Summary Report – final Project

Smart Reminder

Hala Grefat 207921008 Hala.gr11@gmail.com

Joumana Khateeb 206610826 joumana.1444@gmail.com



University of Haifa

Computer since

2021 -2022

Table of Contents

[Defining the Problem 3](#_Toc112450506)

[Project Definition 4](#_Toc112450507)

[Implementation 5](#_Toc112450508)

[Features 7](#_Toc112450509)

[How we would continue 9](#_Toc112450510)

[Knowledge 10](#_Toc112450511)

[Tallest hill to climb (most deficit problem) 11](#_Toc112450512)

# Defining the Problem

With the world evolving and “hustle” being the new trend, it’s more difficult for people to keep up with everything they need to do in the midst of a hectic day.

In addition, some tasks need to be done at a specific time or place or both!

Therefore, it would be convenient to have an alarm go off with a note stating a task **when** or/and **where** needed.



# Project Definition

Our project is an android application that sends notifications/reminders based on the users location, Date and/or time (hour).

For each user, when creating an account, they can add notes and reminders to their list of reminders, the app will send a notification with a note based on the reminders they set. even when the app is off, set reminders notification will pop up.

The application saves previously used locations to the users account map to make it easier to set reoccurring reminders.

In conclusion, the application is a smart TO-DO list, that helps its users do all their tasks and chores in an easy way to stay organized and not miss out on any chances in their hectic schedule.



# Implementation

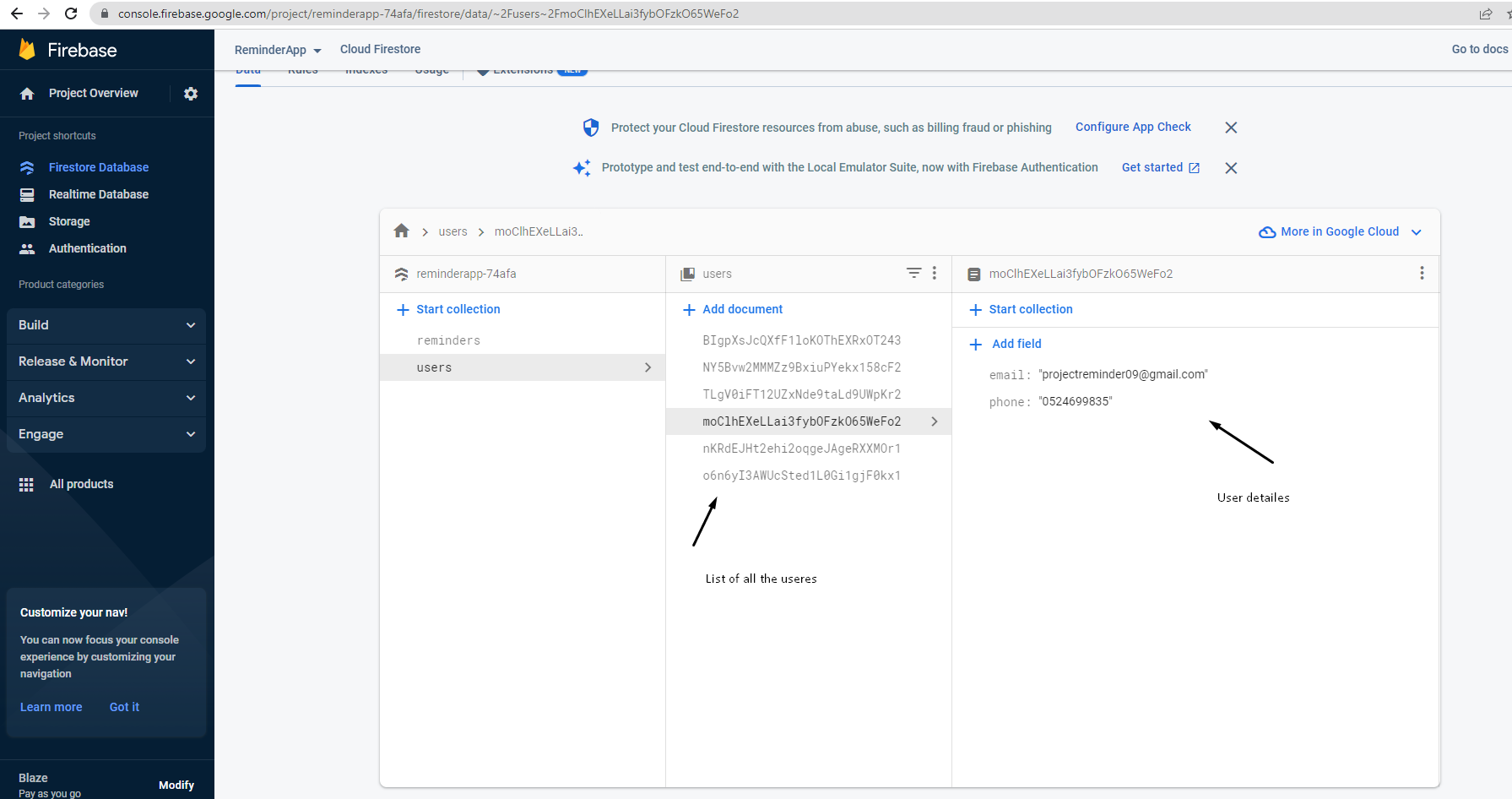
For our project we used **Android Studio** as our workspace to create an android application, the source code language is java, and XML for GUI.

We used **location Geofence** library for location-based functionalities, such as tracking the location of users phone, as well as save previously mentioned locations and other functions, we used **Google cloud services**, specifically **Google**-**maps** for the map UI, we created an API key for authentication because Google Cloud APIs only accept requests from registered applications.

For time related functionalities we used **AlarmManager** library.

For a serverless application we used **firebase database**, specifically **Firestore**, it’s a NoSQL cloud database that’s easy to connect to android studio, it allows us to store, sync, and query data for our mobile app - at global scale.

Cloud Firestore also integrates with **Firebase Authentication** to access the data securely with user-based authentication.



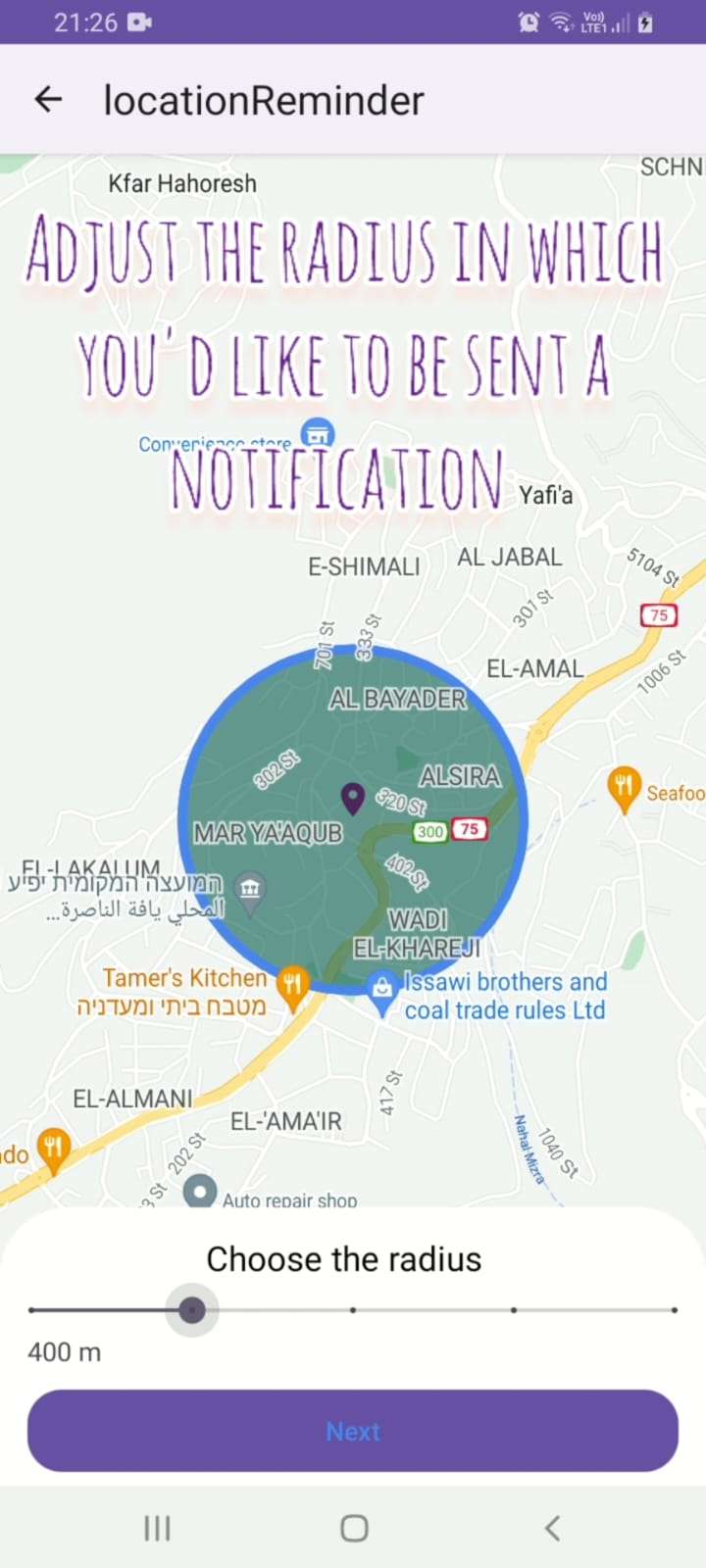
For testing the application, we needed to simulate real-time usage of the application, therefore we created a mock path, there are two ways to create a mock path, one is while running the application with android studio, the other is used when running the app on an android device, using an application called **Fake GPS Pro** that fakes the location of the phone to simulate a predesignated route.

To be able to work together as a team, we used **GIT open source** to save the code and help have multiple versions of it at the same time.



# Features

* Every user has their own account with all their personal data saved for them when they log in to the application.
* If password is forgotten, there’s an easy way to restore it/ create new one with the help of Gmail account.
* User can have more than three different types of reminders: based on date, time, location, or all at once.
* For location reminders, user can choose the desired location to have a notification sent to them there, as well as how close they are to that location, In the app they can set the radios around that location.



* User can delete or edit reminders/alarms after they saved them, reminder gets deleted/edited from DB as well as notification is canceled/ edited.
* Previously saved locations (used as destination location before), are saved in the map so that users can find them easily next time they want to set a reminder at that location again.
* The application works in background, which means that set reminders push notifications even if the application isn’t open, or even when it’s not in the recently opened applications (the app is completely closed).
* The notification the user receives contains the title and note set in the reminder that triggered the notification.
* From the map GUI, user can search locations by name, as well as search near by public places, such as gas station, restaurants, hospitals etc..., as well as set a reminder with the condition of being near a certain type of public place.



# How we would continue

For a smart reminder to be genus, we thought of some features to add:

* Adding the weather condition as a condition for a notification to be sent, example if it’ll rain at night remind me when at home to carry with me an umbrella.
* Adding the option to save photos in the details of the reminders or other types of files.
* Instead of having the application send a notification, have the application perform a previously set action/task such as, send a message to one of the users contacts, set the phone on silent or un-set it, the possibilities are endless …

Example, send mom a message when I get to school telling her I got there safely.

# Knowledge

**Previous knowledge we had before working on this project**:

Both of us had previous knowledge of java coding language, as well as the concept of server client applications.

**GUI development**, we were part of a team that created a desktop application with javaFX using eclipse IDE.

We already had the knowledge of how to use **GIT**, to create repository and branches to keep track of our progress and keep the already running code safe.

**Knowledge gained after working on this project**:

Working on this project we learned how to create an android application, as well as use **Android Studio IDE** for the first time.

We learned **NO-SQL** database and how it works, specifically firebase database (backend development).

We learned how to use GPS to get current location as well as fake locations to create mock paths for testing porpoises.

# Tallest hill to climb (most deficit problem)

Working on this project was very interesting and full of new technology that we don’t have experience with, hence it was very difficult to start working on the project.

Even though this isn’t our first project, but it was our first experience with android applications, and android studio IDE.

So, to be able to start we hade to gain some basic knowledge about Android studio, before even stating to plane out the structure of the application.

Another problem we face was location-based notifications, the most difficult task is identifying what’s wrong with your code when it doesn’t give any error but doesn’t work as expected.