# Harshak Krishnaa Keerthipati

## Project 4

## Steps Monitor


## Abstract:

StepsMonitor is a project dedicated to implementing a user-friendly system for effective step tracking and goal-setting. The goal is to encourage users to achieve their daily step targets seamlessly. The project involves the development of an Android application, also named StepsMonitor, with a push notification feature integrated into the backend. These notifications serve as reminders for users to meet their goals, and when interacted with, the app navigates to a map interface showcasing nearby parks and gyms, promoting a holistic approach to fitness.


## Alignment with Guardian Angel:

**About Guardian Angel:** The Personalized Wellness Assistant is a holistic solution, prioritizing user well-being through diverse features. It tracks vital signs, notifies emergency contacts, offers weather-driven suggestions, monitors women's health, provides personalized dietary guidance, encourages daily activity with step goals, and optimizes sleep patterns.

**Alignment with Guardian Angel:** My project, StepsMonitor, a pivotal component within this initiative, seamlessly integrates into the Wellness Assistant. Focused on step tracking and goal-setting, it aligns with the project's broader goal of promoting well-being. The app's notifications and map interface for nearby fitness locations contribute to its holistic approach.

## Specifications:

**Control Flow:** The user sets his daily goals in the app screen. The user also has an option to get notified at a specific time of his liking, whether he has reached his daily goal or not. If the user hasn’t reached the daily goal, when he clicks the notification he gets navigated to a suggestions page, that shows parks and gyms nearby his location, where he can go and finish his goal.

**Components:**

1) **MainActivity**: This activity shows the user his daily step count and also his goal for the day. The user can also set remainders for the time he wants to get notified of his daily goals.

2) **AlarmActivity**: When the user sets the time in the MainActivity, it gets sent to this AlarmActivity that sets the notification for the time picked by the user.

3) **AlarmReceiver**: This activity runs in the background and receives the payload exactly at the time set by the user with the data of daily goals and daily step count. It sends notification whether the user has successfully completed his daily goal or not.

4) **SuggestionsActivity**: This activity displays a map with the users current location and also suggests nearby parks and gyms where the user can go and finish his daily goal.

## Design:

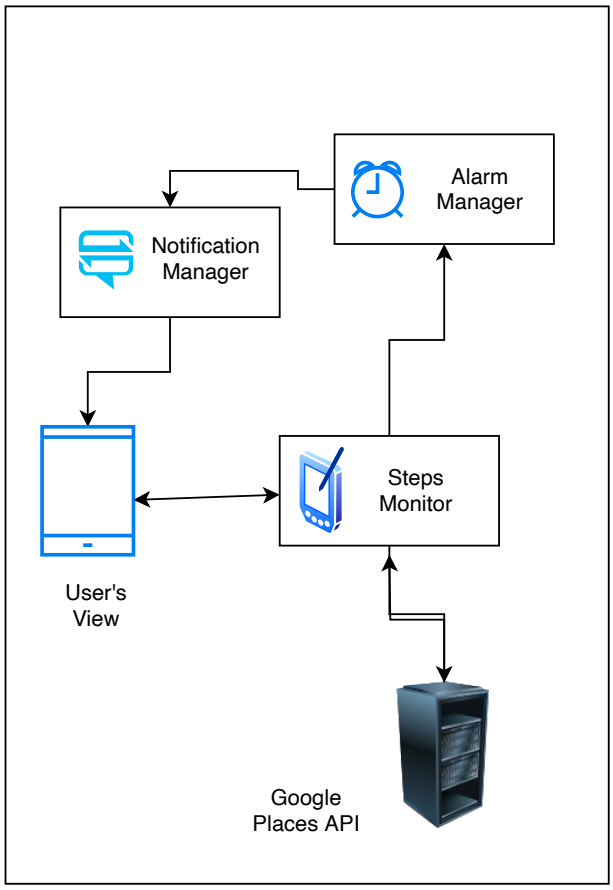
**The Components:**

**Notification Manager:** Responsible for sending notifications to the user.

**Alarm Manager:** Responsible for scheduling and triggering alarms.

**Steps Monitor:** Responsible for monitoring the user's steps and activity.

**Google Places API:** Responsible for fetching the possible walking spots near the user's location.




The design of the system is modular and decoupled, which makes it easy to maintain and extend. The three main components are responsible for specific tasks, and they communicate with each other through well-defined interfaces. This makes the system easy to understand and debug.

The use of the Google Places API allows the system to provide users with relevant information about nearby parks and gyms. This can help users to stay motivated and reach their fitness goals.

The Alarm Manager is a reliable solution for sending push notifications to users. It allows the system to send notifications even when the app is not running as it can be set and the background process will successfully send the notification once alarm is set.

**The tech stack used for this system could include:**

**Android SDK(Kotlin):** For developing Android apps.

**Google Places API:** For displaying nearby parks and gyms.

**Junit and Robolectric:** For testing the modules

## Testing Strategy:

**Unit testing:** Wrote unit tests for the different methods in each class and validated if they worked correctly for different scenarios.

The test cases includes:

1) **MainActivity:** Checks if all the elements are displayed properly, if all the elements are changed accordingly and also if the alarm notifications are set perfectly.

2) **AlarmActivity:** Checks if the time is sent in proper format and if the set alarms is actually set in the android environment.

**UI testing:** Conduct UI tests to verify proper notification display in various app environments and across different devices.

## Navigating Challenges:

**1) Intuitive UI Design:** Utilize established UI design principles such as consistency, hierarchy, and simplicity to overcome designing user interactive UI.

**2) Pop-up Preference Settings:** Designing interactive pop-ups for a better user experience was challenging, but I overcame it by effectively learning and utilizing a dialog module in the design process.

**3) Alarm Notification Design:** The alarm notification challenge was resolved by ensuring messages were clear and brief, implementing a user-friendly system for preferences customization, and ensuring smooth functionality across various devices.

**4) Integration of Maps API and Suggested Spots:** I tackled the Maps API challenge by studying the documentation, seeking help from developer communities, and testing to make sure suggested spots are right for users.