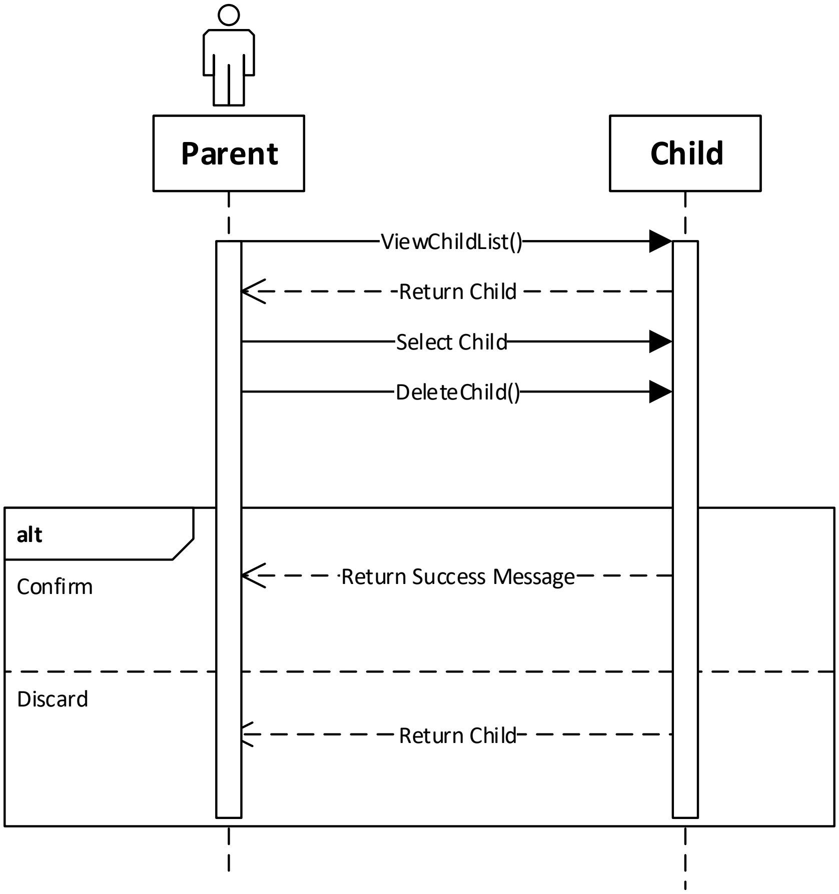


Figure 4.18 Sequence Diagram of Delete Child

4.2.4.3 Add Geo fence

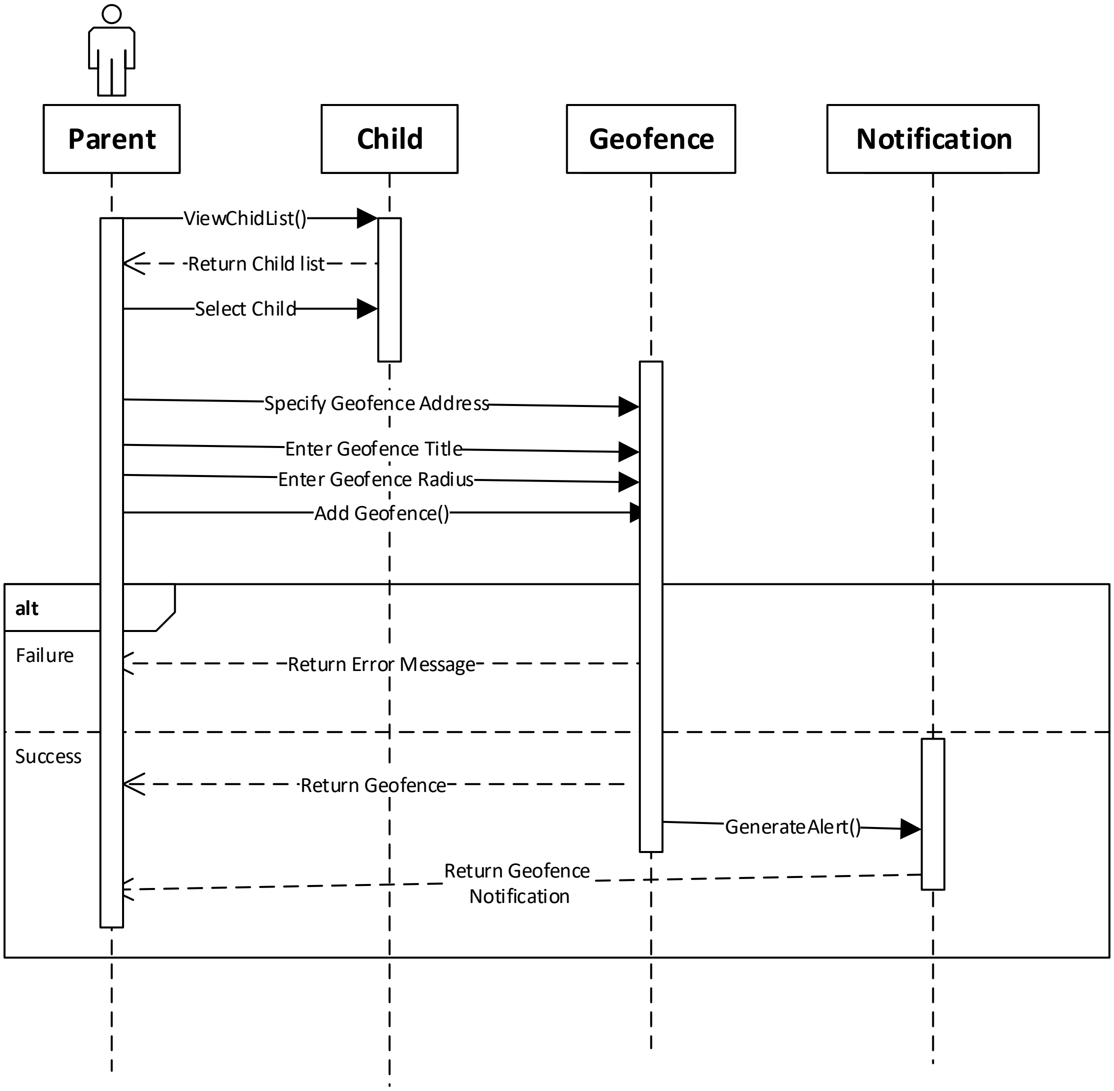


Figure 4.19 Sequence Diagram of Add Geo fence

4.2.4.4 Delete Geo fence

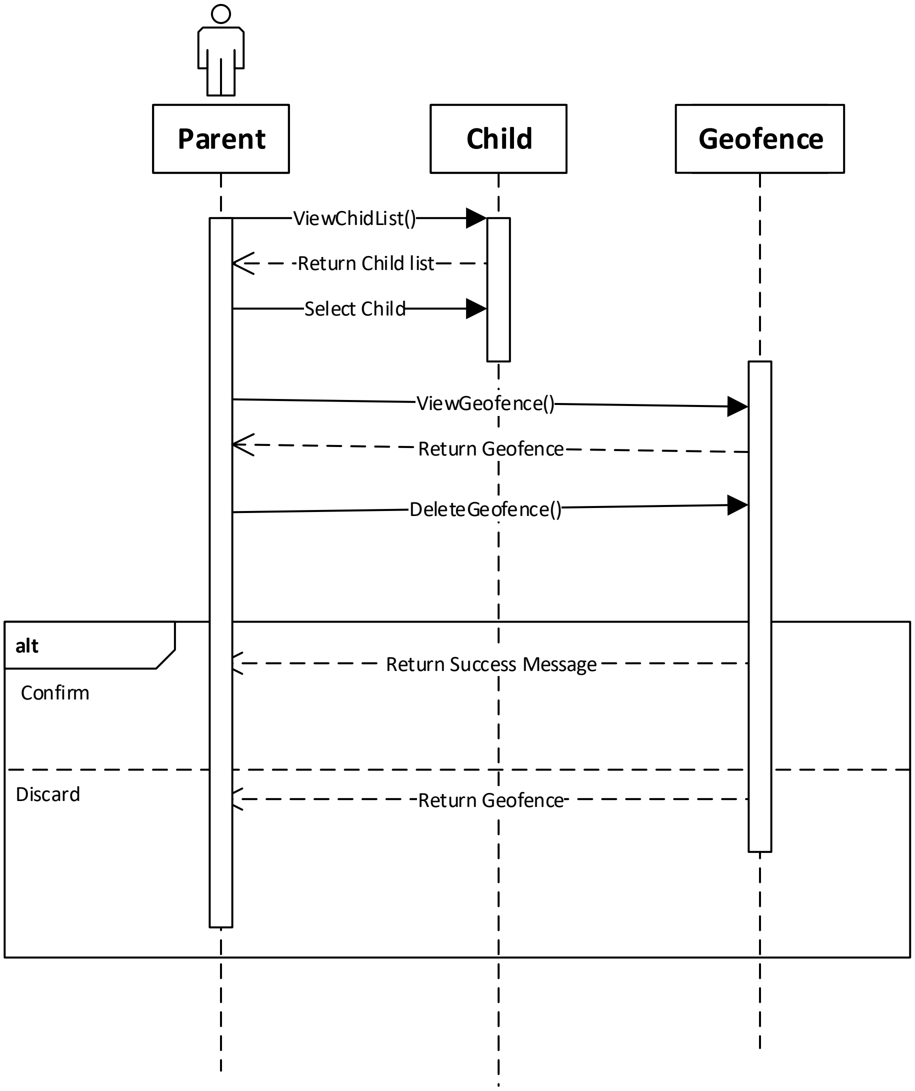


Figure 4.20 Sequence Diagram of Delete Geo fence

4.3 Data Design

As far as database is concerned, we are using the Firebase Firestore database for storing and syncing our data across our mobile application. Firebase database is a real-time and cloud hosted NoSQL database. Other advantages include the supportive SDKs for different platforms like android, iOS and Web. Moreover, the prominent feature is the response to different events activated by database. Last but not the least is its optimization for offline usage.

Below is shown the database dictionary for different database collections and sub collections;

4.3.1 Data Dictionary

4.3.1.1 Parent User

{

***Role**: "PARENT"*

***Email**:"testafshan@gmail.com"*

***Name**:"afshan"*

***Phone**:03338418926*

***userID**:"Mf5qxX4UdBYNXu8i157hI0z7PEB2"*

***Children**: []*

***MyGeofences:** []*

}

Table 4.1 Data Dictionary for Parent

Attribute	**Data Type**	**Description**
Id	String	Unique ID of the Parent
Name	String	Name of the Parent
Password	String	Password set by the Parent, used to get logged into the system
Role	String	Role of the user, in this case it is a Parent
Email	String	Email of the Parent
Phone Number	Number	Phone number of the Parent
Children	Sub Collection	A sub-collection of all child added by parent.
MyGeofence	Sub Collection	A sub-collection of all geo fences added by parent.

{

4.3.1.2 Child User

{

***Role:** "CHILD"*

***Status:** "CREATED"*

***Age:** "15"*

***Email:** "ayesha@gmail.com"*

***Name:** "ayesha"*

***ParentID:** "Mf5qxX4UdBYNXu8i157hI0z7PEB2"*

***Password:** "ayesha12345"*

***SOSMessage:** "Help! Please reach me as soon as possible"*

}

Table 4.2 Data Dictionary for Child

Attribute	**Data Type**	**Description**
Id	String	Unique ID of the Child
Name	String	Name of the Child
Password	String	Password for the child
Age	Number	Age of the Child
Status	String	Status of the Child i.e. ‘Pending ,Created’
Role	String	Role of the user as Child
Email	String	Email of the Child
SOS Message	String	Default SOS Message of the child that is sent in any emergency situations

4.3.1.3 Location History

***Address** :[*

0:

{

***address:**"Park Road Rawal Town ",*

Date: *"2021-11-12"*

Lat: *33.6878783*

Lng: *73.1248183*

Time :*"22:30:24.901824"*

}

]

Table 4.3 Data Dictionary for Location History

Attribute	Data Type	Description
Address	String	Address of the child Location
Date	Date	Date of Child Location
Lat	Number	Latitude of the Child Location
Lng	Number	Longitude of the Child Location
Time	String	Time of Child Location data

4.3.1.4 SMS History

Sms*:*

[0:{

Address*: "Telenor"*

body*: "Apni band sim 923484566869 lga kr 2222 pe call/sms karen aur 60 din k liye roz 50 Telenor Mins + 10GB(raat 12 se shaam 7 tak) 60 din k liye paie SIRF 1 paisa mei"*

date*: "4-Dec-2021 19:33:9"*

}]

Table 4.4 Data Dictionary for SMS

Attribute	Data Type	Description
Address	String	Address of the SMS
Body	String	Message Body of the SMS

Date	string	Date of the SMS

4.3.1.5 Install Apps

Apps: *[0:{*

appName*: "Chrome"*

icon*: "https://firebasestorage.googleapis.com/v0/b/fyp-cms-7ad55.appspot.com/o/images%2F1638696472679258?alt=media&token=e3b2cd6a-8c66-4290-8715-563be477b768"*

packagename: *"com.android.chrome"*

version: *"96.0.4664.45"*

}]

Table 4.5 Data Dictionary for Installed Apps

Field/Attribute	**Data Type**	**Description**
Name	String	Name of the Installed Apps
Icon	String	Icon of the Installed Apps
Package Name	String	Package name of the app.
Version	String	Version of the apps

4.3.1.6 Geo fence

{

childEmail: "child1@gmail.com"

childName: "child1"

geoAddress: "COMSATS Institute of Information Technology, Park Road, Chak Shahzad Campus, Islamabad Capital Territory, Pakistan"

geoName: "university"

latlng: [33.6510265° N, 73.1561505° E]

radius: 250

timestamp: November 29, 2021 at 12:36:16 PM

}

Table 4.6 Data Dictionary for Geo fence

Attribute	**Data Type**	**Description**
GeoAddress	String	Address of the Geofence
GeoName	String	Title name of the Geofence
Radius	Number	Radius of the geofence Address
Timestamp	Timestamp	Date and time of the created geofence

4.3.1.7 Call History

Calls: *[0:{*

callType: *"CallType.incoming"*

displayName: *"SIM1"*

duration: *"21s"*

name: *"Abu"*

number: *"03445218899"*

timestamp: *"3-Dec-2021 19:25:28"*

}

Table 4.7 Data Dictionary for Call History

Attribute	**Data Type**	**Description**
Name	String	Name of the call log Contact
Phone Number	Number	Number of the call log
Type	String	Type of the call log i.e. outgoing, ingoing or missed etc.
Duration	String	Duration of call i.e. hr,min and sec
Timestamp	Timestamp	Date and time of the call log

4.3.1.8 Notification

{

***Body:** "child1 has low battery level: 12 %"*

***Notitype:** "battery"*

***Order:** 1638170196566556*

***Parented:** "Mf5qxX4UdBYNXu8i157hI0z7PEB2"*

***Timestamp:** November 29, 2021 at 12:16:36 PM UTC+5*

***Title:** "Low Battery"*

}

Table 4.8 Data Dictionary for Notification

Attribute	**Data Type**	**Description**
Title	String	Title of the notifications that display on top
Order	Number	Order of the notifications
Body	String	Content of the notifications that shown below title
Notitype	String	Type of the notifications i.e. sos, geo fence and battery
Timestamp	Timestamp	Date and time of the notifications

4.3.1.9 Usage Stats

{

Daily:[0:{

***Minutes:** "416m"*

***Package:** "com.instagram.android"*

***appUsage:** "6:56:20.000000"*

***percent:** 3.571428571428571]}*

weekly: [0:{

```
Minutes: "28m"
appUsage: "0:28:13.000000"
hours: "0h"
package: "com.soundcloud.android"
percent: "0.0%"
]}
Monthly: [0:{
Minutes: "252m"
Package: "com.example.cms"
appUsage: "4:12:41.000000"
percent: 2.380952380952381

]}
Yearly: [0:{
Minutes: "252m"
Package: "com.example.cms"
appUsage: "4:12:41.000000"
percent: 2.38095238095238

]}
}
```

Table 4.9 Data Dictionary for Usage Stats

Attribute	**Data Type**	**Description**
Minutes	String	Minutes for app was used by daily, weekly, monthly and yearly basis.
Package	String	Package Name of application
App Usage	String	Total usage time of the application
Percent	Number	Total percent for which the app was used based on usage time.

4.4 Human Interface Design

4.4.1 Screen Images

Following are some of the mockups which were designed in the early stages;

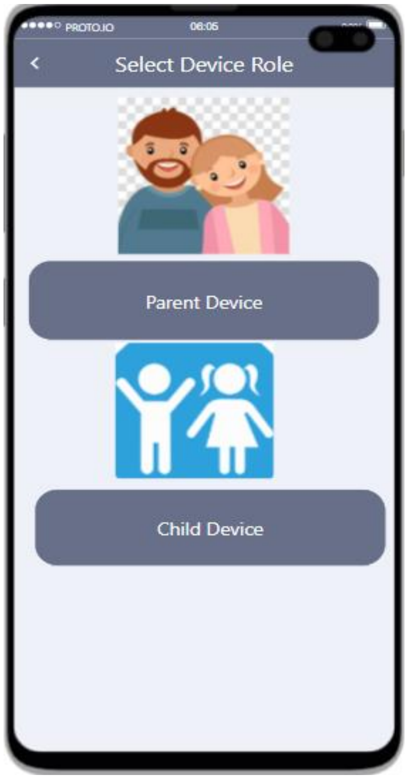


Figure 4.21 Device Role Mockup

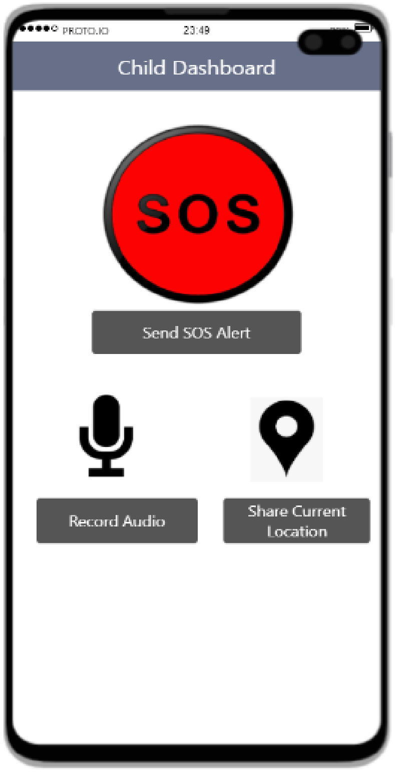
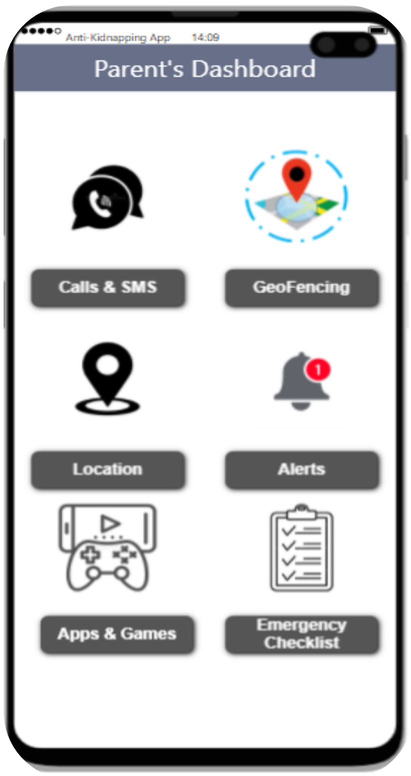


Figure 4.22 Child Dashboard Mockup



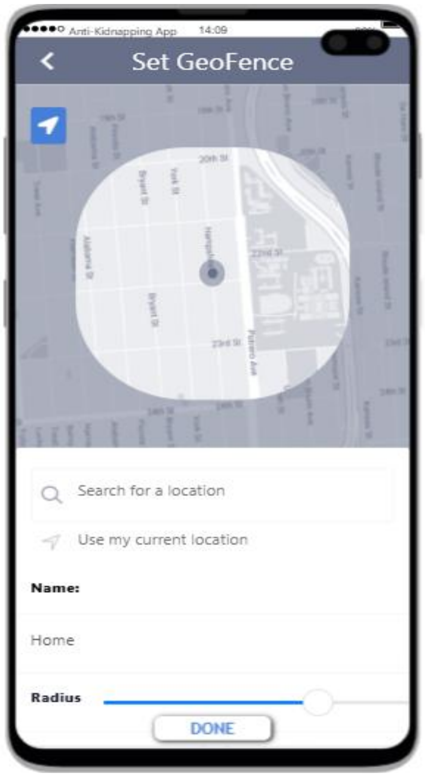


Figure 4.23 Parent Dashboard Mockup **Figure 4.24 Set Geofence Mockup**

4.4.2 Screen Objects and Actions

4.4.2.1 Parent Role Tab

When user select role by tapping Parent tab on Onboard Screen then user will be redirected to the Parent Mode where he can login or create an account in the system as a Parent.

4.4.2.2 Child Role Tab

When user selects role by tapping Child tab on Onboard Screen then user will be redirected to the Child Mode where he can login into the system as a Child.

4.4.2.3 Login Button

When user presses the login button on the login form then it will show success tick on button and will navigate the user to their respective dashboard screen according to their role selected after authentication of user credentials and if the authentication result in failure then it will show error cross on login button and will be reset to login text.

4.4.2.4 Signup Button

When Parent presses the signup button on the signup form then it will show success tick sign on button and will navigate the parent to login screen after authentication of its credentials and if the authentication result in failure then it will show error cross sign on signup button and will be reset to sign up text.

4.4.2.5 Forget Password Link

When user presses the Forget Password text link on Login Screen then the system will navigate the user to the forget password screen where the user can enter email address in the text field for reset password.

4.4.2.6 Parent Bottom Tab

Parent dashboard will consist three bottom tabs navigation i.e. My Child Tab, Child Locations and Notices Tab.

4.4.2.7 Add Child Icon

When parent presses the Add Child Icon on My Child Tab in parent dashboard screen then the system will navigate the user to the add child form where user can enter his/her child details and add them into the system.

4.4.2.8 Menu Drawer Icon

On Parent dashboard menu drawer icon is displayed which when pressed show side drawer that contains Edit Profile, FAQ and Logout buttons.

4.4.2.9 Child Forward Icon

When the parent presses my child tab, child list will be displayed with each child have forward icon. When this icon is pressed the parent will be navigated to feature screen where multiple monitoring feature tabs i.e. location tab, monitor content tab, usage stats tab and geo fence tab are shown on top slider bar.

4.4.2.10 Location Tab

When parent presses on a particular child in my child tab then the parent will be navigated to feature screen where location tab will appear on top slider bar by pressing on Location tab then the parent can view the location of their child on map.

4.4.2.11 Location Marker

When parent presses Live Location Tab on location screen a red marker will be placed on map showing child current location.

4.4.2.12 Location History Calendar

When parent presses location history tab on location screen a calendar with dates will be shown to be selected to view selected date location history.

4.4.2.13 Monitor Content Tab

When parent click on monitor content tab then new tab bar will display which consists of five new tabs i.e. Call log, message, contact, app list from where user can view their respective child device content.

4.4.2.14 Monitor App Stats Tab

When parent click on Usage Stats tab then new tab bar will display which consists of four new tabs i.e. Daily, Weekly, Monthly, Yearly from where user can view their respective child device app usage stats.

4.4.2.15 Geo fence Tab

When parent click on Geo fence tab then new screen will display form where user can add new Geo fence and view details of the already added Geo fence.

4.4.2.16 Add Geo fence Icon

When parent presses the Add Geo fence Icon on Geo fence tab in feature screen then the system will navigate the parent to the add geo fence form where user can enter geo fence details and add them into the system.

4.4.2.17 Radius Slider

A slider radius bar will be shown on add geo fence form which will reduce or increase the size of circle drawn around a specific location on map.

4.4.2.18 Voice SOS Button

When child presses Voice SOS command button on child dashboard screen then system will navigate the user to the new screen from where user can record his voice by pressing on microphone icon and if recorded voice is matched with the text “help” then SOS caution will be send to the parent.

5 Implementation

This section of our report will include the implementation details and describe the user interface of the project. It includes all algorithms, external APIs and libraries that are used by the project.

5.1 Algorithm

Following are some of the algorithm that will be used in the project;

Table 5.1 Algorithms for CMS

Algorithm 1 Signup
Input: name, email ,phone,password,confirmpassword
Output: Successfully Signup message on Valid Credential and Error message on invalid Credential
String msg = ‘’ **if** (name.isEmpty ||

```
   email.isEmpty ||
   password.isEmpty ||
   confirmPassword.isEmpty ||
   phone.isEmpty) then

   msg = ‘Error’
 end if

if(!email.contains("@") || !email.contains(".com")then
   msg = ‘Error’
end if

if (phone.length != 11) then
   msg = ‘Error’
end  if
if  (password.length < 8)then
   msg = ‘Error’
end if
if (password != confirmPassword)then
   msg = ‘Error’
end  if
else
   msg = ‘Success ’
return  msg
```

Algorithm 2 Forget Password

Input: email address

Output: Boolean value i.e. true for send link false for not send link

```
Flag ← false
if ( email.isEmpty) then
   Flag ← false

end if
if(!email.contains("@") || !email.contains(".com")then
   Flag ← false

end if
else
   Flag ← true

return  Flag
```

Algorithm 3 Add Child

Input: child name,email,password,age

Output: Boolean Value i.e. true for child added false for not child added

```
Flag ← false
if (name.isEmpty || email.isEmpty || password.isEmpty || age.isEmpty) then
    Flag ← false
end if
if(!email.contains("@") || !email.contains(".com")then
    Flag ← false
end if
if (age.length > 16 ) then
    Flag ← false
end if
if (password.length < 8)then
    Flag ← false
end if
else
    Flag ← true
return Flag
```

Algorithm4: CheckChildemailAlreadyExists

Input: child email Address

Output: Boolean value i.e. true for exist email false for not exist

```
Flag ← false
if (email.exists)then  // if email exists
    Flag = true
end if
else
    Flag ← false

return Flag
```

Algorithm 5 Generate Pdf Report

Input: Three Data List Cal log list, Sms History List and Location History List
Output: Generated Pdf Contain Three Data Grid

```
PdfGrid Grid1,Grid2,Grid
for (int i = 0; i <= 10; i++) :

    PdfGridRow row = grid.rows.add()

    row.cells[0].value = '${i}';
    row.cells[1].value = mycall[i]["callType"]

    row.cells[2].value = mycall[i]["duration"]
    row.cells[3].value = mycall[i]["number"]
    row.cells[4].value = mycall[i]["timestamp"]
end for
return Grid1

for (int i = 0; i <= 10; i++) :

    PdfGridRow row = grid.rows.add()


    row2.cells[0].value = '${i}'
    row2.cells[1].value = msg[i]['address']
    row2.cells[2].value = msg[i]['date']
    row2.cells[3].value = msg[i]['body']

end for
return Grid2

for (int i = 0; i <= 10; i++) :

    PdfGridRow row50 = grid.rows.add()
    row50.cells[0].value = '${i}'
    row50.cells[1].value = history[i]['date']
    row50.cells[2].value = history[i]['time']
    row50.cells[3].value = history[i]['address']

end for
return Grid3
```

Algorithm 6 Add Geofence	**Algorithm 7 Scan Voice Sos Text**
Input: Geofence name ,radius, location ,latlng	**Input:**Voice Sos command
Output: Boolean Value i.e. true for add and false for not added	**Output:** Boolean value true for scan and false for not scan

Flag ← false
if (geoNameController.text.isEmpty) **then**
Flag ← false
end if
if (location == "Please select a location") **then**
Flag ← false
end if
else
Flag ← true
return Flag
	Flag ← false
	if (text.contains(Command.voicesos) **then**
	Flag ← true
	end if
	else
	Flag ← false
	return Flag

Algorithm 8 Notifiaction **Input** : notitype **OutPut**: Boolean true for send notification and False for not send notification	**Algorithm 9 Location History** **Input**: Location list ,address ,date time **OutPut:** Boolean true for location history list and false for not display location history

Flag ← false
if (notitype) == "Geofence") **then**
return geofencenotification
Flag ← true
end if
if(notitype == ‘Sos’) **then**
return Sosnotification
Flag ← true
if
else if(notitype == ‘batery’) **then**
return baterynotification
Flag ← true
end if
else
Flag ← false
return Flag
	Flag ← false
	for (int i = 0; i <= locations[0].length; i++):
	if (locations[0][i]['date'] == myselecteddate) **then**
	filteredlist.add(locations[0][i])
	Flag ← true
	end if
	end for
	else
	Flag ← false
	return Flag

Algorithm 10 Update Profile **Input** :name ,phone,email **OutPut**:Boolean true for Update and False for not Update	**Algorithm 11 Check Geofence** **Input** :parentid,username,geofencename **Output:**Boolean value i.e true for geofenceservice start false for geofenceservice stop

```
Flag ← false
if (name.isEmpty ||
    email.isEmpty ||
    phone.isEmpty) then
    Flag ← false
end if
if(!email.contains("@") || !email.contains(".com")then
    Flag ← false
end if
if (phone.length != 11) then
    Flag ← false
end if
else
    Flag ← true
return Flag
```

```
Flag ← false
if (geofencedata.exists) then
    geofenceName = geofencedata.data()["geoName"]
    if (geofencingStarted) then
        GeoPoint geoPoint = geofencedata.data()["latlng"]
        double radius = geofencedata.data()["radius"]
    end if
    geofencingStarted = true
    Flag ← true
end if
else
    _geofenceService.stop()
    Flag ← false
return Flag
```

5.2 External APIs/SDKs

Table 5.2: Details of APIs used in the Project

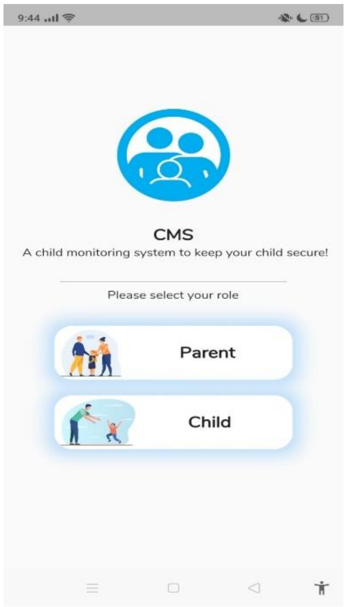
Name of API	**Description of API**	**Purpose of Usage**	**List down the class name in which it is used**
Google Map	This API is used to include map to our mobile application. It provides imagery and local data from Google maps. We can also style our map and can visualize data on map.	We used it to accomplish the feature of "Location Tracking" and Geo fence. Marker of child location is shown to parent. Also, tracking data is shown based on underlying map built using this API. Marker of Geo fence Address is shown on Map with marker and circle to increase and decrease radius.	File: Location.dart AllLocation.dart GeofenceLocation.dart

MapBox	This API is used to get a list of places base on search string.	We are using it to accomplish the feature of setting Geo fence address for child.	File: SearchPlace.dart
Geocoding	This is used to convert latitude longitude value to addresses.	We used it to show the address of child’s location. Whenever parent will press the marker then location’s address will be shown	File: Location.dart AllLocation.dart GeofenceLocation.dart
FCM	Firebase Messaging API will be used to send notifications.	We used it to accomplish the feature of sending various notifications to parent.	File: GeofenceNotification.dart LowBatteryNotification.dart SOSNotification.dart

5.3 User Interface

5.3.1 Role Screen

Select role screen for our android app where user can select his role i.e. parent, child.



A child monitoring system to keep your child secure!

Please select your role

Figure 5.1: Role Screen

5.3.2 Sign up Screen

Sign up screen for our android app where parent can sign up by entering credentials.

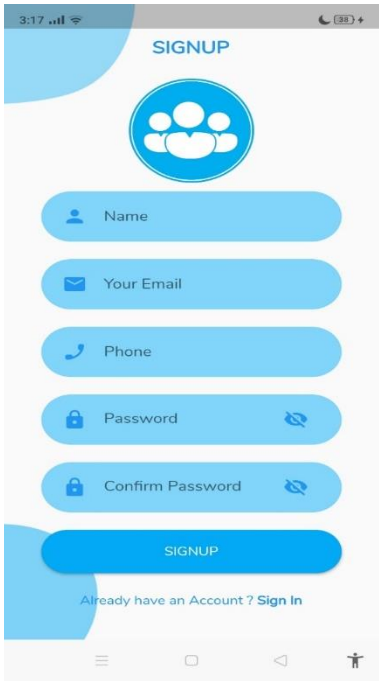


Figure 5.2: Sign up Screen

5.3.3 Login Screen

Login screen for our android app where user can login by entering email and password.

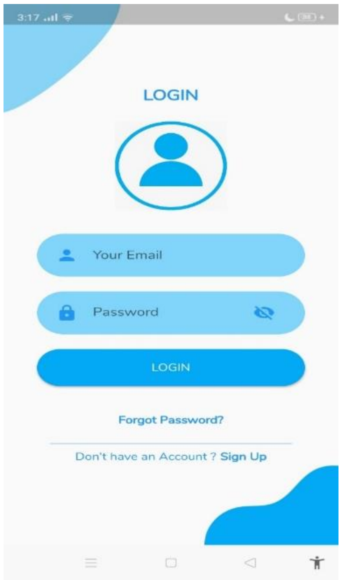
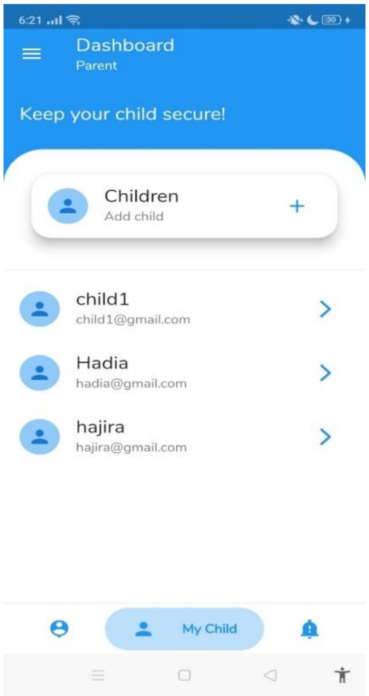


Figure 5.3: Login Screen

5.3.4 Parent Dashboard

Parent Dashboard screen for our android app where parent can monitor child.



child1
child1@gmail.com

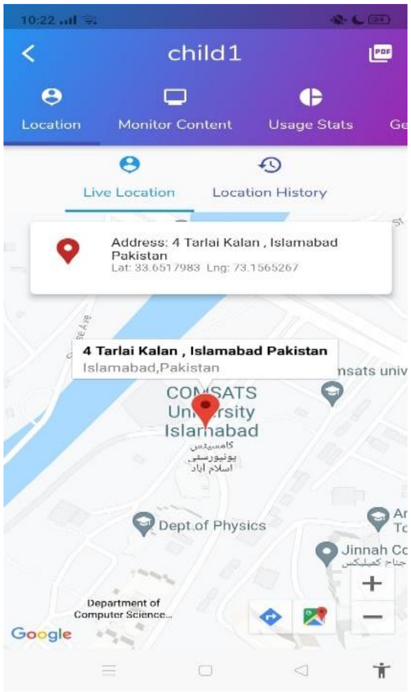
Hadia
hadia@gmail.com

hajira
hajira@gmail.com

Figure 5.4: Sign up Screen

5.3.5 Location Screen

Location Screen for our android app where parent can view child's location.



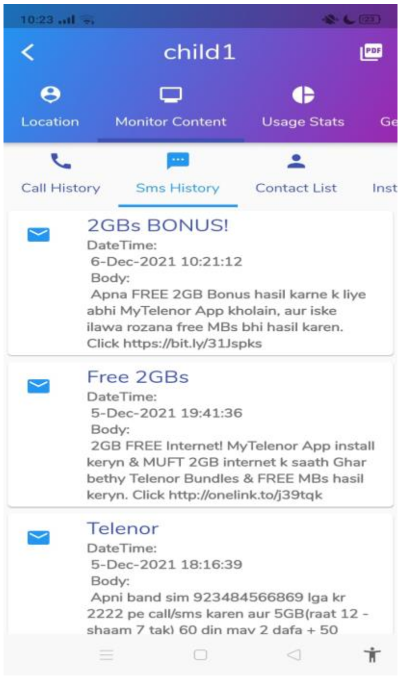
Address: 4 Tarlai Kalan , Islamabad Pakistan
Lat: 33.6517983 Lng: 73.1565267

4 Tarlai Kalan , Islamabad Pakistan
Islamabad,Pakistan

Figure 5.5: Location Screen

5.3.6 Monitor SMS Screen

Monitor SMS Screen for our android app where parent can view child's text history.



2GBs BONUS!
DateTime:
6-Dec-2021 10:21:12
Body:
Apna FREE 2GB Bonus hasil karne k liye abhi MyTelenor App kholain, aur iske ilawa rozana free MBs bhi hasil karen. Click https://bit.ly/31Jspks

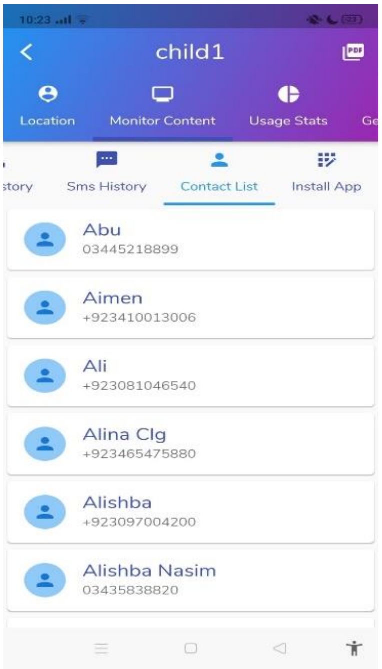
Free 2GBs
DateTime:
5-Dec-2021 19:41:36
Body:
2GB FREE Internet! MyTelenor App install keryn & MUFT 2GB internet k saath Ghar bethy Telenor Bundles & FREE MBs hasil keryn. Click http://onelink.to/j39tqk

Telenor
DateTime:
5-Dec-2021 18:16:39
Body:
Apni band sim 923484566869 lga kr 2222 pe call/sms karen aur 5GB(raat 12 - shaam 7 tak) 60 din mav 2 dafa + 50

Figure 5.6: Monitor SMS Screen

5.3.7 Monitor Contact Screen

Monitor Contact Screen for our android app where parent can view child's contact.

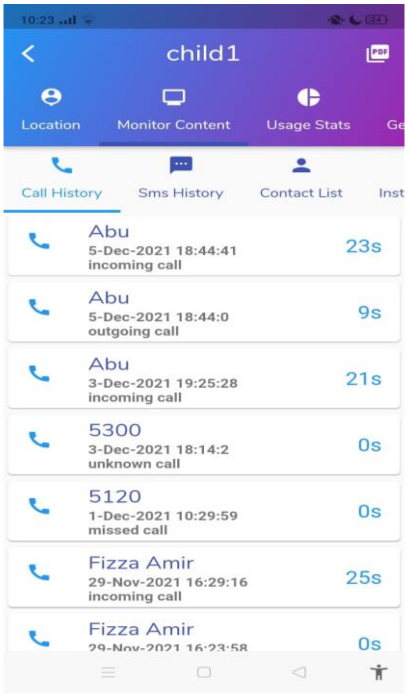


Abu	03445218899
Aimen	+923410013006
Ali	+923081046540
Alina Clg	+923465475880
Alishba	+923097004200
Alishba Nasim	03435838820

Figure 5.7: Monitor Contact Screen

5.3.8 Monitor Call Screen

Monitor Call Screen for our android app where parent can view child’s Call.



Abu
5-Dec-2021 18:44:41
incoming call
23s

Abu
5-Dec-2021 18:44:0
outgoing call
9s

Abu
3-Dec-2021 19:25:28
incoming call
21s

5300
3-Dec-2021 18:14:2
unknown call
0s

5120
1-Dec-2021 10:29:59
missed call
0s

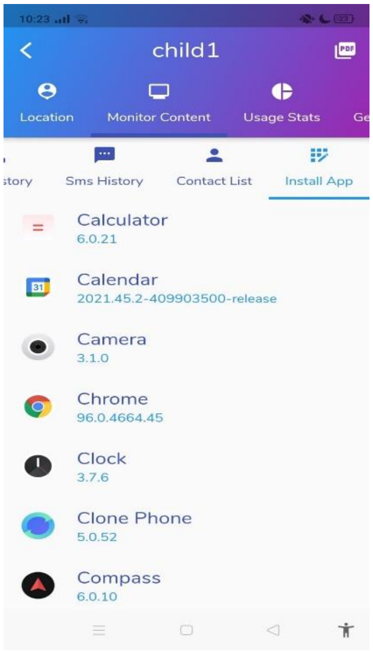
Fizza Amir
29-Nov-2021 16:29:16
incoming call
25s

Fizza Amir
29-Nov-2021 16:23:58
0s

Figure 5.8: Monitor Call Screen

5.3.9 Monitor Install App Screen

Monitor Install App Screen for our android app where parent can view child’s installed app.



Calculator
6.0.21

Calendar
2021.45.2-409903500-release

Camera
3.1.0

Chrome
96.0.4664.45

Clock
3.7.6

Clone Phone
5.0.52

Compass
6.0.10

Figure 5.9: Monitor Install App Screen

5.3.10 App Usage Stat Screen

App Usage Screen for our android app where parent can view child's app usage.

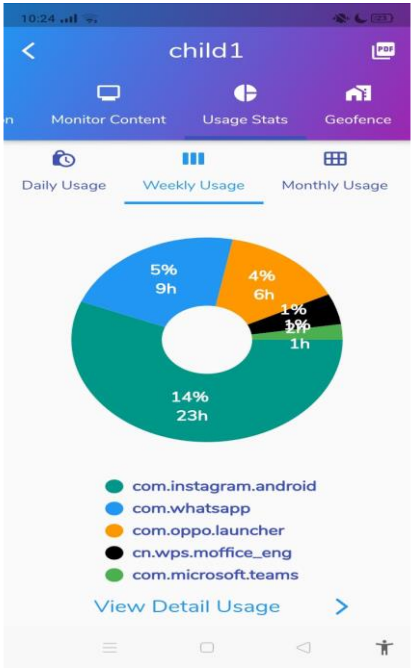
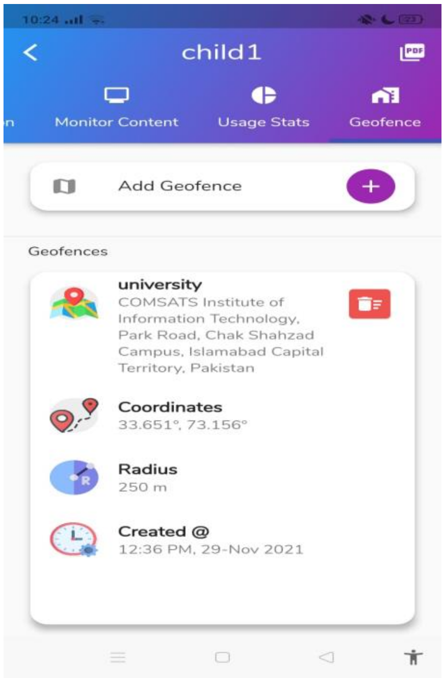


Figure 5.10: App Usage Stats Screen

5.3.11 Geo fence Screen

Geo fence Screen screen for our android app where parent can view child's Geo fence.



university
COMSATS Institute of Information Technology, Park Road, Chak Shahzad Campus, Islamabad Capital Territory, Pakistan

Coordinates
33.651°, 73.156°

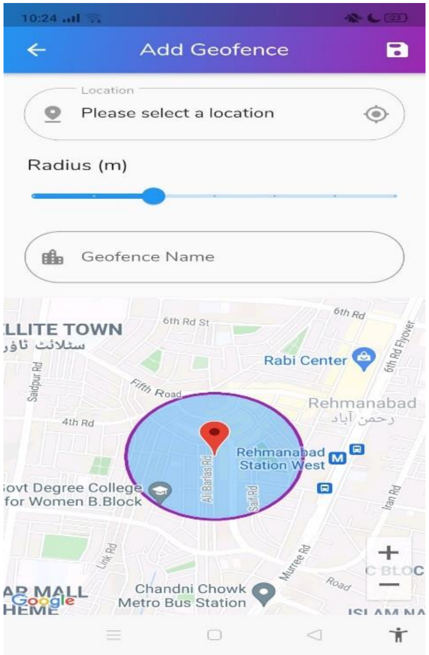
Radius
250 m

Created @
12:36 PM, 29-Nov 2021

Figure 5.11: Geofence Screen

5.3.12 Add Geo fence

Add Geofence Screen screen for our android app where parent can view child’s geo fence.



Add Geofence

Location: Please select a location

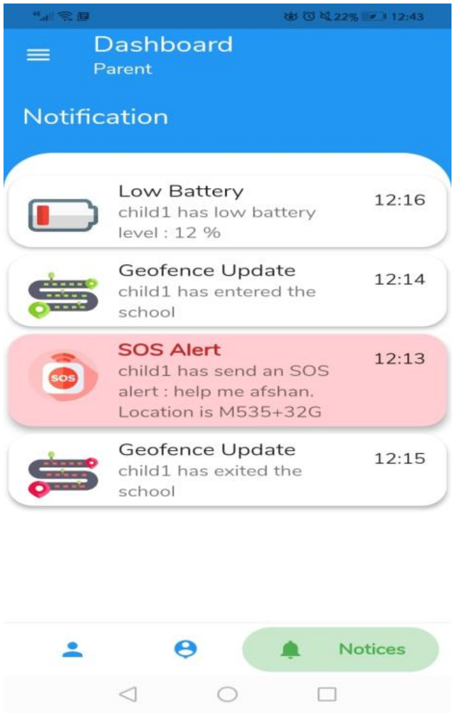
Radius (m)

Geofence Name

Figure 5.12: Add Geofence Screen

5.3.13 Notification Screen

Notification Screen for our android app where parent can view child’s Notification.



Dashboard
Parent

Notification

Low Battery 12:16
child1 has low battery level : 12 %

Geofence Update 12:14
child1 has entered the school

SOS Alert 12:13
child1 has send an SOS alert : help me afshan. Location is M535+32G

Geofence Update 12:15
child1 has exited the school

Figure 5.13: Notification Screen

6 Testing and Evaluation

This section of our report include testing and evaluation of the project to ensure that the system working as intended. It will include multiple testing techniques i.e. unit testing, functional testing, business rule testing and integration testing of the project.

6.1 Unit Testing

Unit Testing 1: Signup

Testing Objective: To ensure the signup form is working correctly.

Table 6.1: Test Cases for Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user's account registration after click on the 'Sign up' button on Sign up form with complete valid input data.	**User Name:** hadia **Email:** testparent35@gmail.com **Phone Number:** 012345567891 **Password:** Hadia123 **Confirm Password:** Hadia123	Successfully register into the account.	As Expected	Pass
2.	Verify user's account registration after click on the 'Sign up' button on sign up form with invalid data	**User Name:** afshan **Email:** afshan@gmail **Phone Number:** 012345567891 **Password:** afshan123 **Confirm Password:** afshan123	Error message is displayed saying Please enter correct email.	As Expected	Pass

Unit Testing 2: Send Verification Email

Testing Objective: To ensure that an email is sent to the inbox of the user's provided email address.

Table 6.2: Test Cases for Send Verification Email

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Check whether the verification link is send to user.	**Email:** testafshan@gmail.com	An email is sent to the email address testafshan@gmail.com containing a link to verify the email.	As Expected	Pass

Unit Testing 3: Login

Testing Objective: To ensure the login form is working properly.

Table 6.3: Test Cases for Login

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user login after click on the 'Login' button on login form with correct input data	**Email:** testparent35@gmail.com **Password:** Hadia123	Successfully log into the account.	As Expected	Pass
2.	Verify user login after click on the 'Login' button on	**Email:** testparent35@gmail.com **Password:**	User will get an error message saying fill all the credentials.	As Expected	Pass

	login form with invalid input.				

Unit Testing 4: Forget Password

Testing Objective: To check that forget password functionality works in the system.

Table 6.4: Test Cases for Forget Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify the system sends a reset link to the user after the user clicks Forget Password and enters his email address.	Email: testafshan@gmail.com	A recovery password email will be sent to the user.	As Expected	Pass

Unit Testing 5: Edit Profile

Testing Objective: To ensure the edit profile form is working properly.

Table 6.5: Test Cases for Edit Profile

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify profile update after click on the 'Edit Profile' button on edit profile form with correct input data	**Username:** Hadia **Phone number:** 0228312391	System displays successfully updated message.	As Expected	Pass

2.	Verify profile update after click on the 'Edit Profile' button on edit profile form with incorrect data.	**Username:** Hadia **Phone number:** 0002	System displays message incorrect phone number.	As Expected	Pass

Unit Testing 6: Add child form

Testing Objective: To ensure the add child form is working properly.

Table 6.6: Test Cases for Add Child Form

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify add child form is displayed after user clicks Add child button	Add child Button is clicked	Successfully display Add Child Form on the Screen	As Expected	Pass

Unit Testing 7: Add child

Testing Objective: To ensure the add child form is working properly.

Table 6.7: Test Cases for Add Child Form

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1	Verify child is added after filling the form with valid credentials.	Name Hadia Age: 10 Email: hadia@gmail.com Password: 1234567890	Successfully new Child Added in Child list.	As Expected	Pass

2.	Verify child is added after filling the form with invalid credentials.	Name abx Age: 10 Email: xxy Password: 12345	System displays appropriate error message.	As Expected	Pass

Unit Testing 8: My Child List

Testing Objective: To ensure that my child list is working properly.

Table 6.8: Test Cases for My Child

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify when the user clicks All Child Tab the child list is shown.	Press My Child Tab	Successfully displays all child list.	As Expected	Pass

Unit Testing 9: Delete Child Dialog

Testing Objective: To ensure the delete child is working properly.

Table 6.9: Test Cases for Delete Child Dialog

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that pop delete dialogue is shown after user long press child name in child list.	User long press child name.	System shows a confirmation delete dialogue box.	As Expected	Pass

Unit Testing 10: Delete Child

Testing Objective: To ensure the delete child is working properly.

Table 6.10: Test Cases for Delete Child

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1	Verify that child is deleted after user clicks ok Button on delete dialog.	User clicks ok button	System displays successfully deleted message.	As Expected	Pass
2.	Verify that child is not deleted after user long press child name on My Child Tab and click Discard on delete dialog	User clicks Discard button	No child is deleted	As Expected	Pass

Unit Testing 11: Show Marker

Testing Objective: To ensure that location marker and info window is working properly.

Table 6.11: Test Cases for Show Marker

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify marker and info window are shown when user clicks on child location tab	Child Location Data, Google Map Data.	Successfully shows marker and info window on map.	As Expected	Pass

Unit Testing 12: Load Map

Testing Objective: To ensure the map loading is working properly.

Table 6.12: Test Cases for Load Map

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that map is loaded after user clicks on child location tab.	Google map data.	Successfully loads map on child location screen.	As Expected	Pass

Unit Testing 13: Show All Child Location

Testing Objective: To ensure the all child location is working properly.

Table 6.13: Test Cases for All Child Location

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that all child location with markers and info window are shown when user clicks on Child Location tab.	Child Location Data, Google Map	Successfully shows all added child location on map.	As Expected	Pass

Unit Testing 14: View Live Location

Testing Objective: To ensure the live location track is working properly.

Table 6.14: Test Cases for View Live Location

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that live location updates are show on map after user clicks on Live Location tab.	Child Location Data, Google Map	Successfully display child live location updates on map.	As Expected	Pass

Unit Testing 15: View Location History

Testing Objective: To ensure the location history is working properly.

Table 6.15: Test Cases for View Location History

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that location history is shown after user selects a date on Location History tab.	Selected date, Child location data	Successfully shows location history date wise to the user.	As Expected	Pass

Unit Testing 16: Unavailable Location

Testing Objective: To ensure the unavailable location is working properly.

Table 6.16: Test Cases for Unavailable Location

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that system displays unavailable location when location is not found on child location screen.	Child data	Successfully shows unable to find location message to user.	As Expected	Pass

Unit Testing 17: View Call History

Testing Objective: To ensure the monitor call history is working properly.

Table 6.17: Test Cases for View Call History

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that call history is shown after user clicks on Call History tab in monitor content screen.	Child calls data	Successfully displayed call logs to parent.	As Expected	Pass

Unit Testing 18: View SMS History

Testing Objective: To ensure the SMS History is working properly.

Table 6.18: Test Cases for View SMS History

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended

1.	Verify that SMS history is shown after user clicks on SMS History tab in monitor content screen.	Child SMS data	Successfully displayed child SMS to parent.	As Expected	Pass

Unit Testing 19: View Contact List

Testing Objective: To ensure the monitor contact is working properly.

Table 6.19: Test Cases for View Contact List

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that child contacts are shown after user clicks on Monitor Contact tab in monitor content screen.	Child contacts data	Successfully displayed child contact to parent.	As Expected	Pass

Unit Testing 20: View Device Apps

Testing Objective: To ensure monitoring of Device Apps is working properly.

Table 6.20: Test Cases for View Device Apps

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that Device Apps is shown after user clicks on Monitor Apps tab in monitor content screen.	Child Apps Name, Apps Icon data	Successfully displayed child Apps to parent.	As Expected	Pass

Unit Testing 21: Show Daily App Usage

Testing Objective: To ensure the daily app usage is working properly.

Table 6.21: Test Cases for Show Daily Usage

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the child daily most used app is shown after user clicks on Daily Usage Tab in Usage Statistics screen.	Child apps usage data	Successfully displays child daily usage time spent on apps to parent	As Expected	Pass

Unit Testing 22: Show Weekly App Usage

Testing Objective: To ensure that weekly app usage is working properly.

Table 6.22: Test Cases for Show Weekly Usage

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the child weekly most used app is shown after user clicks on weekly usage tab in Usage Statistics screen.	Child apps usage data	Successfully displays child weekly usage time spent on apps to parent.	As Expected	Pass

Unit Testing 23: Load Usage Chart

Testing Objective: To ensure that weekly usage chart is working properly.

Table 6.23: Test Cases for Load Usage Chart

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended

1.	Verify that pie chart is displayed for child weekly apps usage percentage and hours after user clicks Weekly Stats Tab.	Child Apps name, usage data	Successfully loads and displayed pie chart of child most weekly used apps to parent.	As Expected	Pass

Unit Testing 24: Show Monthly App Usage

Testing Objective: To ensure that Monthly app usage is working properly.

Table 6.24: Test Cases for Show Monthly Usage

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the child Monthly most used app is shown after user clicks on Monthly usage tab in Usage Statistics screen.	Child apps usage data	Successfully displays child Monthly usage time spent on apps to parent.	As Expected	Pass

Unit Testing 25: Show Yearly App Usage

Testing Objective: To ensure that Yearly app usage is working properly.

Table 6.25: Test Cases for Show Yearly Usage

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the child Yearly most used app is shown after user clicks	Child apps usage data	Successfully displays child Yearly usage time spent on apps to parent.	As Expected	Pass

	on Yearly usage tab in Usage Statistics screen.				

Unit Testing 26: Add Geofence

Testing Objective: To ensure the add geofence form is working properly.

Table 6.26: Test Cases for Add Geofence

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify add geofence form is displayed after user clicks Add geofence icon	Add Geofence icon clicked.	Successfully display Add geofence Form on the Screen	As Expected	Pass

Unit Testing 27: Save Geofence

Testing Objective: To ensure the save geofence form is working properly.

Table 6.27: Test Cases for Save Geofence

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1	Verify geofence is save after save button is clicked with correct data	Name: childgeofence radius: 300 address: comsats university	Successfully new Geofence saved in the Database	As Expected	Pass

Unit Testing 28: View Geo fence

Testing Objective: To ensure the view Geo fence functionality is working correctly

Table 6.28: Test Cases for View Geo fence

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1	Verify geo fence details display in the list	**address:** Comsats university **coordinates:** 33.5,73.15 **radius:** 250m **status:** created	Successfully geofence details display in the list	As Expected	Pass

Unit Testing 29: Set geo fence radius range

Testing Objective: To ensure the geo fence radius slider bar range is working correctly

Table 6.29: Test Cases for Set geo fence radius range

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1	Verify that geo fence radius range is set after user slides the radius range.	**min range:** 200 **max range:** 500	Successfully min and max range shown on geo fence radius slider bar.	As Expected	Pass

Unit Testing 30: Show map circle

Testing Objective: To ensure the circle shown on map when user set geo fence radius

Table 6.30: Test Cases for show map circle

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1	Verify circle draw on the map according to geo fence radius when user set radius	**Geofence radius range:** 250	Successfully drawn circle of radius range 250	As Expected	Pass

Unit Testing 31: Delete Geo fence

Testing Objective: To ensure the delete geo fence is working properly.

Table 6.31: Test Cases for Delete Geo fence

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that pop delete dialogue is shown after user press delete icon.	Delete icon	System shows a confirmation delete dialogue box.	As Expected	Pass
2	Verify that geofence is deleted after user clicks “Yes” Button.	User clicks Yes button	System displays successfully deleted message.	As Expected	Pass
3.	Verify that geofence is not deleted after user on “No” button	User clicks No button	No geofence is deleted	As Expected	Pass

Unit Testing 32: Send SOS notifications

Testing Objective: Verify that SOS notification is send to the parent.

Table 6.32: Test Cases for Send SOS notifications

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the parent receives SOS notification when child clicks SOS Button.	Child SOS message, Location Data	System display SOS notification to parent.	As Expected	Pass

Unit Testing 33: Send low battery notifications

Testing Objective: Verify that low battery notification is send to parent

Table 6.33: Test Cases for Send low battery notifications

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the parent view low battery notifications	Child device battery level data.	System display low battery notification to parent	As Expected	Pass

Unit Testing 34: Send check in geo fence notifications

Testing Objective: Verify that check in geo fence region alert view by parent

Table 6.34: Test Cases for Send check in geo fence notifications

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the parent view and receive check in alert when child enter geo fence area.	Child location data, Geo fence data.	System display check in alert notification to parent	As Expected	Pass

Unit Testing 35: Send check out(exit) geo fence notifications

Testing Objective: Verify that check out alert is send to parent

Table 6.35 Test Cases for Send check out geo fence notifications

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the parent view checks out alert when child exist geo fence area	Child location data, Geo fence data	System display check out alert notification to parent.	As Expected	Pass

Unit Testing 36: Download report

Testing Objective: Verify that report is downloaded and displayed to user.

Table 6.36 Test Cases for Download report

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the parent view report when click on report tab	Report tab	System downloads the report successfully.	As Expected	Pass

Unit Testing 37: Generate content report

Testing Objective: Verify that content report is generated

Table 6.37 Test Cases for Generate content report

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the content report is generated when user click on content report button on report screen	content report button	System generate content report	As Expected	Pass

Unit Testing 38: Generate usage report

Testing Objective: Verify that usage report is generated

Table 6.38 Test Cases for Generate usage report

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the usage report is generated when user click on usage report button on report screen	Usage report button	System generated usage report	As Expected	Pass

Unit Testing 39: Display voice SOS

Testing Objective: Verify that voice SOS is loaded

Table 6.39 Test Cases for Voice SOS Screen

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the voice sos page is display when child press voice sos tab	Voice sos tab	System display voice sos screen.	As Expected	Pass

Unit Testing 40: Record voice

Testing Objective: Verify that voice record when child press mic icon

Table 6.40 Test Cases for Record voice

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the voice is record when child press mic icon	mic icon	System record voice	As Expected	Pass

Unit Testing 41: Display voice text

Testing Objective: Verify that voice to text functionality working

Table 6.41 Test Cases for Display voice text

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the voice text display on top when user speak on mic	mic icon	System display text of the voice	As Expected	Pass

Unit Testing 42: Scan 'help' command

Testing Objective: Verify that help command is scan when user speak help on mic

Table 6.42 Test Cases for Scan help command

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the help command is highlighted when child speak help on mic.	mic icon command = 'help'	System display highlighted help text .	As Expected	Pass

Unit Testing 43: Display FAQ screen

Testing Objective: Verify that the FAQ screen is display on the screen

Table 6.43 Test Cases for Display FAQ screen

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the FAQ screen display when user press FAQ tab from drawer menu	FAQ tab	System display FAQ page on screen	As Expected	Pass

Unit Testing 44: Logout as Parent

Testing Objective: Verify if the user is signing out when log out button is clicked

Table 6.44: Test Cases for Log out as Parent

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended

1.	Verify that parent is logout and redirected to the role screen when 'Logout' button is pressed	Logout Button	When parent logout, system shows role screen.	As Expected	Pass

Unit Testing 45: Logout as Child

Testing Objective: Verify if the user is signing out when log out button is clicked

Table 6.45: Test Cases for Log out as Child

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that child is sign out and redirected to the role screen when 'Logout' button is pressed	Logout Button	System displays the role screen screen.	As Expected	Pass

6.2 Functional Testing

Functional Testing 1: Login with different roles

Testing Objective: To ensure that user management functionality is working correctly.

Table 6.46: Test Cases for Login with different roles

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Login as the 'Parent'.	Email: testparent35@gmail.com Password: Testparent35	Dashboard for the parent is loaded.	User is redirected to the "Parent Dashboard" screen	Pass
2.	Login as the "Child"	Email: Testchild1@gmail.com Password: Testchild1	Dashboard for the child is loaded	User is redirected to the "Child Dashboard" screen	Pass

Functional Testing 2: Manage Geofence

Test Objective: To make sure that Manage Geofence functionality is working correctly

Table 6.47: Test Cases for Manage Geofence

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify Parent added geofence with all input credentials .	**address:** comsats university **radius:** 250m **title:** my university.	Geofence successfully added	As Expected	Pass
2.	Verify parent save geofence details after click on save button	Save button	Geofence details successfully saved	As Expected	Pass
3	Verify parent delete geofence after click on delete icon	Delete icon	Geofence deleted successfully	As Expected	Pass

Functional Testing 3: Manage Child

Test Objective: To make sure that Manage Child fence functionality is working correctly

Table 6.48: Test Cases for Manage Child

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify parent added child with all valid input credentials	**name:** hadia email: hadia@gmail.com age: 14 phone: 03498964154	Child successfully added	As Expected	Pass
2	Verify parent delete child after long press on child name	childname: hadia	Child deleted successfully deleted.	As Expected	Pass

3.	Verify that parent view childlist after click on mychild tab	Mychildtab	Child list is successfully displayed	As Expected	Pass

Functional Testing 4: Display Child content

Test Objective: Display all child mobile content monitoring data in respective data list.

Table 6.49: Test Cases for Content Monitoring

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	To check that messages data list should be filled with child mobile messages after user has pressed the "SMS History" Tab	Child SMS Data List	The data list must display all the messages of child phone.	Data List has been populated with messages.	Pass
2.	To check that contacts data list should be filled with child mobile contacts after user has pressed the "contacts List" Tab.	Child Contact Data List	The data list must display all the Contacts of child phone.	Data List has been populated with Contacts.	Pass
3.	To check that calls data List should be filled with child mobile calls after user has pressed the " Calls History" Tab.	Child Call Data List	The data list must display all the Call Logs History of child phone.	Data List has been populated with Calls Log	Pass
4.	To check that Apps data List should be filled with child mobile Apps after user has pressed the "App List " Tab	Child App Data List	The data list must display all the Device App List of child phone.	Data List has been populated with Device App List	Pass

Functional Testing 5: Display child Usage Stats

Test Objective: To ensure that correct stats are displayed daily, weekly, yearly, monthly.

Table 6.50: Test Cases for Display Usage Stats

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the child daily most used app is shown after user clicks on Daily Usage Tab in Usage Statistics screen.	Child Daily Usage Data List	The data list must display all the daily usage apps data of child phone.	Data List has been populated with daily usage stats.	Pass
2.	Verify that pie chart is displayed for child weekly apps usage percentage and hours after user clicks Weekly Stats Tab.	Child weekly Usage percentage	Pie chart of child most weekly used apps of child phone must be displayed.	Pie Chart has be made with weekly usage stats.	Pass
3.	Verify that the child detailed list about most used weekly app is shown after user clicks on Detail Weekly Usage Tab in Usage Statistics screen.	Child Weekly Usage Data list	The data list must display all the most used weekly usage apps data of child phone.	Data List has been populated with weekly usage stats.	Pass
4.	Verify that the child monthly most used app is shown after user clicks on Monthly Usage Tab in Usage Statistics screen.	Child Monthly Usage Data List	The data list must display all the monthly usage apps data of child phone.	Data List has been populated with monthly usage stats.	Pass
5.	Verify that the child yearly most used app is shown after user clicks on Yearly Usage Tab in Usage Statistics screen.	Child yearly Usage Data List	The data list must display all the yearly usage apps data of child phone.	Data List has been populated with yearly usage stats.	Pass

Functional Testing 6: Reporting

Testing Objective: To ensure the reporting works properly for app usage and content monitoring and location of child.

Table 6.51: Test Cases for Display Usage Stats

No.	Test case	Attribute and value	Expected result	Result
1.	Verify emergency report is generated when user click on 'emergency report " tab	Emergency report Data	Successfully generated the emergency report (call log, sms, and location history of child device).	Pass
2.	Verify app usage report is generated when user click on 'usage report " tab	App usage Usage Data	Successfully generated app usage report	Pass

Functional Testing 7: Notifications

Testing Objective: To ensure the notification between parent and child works properly

Table 6.52: Test Cases for Notifications

No.	Test case	Attribute and value	Expected result	Result
1.	Verify SOS notification send to parent when child press SOS button	Child Location data, SOS data.	Successfully sent SOS notification to parent.	Pass
2.	Verify parent can view SOS notifications when press notification tab	Notices Tab	System displays the SOS notification on Notices Tab.	Pass
3.	Verify Geo fence notification is send to parent when child enters the region.	Child Location, Geo fence Data	System sent check in alert notification to parent.	Pass
2.	Verify parent can view Check in Geofence	Notices Tab	System displays the check in Geo fence notification on Notices Tab.	Pass

	notifications when press notification tab			
4.	Verify Geo fence notification is send to parent when child enters the region.	Child Location, Geo fence Data	System sent check out alert notification to parent.	Pass
5.	Verify parent can view check out Geo fence notifications when press notification tab	Notices Tab	System displays the Checkout Geo fence notification on Notices Tab.	Pass

6.3. Business Rules Testing

Business Rule Testing 1: Add Child

Testing Objective: Parent can’t create an account for child with age under 13 and above 18.

Table 6.53: Decision Table for BR-1

Condition	Rule 1	Rule 2	Rule 3
Child Age<18?	Y	Y	N
Child Age>13?	Y	N	Y
Create child account	Y	N	N

Table 6.54: Test Cases for Add Child

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that child account is created if the age of child is less than 18 and greater than 13.	Child age: 15	Child account will be created.	As Expected	Pass
2.	Verify the child account is not created if the age of child is less than 18 and not greater than 13.	Child age: 10	Child account will not be created.	As Expected	Pass

3.	Verify the child account is not created if the age of child is not less than 18 and is greater than 13.	Child age: 20	Child account will not be created.	As Expected	Pass

6.4. Integration Testing

Integration Testing 1: Parent and Child Login Integration

Testing Objective: To ensure the adding child is being done correctly and *the **interface** between* Child and Parent are run correctly.

Table 6.55: Test Cases for Parent and Child Login Integration

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Register Child	Parent will add a child by entering credentials.	Parent can view added child details and Child can login with those credentials.	As Expected	Pass

Integration Testing 2: Parent and Child Location Tracking Integration

Testing Objective: To ensure the location tracking is being done correctly and *the **interface** between* Child and Parent are run correctly.

Table 6.56: Test Cases for Parent and Child Location Integration

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Track Location	Child will grant Location Permission.	Child Location data has been stored and Parent can view the child’s live location and location history.	As Expected	Pass

Integration Testing 3: Parent and Child Content Monitoring Integration

Testing Objective: To ensure the content monitoring is being done correctly and ***the interface** between* Child and Parent are run correctly.

Table 6.57: Test Cases for Parent and Child Content Integration

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Monitor Content	Child will grant Content Monitoring Permissions	Child Content Monitored Data has been stored and Parent can view the child's content.	As Expected	Pass

Integration Testing 4: Parent and Child Notifications Integration

Test Objective: To ensure the notification is being done correctly and *the **interface** between* Child and Parent are run correctly.

Table 6.58: Test Cases for Parent and Child Notification Integration

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Send Notification	Child will send SOS notification.	Parent can view the child's notification.	As Expected	Pass

7 Conclusion and Future Work

This section of our report include describing the project conclusion and highlights future work for our project.

7.1 Conclusion

The main aim of this project was to develop a Child Monitoring Mobile application for android phones. The monitoring software named as "CMS" was fully developed using Flutter Technology. By analyzing the busy routine of Parent and focusing on child safety issue. We concluded that the dire need of a system allowing parents to stay connected with their kid’s use of technology. Additionally, it should provide full access of child mobile phone to parents. We also highlighted the need of monitoring application for parents to prevent cyberbullying, harassment and smartphone addiction to their child. Moreover, we also reviewed different applications already available in the market and critically analyzed their pros and cons. Findings revealed that most of the apps do not provide geo fencing features. We accomplished the main goal of connecting the parents with their kid’s use of technology. Moreover, feature like monitoring of SMS, call, contact and installed applications allowed parents to have firm control on their child’s communication. Our developed system outsmarts other applications of the same category in many aspects. First and foremost is the availability of our application without any cost. Additionally, we provided the feature of geo fencing to help parent to stay aware of their child’s location. We utilized the latest technology for development of our database i.e. Firebase, which enabled us to synchronize the data on both ends within seconds. Another strong feature provided by our application is “SOS Caution”. With the help of this feature, child felt safe and secure as he/she is just now a tap away from their parents in case of any emergency. By this project we will learn android development, usage of API and database management. We will try to complete project on time and make it a complete product.

7.2 Future Work

In future, we will extend this project by providing web portal and IOS app through which parent will have the facility to monitor their child devices from wide different variety of devices.

8. References

[1]. Jabeen, T. (2020). Child Protection Data: an analysis of Newspapers Coverage of child protection issues in Pakistan. *South Asian Studies*, *29*(1).

[2]. M. E. Hossain, M. Rahman, K. M. Qaiduzzaman, A. K. Shakir and M. M. Hassan, "Efficient Anti-Kidnapping and Anti-Harassment (Avoidance-Detection-Notification) Mobile Application for Unwanted Incidents," *2019 IEEE Student Conference on Research and Development*, Bandar Seri Iskandar, Malaysia, 2019, pp. 112-116

[3]. G. P. Miriyala and P. V. V. N. D. P. Sunil, "Intelligent Safety System for Women Security", *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, vol. 7, no. 2, pp. 41-46, 2016.

[4]. Varma, K., Baikar, P., & Mangale, S. (2015). Human Surveillance with Android based SMART Emergency Application. *making of*, *3*(4).

[5]. Ali Behforooz& Frederick J.Hudson, (1996), Software Engineering Fundamentals, Oxford University Press. Chapter 8, pp255-235.