CSE 535: Project 4 Report

Sarthak Swetang Shah (Group 37)

- **Alignment with Guardian Angel:**

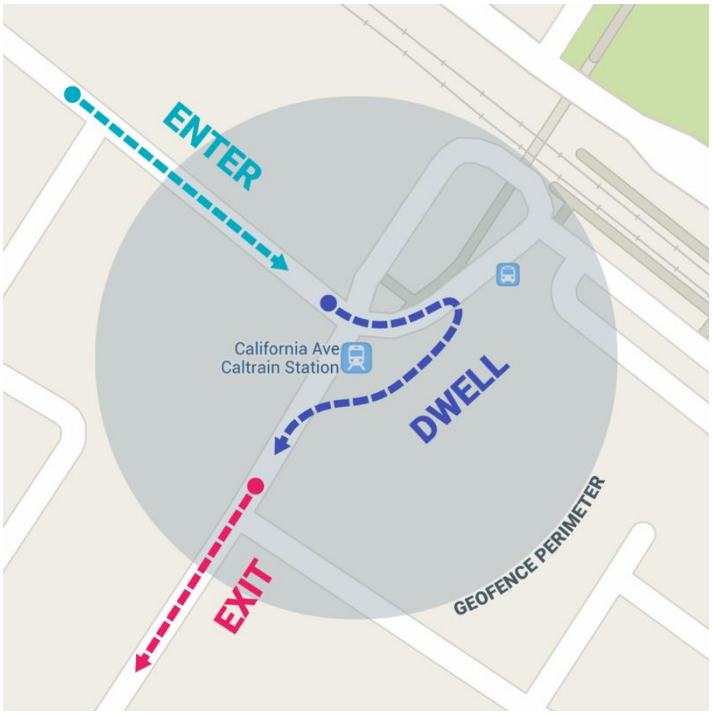
GuardianTeen is an Android application aimed at providing parents with an effective and versatile tool for ensuring the safety and well-being of their teenage children. This application introduces several key features to help parents stay informed and take action when necessary, including Geofencing which is done by me.

The geofencing feature in GuardianTeen is integral for ensuring the safety of teenage children. By setting up virtual boundaries (geofences), parents can receive instant notifications if their child leaves or enters designated safe areas. This feature aligns well with the core objective of the app, which is to provide parents with tools to monitor and ensure the well-being of their children.

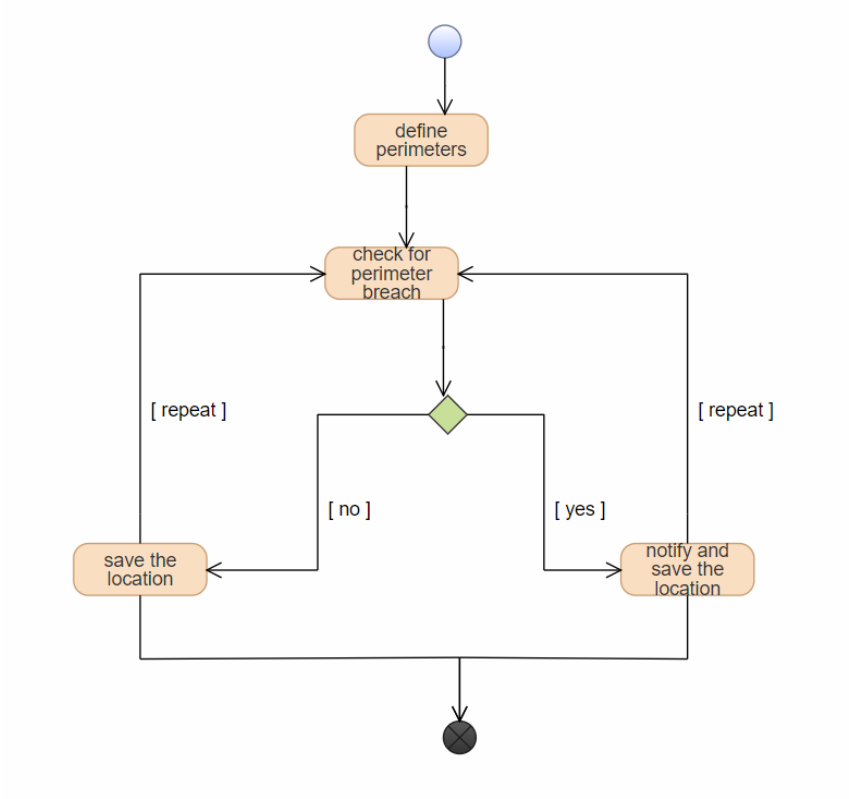
- **Specifications:**

1. Control Flow –

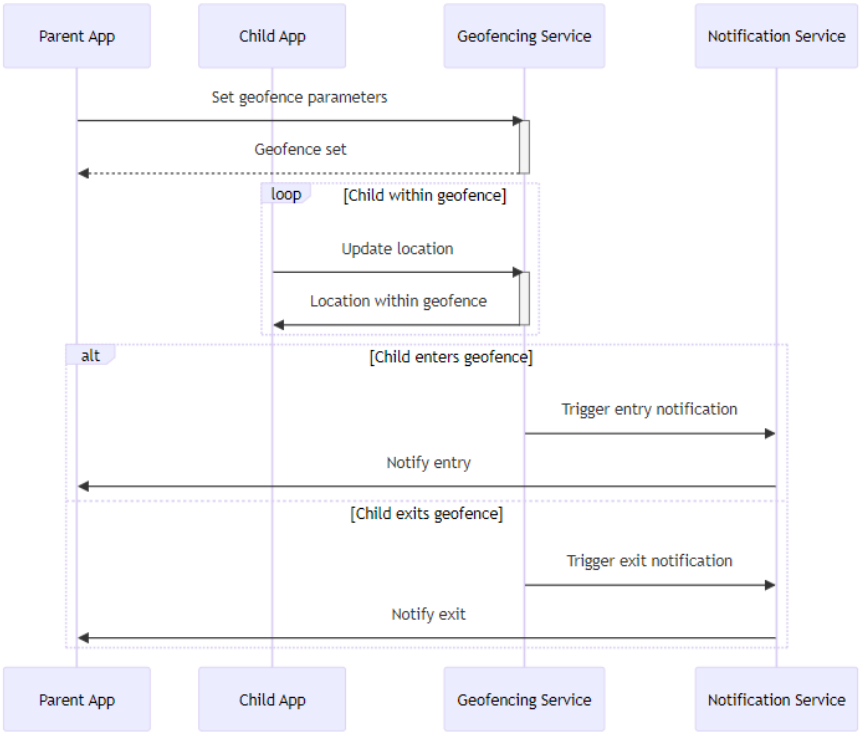
The geofencing feature involves first setting up a geofence by long pressing on the map (a circle with the long press location as the center and a specific radius becomes the geofence), monitoring the child's location, and then triggering notifications when the child enters, roams inside, or exits the geofence.



2. UML –


define perimeters
check for perimeter breach
[repeat]
[repeat]
[no]
[yes]
save the location
notify and save the location

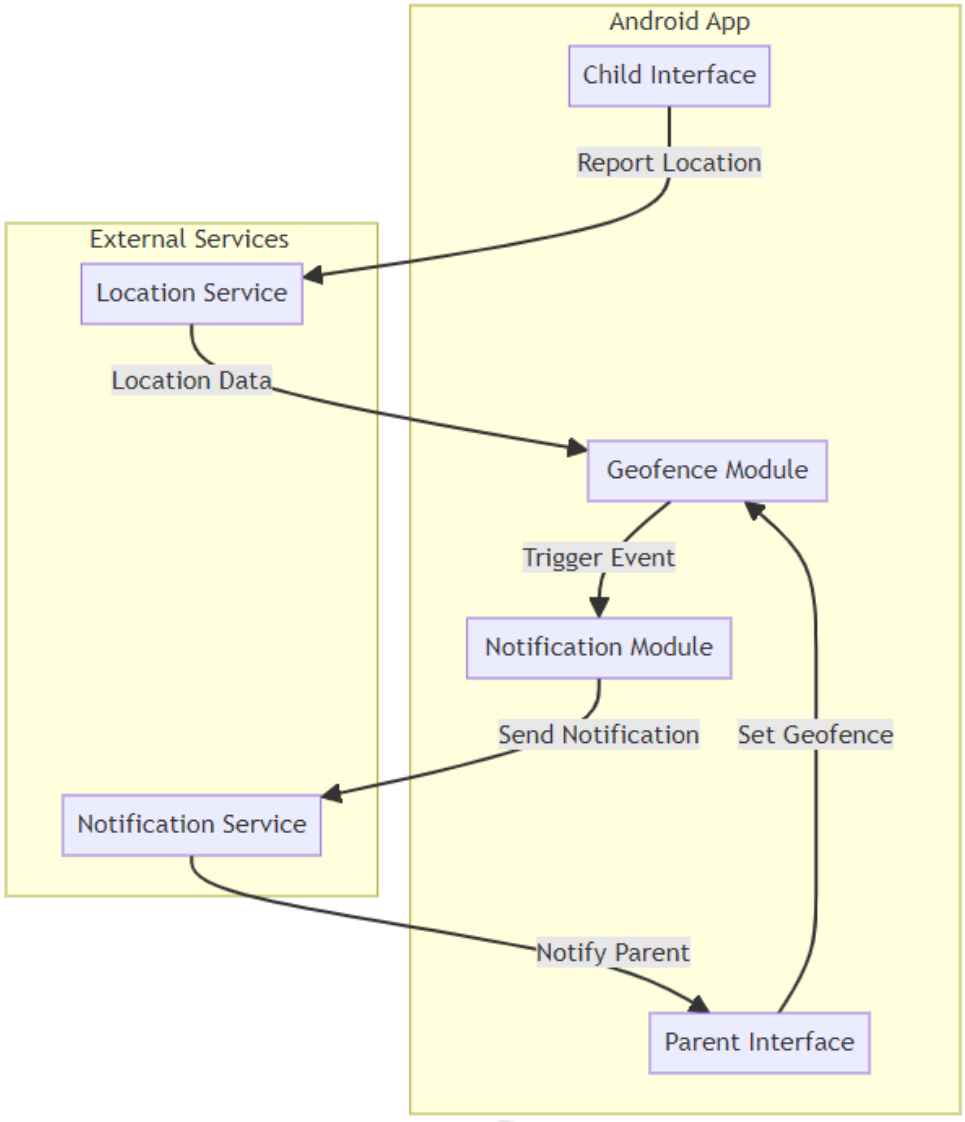
3. AADL –


Parent App
Child App
Geofencing Service
Notification Service
Set geofence parameters
Geofence set
loop
[Child within geofence]
Update location
Location within geofence
alt
[Child enters geofence]
Trigger entry notification
Notify entry
[Child exits geofence]
Trigger exit notification
Notify exit
Parent App
Child App
Geofencing Service
Notification Service

- The Parent App sets geofence parameters with the Geofencing Service.
- The Child App regularly updates its location to the Geofencing Service.
- Depending on whether the child enters or exits the geofence, the Geofencing Service triggers the appropriate notification through the Notification Service, which then notifies the parent.

- **Design:**

1. Component Architecture Diagram –



This diagram includes the interactions between components, showing how they work together to provide the functionality of setting geofences and sending notifications.

- Parent Interface: Where the parent sets the geofence.
- Child Interface: Used by the child, which reports location data.
- Geofence Module: Manages geofencing logic.
- Notification Module: Handles sending notifications.
- Location Service: External service providing location data.
- Notification Service: External service for sending notifications.

2. Overview of Design –

The geofencing feature is designed to be user-friendly, allowing parents to easily set up and manage geofences. It integrates with the device's location services and a backend server to monitor the child's location in real-time.

3. Tech-Stack –

- Android SDK for the app development (Android Studio)
- Google Maps API for mapping and geolocation services (Maps SDK for Android)
- a backend server (using Node.js)
- a database for storing geofence and user data (MongoDB)

- **Testing Strategies:**

Unit Testing: Test individual components of the geofencing feature, like location retrieval, geofence setup, and notification triggering.

Integration Testing: Test the integration with other parts of the application and external APIs.

End-to-End Testing: Simulate real-world scenarios where a child's device moves in and out of geofences.

User Testing: Have real users test the feature for usability and effectiveness.

- **Navigating Challenges:**

I encountered challenges like handling location inaccuracies, optimizing battery usage, ensuring the reliability of notifications, and maintaining user privacy. The approaches I took to overcome these challenges included using advanced location algorithms, efficient data transmission methods, robust error handling, and clear privacy policies. Some of the lessons I learned along the way were reflecting on the insights gained from tackling these challenges, emphasizing the importance of thorough testing, user feedback, and continuous improvement.