Department of Computer and Information Sciences

MSc Advance Computer Science with Data Science

A Context- Aware Behaviour Change Trigger App

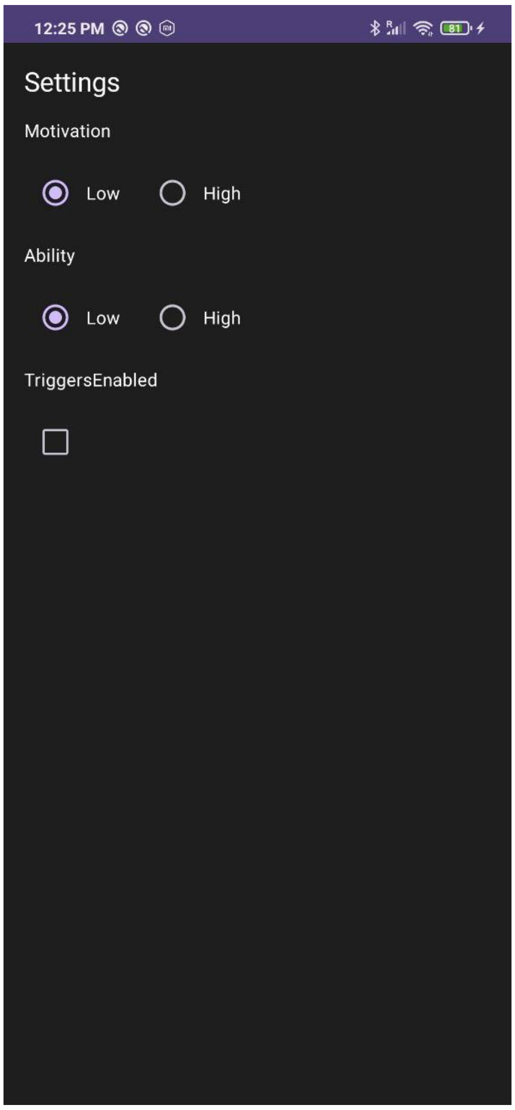
CS5551 Summer Resit Assignment

Manoj Kumar Dharmaraj

202468855

Introduction

This Android app is intended to improve user fitness by sending notifications on calories burnt, meal schedules, and tracking inactivity time with the app. It uses jetpack modules, which is view model to better Ui management data, and WorkManager for better background tasks. The application gets user input and time from DataStore to send notifications to keep the users on the right track. At the same time, user's motivation and ability are high or low according to different types of notifications that will be sent. This method will bring upon a habit for the users and it will continuously give upward progress in their weight loss journey.



12:25 PM

Settings

Motivation

[✓ Low] High

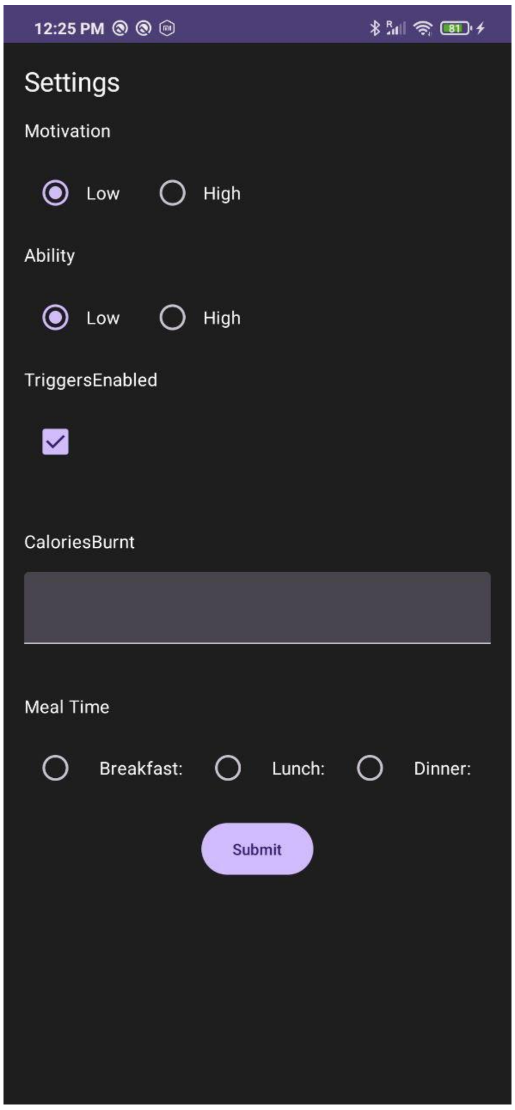
Ability

[✓ Low] High

TriggersEnabled

☐

Settingscreen without triggers enabled



12:25 PM

Settings

Motivation

[✓ Low] High

Ability

[✓ Low] High

TriggersEnabled

☑

CaloriesBurnt

Meal Time

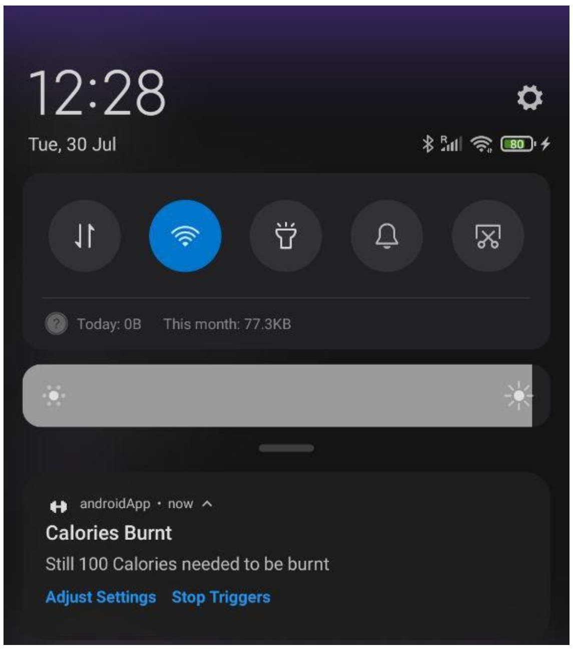
Breakfast: Lunch: Dinner:

Submit

Settingscreen with triggers enabled

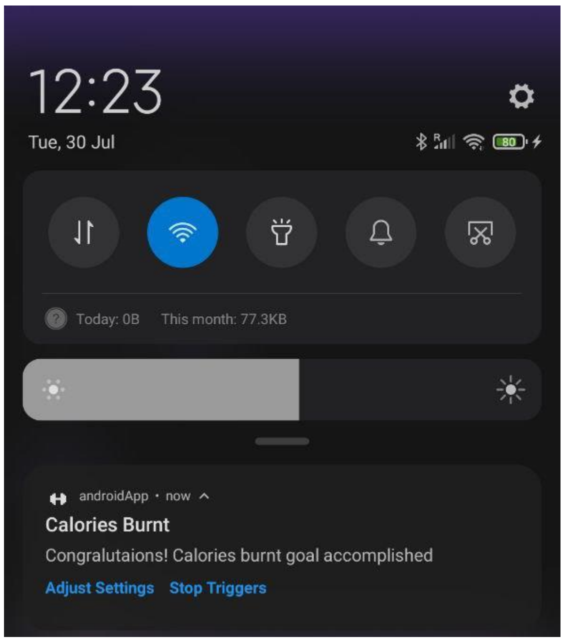
Calories burnt trigger

The NotifiWorker class is an essential component of an Android application that aims to increase user engagement through personalized notifications. This trigger is based on the user's input. When the user enters the calories burnt for the day based on the motivation and ability level the notification will be triggered and sent. Here the worker uses Android's Work Manager framework to run the backend tasks, confirming the notifications are sent based on the given conditions such as battery capacity, university wifi, and range. WorkManager library plays a vital role in scheduling background activities and by using DataStore API data are stored and retrieved easily and safely. The getUserPreferences function retrieves the data and allows the notifications to be tailored to the user's profile. Kotlin Coroutines allow for efficient control of background operations without blocking the main threads. Android Notification API manages and displays notifications to the user, enabling interaction and perseverance. Finally, the notification includes functionality for users to stop and adjust triggers.



Calories Burnt

Still 100 Calories needed to be burnt

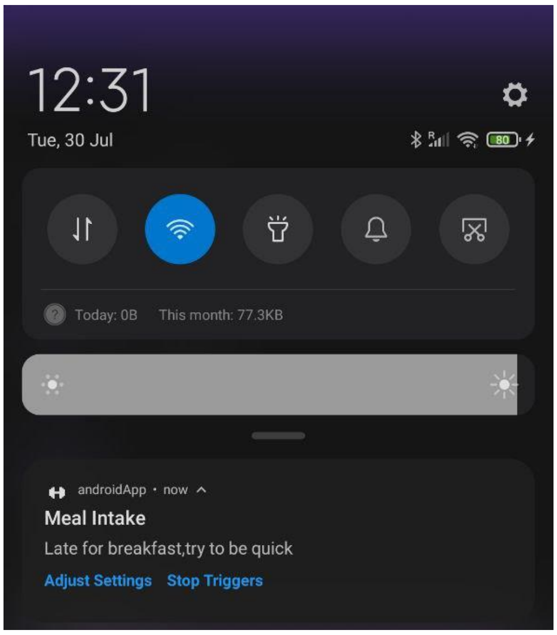


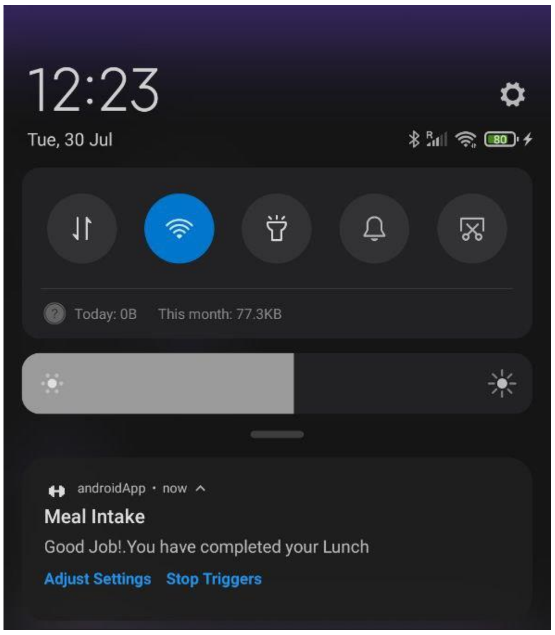
Calories Burnt

Congralutaions! Calories burnt goal accomplished

Meal Intake trigger

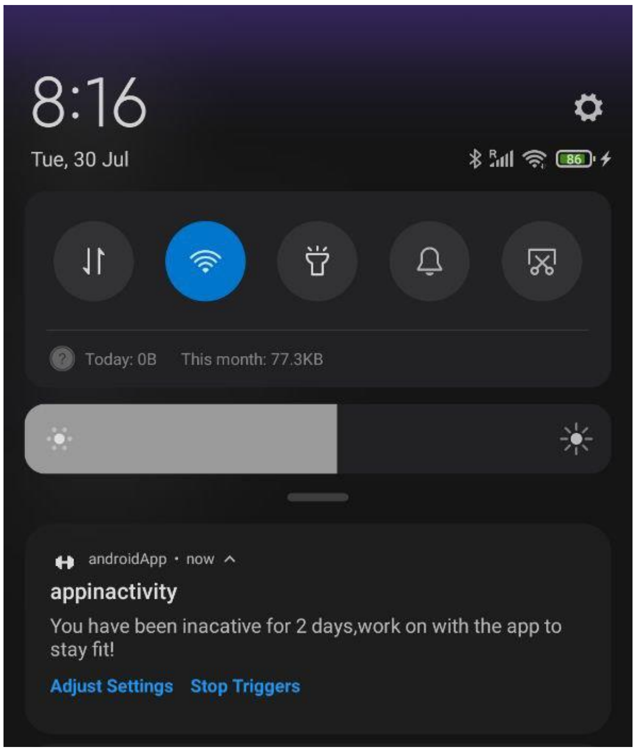
The meal intake is a time-based trigger that allows user to maintain their meal Schedules by getting the specified time, which is stored in DataStore and retrieves the data by using getuserpreference and gives notification. This feature ensures users about specific meal times breakfast, lunch, and dinner, and gives notifications like a signal or warning which depends on users' motivation and ability. One of the main functions and libraries used in meal intake was TimePickerDialog. It is easy to use for selecting time that includes 24-hour format with that user can select a specific time. The selected time is stored in datastore and it will trigger the notifications at user-defined times. This implementation also includes conditions such as battery capacity and network connectivity. The same WorkManager, Data Store API, notification channel with different channel IDs and other functions are used. The notification includes Stop and Adjust buttons for users to stop and adjust the triggers. This feature allows user to maintain their meal times and continues to their weight loss journey.


12:31
Tue, 30 Jul
Today: 0B This month: 77.3KB
androidApp • now
Meal Intake
Late for breakfast,try to be quick


12:23
Tue, 30 Jul
Today: 0B This month: 77.3KB
androidApp • now
Meal Intake
Good Job!.You have completed your Lunch

App activity Trigger

This feature is action-based trigger that gives notifications according to the user’s action. When the user is not active with the application for more than 2 days it will trigger the motivation type of notification to open the app to stay fit. Here LaunchedEffect composable function is used which is provided by Jetpack compose. It has many uses but in Appactivity trigger it is used to track the user's active time since the application was installed and give the notification once the user opens the application. Also, this trigger uses WorkManager to trigger the notifications and the same DataStore API and other functions. But the difference is it has separate settingviewmodel, mainactivity function, and setting screen function because of not to integrate with the other two triggers otherwise it will trigger the notifications for the other two triggers while the user opens the application. By sending notifications based on inactivity will eventually help the user to form habits and stay on track to weight loss.



appinactivity

You have been inacative for 2 days,work on with the app to stay fit!

Strength

The main advantage of this application is it is user based on their activity and preferences. By using Preference DataStore user can set goals and get desired notifications based on their activites or input. Also using Jetpack Compose it makes the code more structured and reliable. Additionally, using LaunchedEffect composable function it makes the application more responsiveness to user interactions.

Weakness

The main disadvantage of the app is every time while opening the application the user has to enable the location manually. The current implementation of code conditions like network connectivity to trigger notifications will lead to delayed notifications because of poor connections of the network meanwhile battery levels can suddenly drop in that case notification will get skipped. This code is not fully adapted to this kind of condition.

Privacy and Security

Some data like user's meal time, calorie count, and motivational levels are personal and sensitive information that could be easily breached by some unauthorized individuals. They can gain access to this sensitive data through weak structures of code or authentication mechanisms. Because user data are not encrypted. All three triggers handle sensitive data so they should have proper access control and permission should be established to guarantee that only authorized users can initiate these workers. Sensitive information going to be displayed on notifications, so this could be seen by anyone who has access to the device.

Mobile computing challenges

There are wide range of mobile devices, software is there so, features as time pickers and notifications behave differently across these devices. So, there must be proper extensive testing and adaptation should be provided. It is not well balanced when comes to notifications that are delivered timely may lead to user exhaustion and delayed notifications or insufficient updates lead to applications value.

References

https://developer.android.com/develop/ui/views/notifications/channels

https://developer.android.com/reference/android/app/TimePickerDialog

https://developer.android.com/topic/libraries/architecture/viewmodel/viewmodel-factories