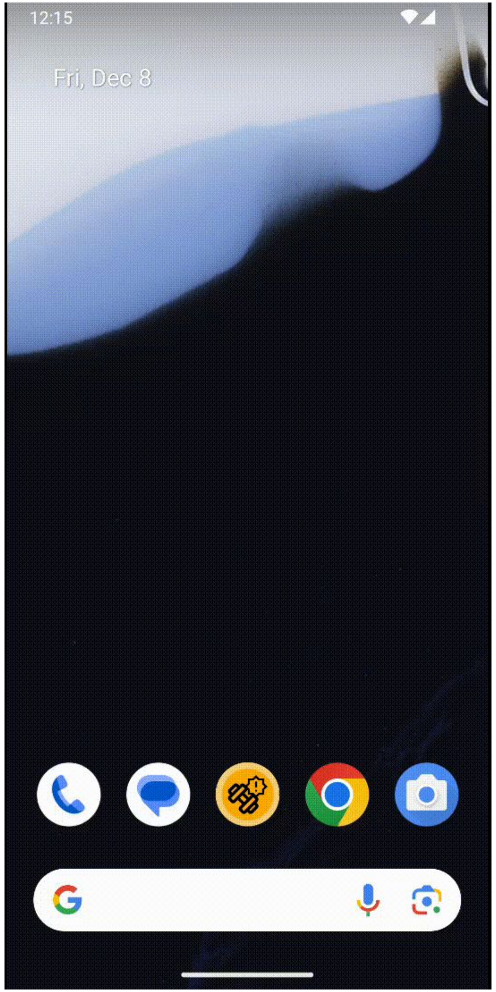
FitQuest

by Chet Gurevitch, Christina Garcia, Alexa McGinnis, Elian Renteria, Nelly Sanchez

App Description & Demo

Features:

- Step Counter
- Run Log
- Weight Training Log
- Exercise Alarm + Notification
- Water Intake Log
- Calorie Counter



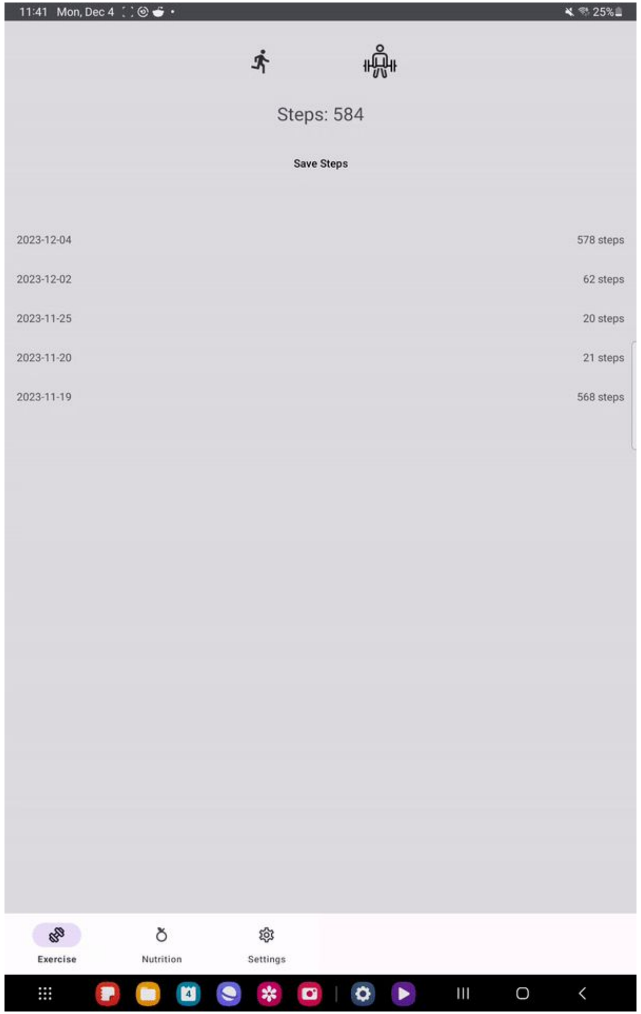
Exercises – Step Counter

Android Concepts:

- Permissions
  - android.Manifest.permission.ACTIVITY_RECOGNITION.
    - Access step counter sensor
- Recycler View
  - Displays and manages a dynamic user step history list efficiently
- Firebase
  - Stores and updates user step data in real-time with Firebase Firestore.
- ViewModel
  - Manage data operations and updates UI with current information.
- Fragment
  - Modular, reusable component

Challenges:

- Steps resetting on a new day
- Foreground Service
  - Not implemented in final version due to time constraints



Steps: 584

2023-12-04	578 steps
2023-12-02	62 steps
2023-11-25	20 steps
2023-11-20	21 steps
2023-11-19	568 steps

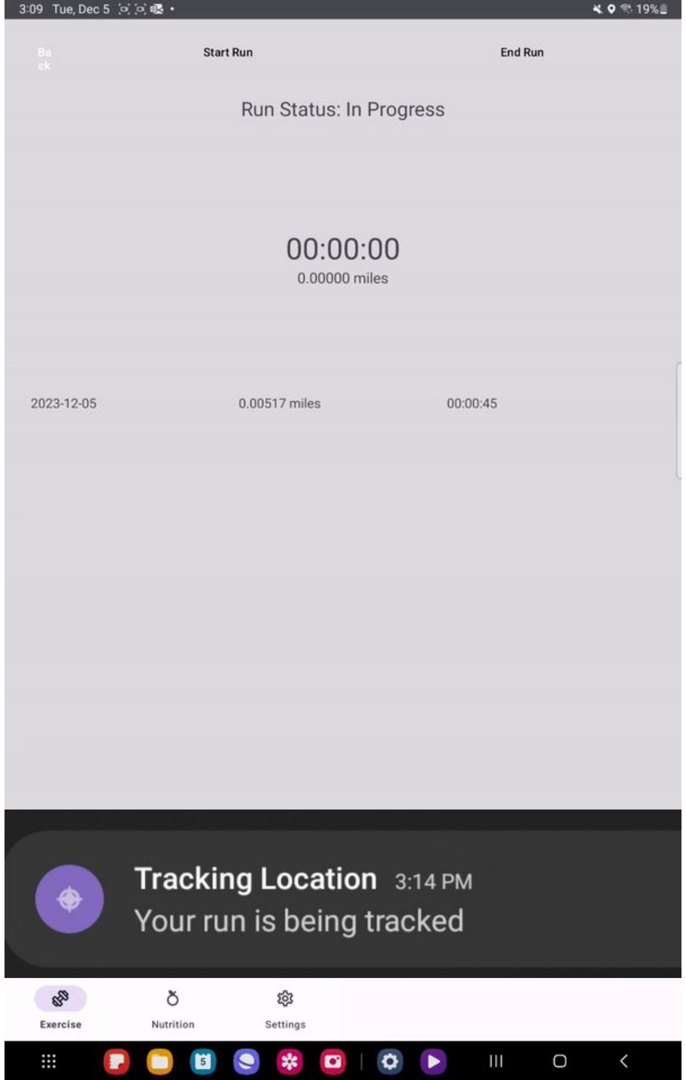
Exercises – Run Tracker

Android Concepts:

- Permissions
  - Coarse, fine, background locations
  - foreground service
  - foreground service location
- Recycler View
  - Displays and manages a dynamic user run history
- Firebase
  - Stores and updates user run data with Firebase Firestore.
- Fragment
  - Modular, reusable component
- Notification
  - Foreground Service Notification

Challenges:

- Track while app is in background
  - Foreground Service
    - Broadcasting data accurately



Run Status: In Progress

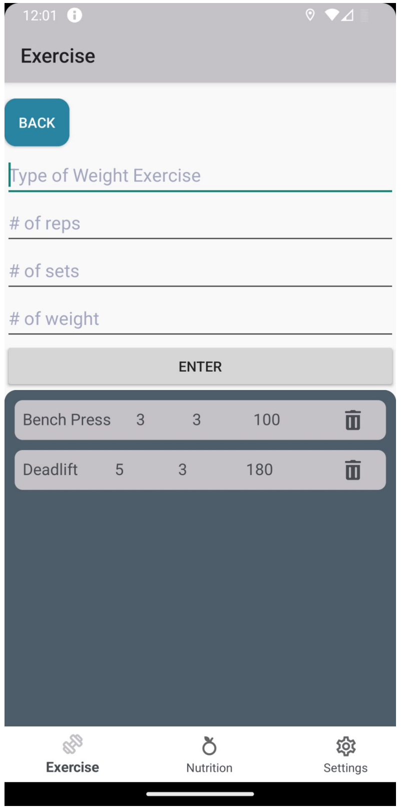
00:00:00
0.00000 miles

2023-12-05 0.00517 miles 00:00:45

Tracking Location 3:14 PM
Your run is being tracked

Exercises - Weight Training Log

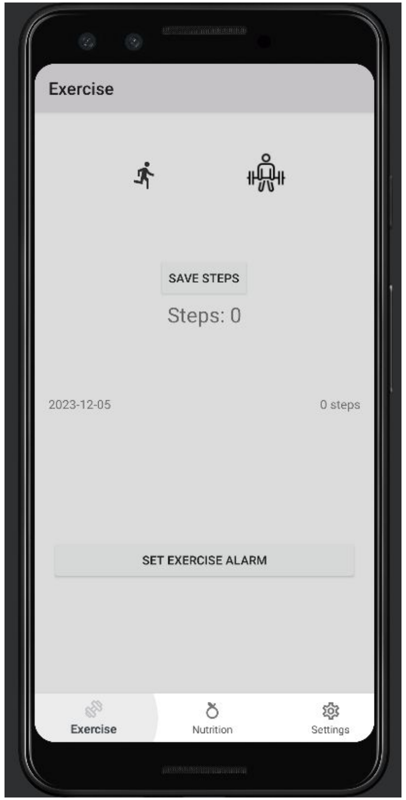
- Recycler View
  - Displays the weight training exercises in a list after the enter button is clicked.
  - The list will constantly update.
- Fragment
  - modular

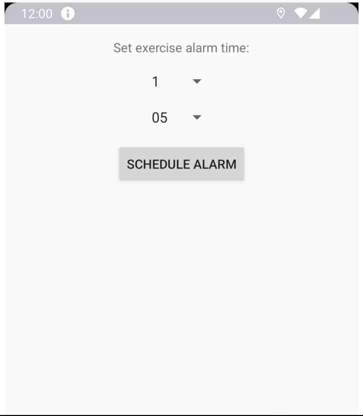


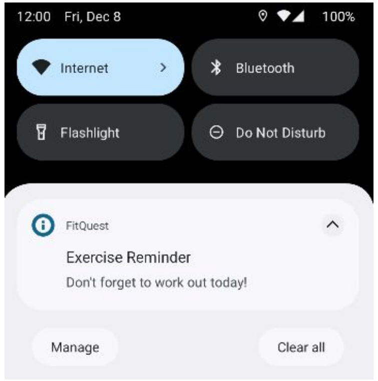
Bench Press	3	3	100
Deadlift	5	3	180

Exercise Reminder Alarm + Notification

- Activity
  - Handles user interaction with alarm setter
- BroadcastReceiver
  - Receive, create, & show the alarm notification
- UI
  - Button, TextView, Spinners,
- Challenges
  - Connecting the new activity, alarm, & ui components with the existing code







FitQuest

Exercise Reminder

Don't forget to work out today!

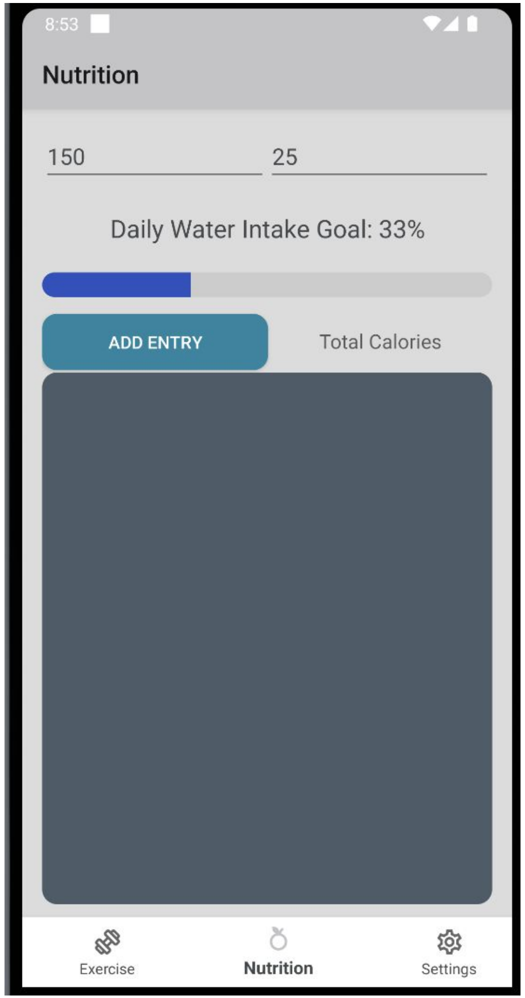
Water Intake Progress Bar

Android Concepts:

- RecyclerView
  - Food and calories get displayed in a list
- UI Elements
  - Buttons, edit texts, textviews

Challenges:

- Formatting
  - Water intake percentage would go past 100 percent and getting the water bar to animate required a custom component



Daily Water Intake Goal: 33%

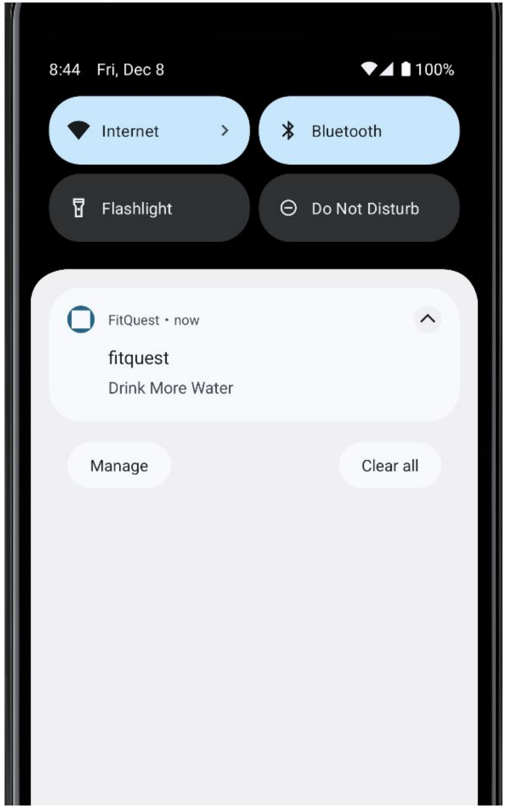
Scheduled Notifications

Android Concepts:

- Notifications
  - Created a alarm component that schedules notifications throughout the day at certain times to remind users to drink water

Challenges:

- Creating the Alarm component
  - Kept track of time and when to send notifications
  - Permissions
  - Displaying notifications



FitQuest • now
fitquest
Drink More Water

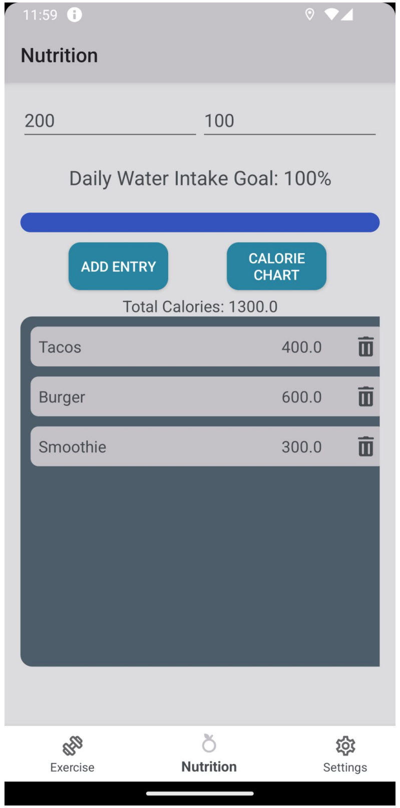
Calorie Counter

Android Concepts:

- Activities/Fragments
  - Modular
- RecyclerView
  - Food and calories get displayed in a list
- UI Elements
  - Buttons, edit texts, textviews
- Unit Testing
  - Calorie calculations

Challenges:

- Formatting
  - When adding the calorie chart button, the other UI elements on this fragment would move around



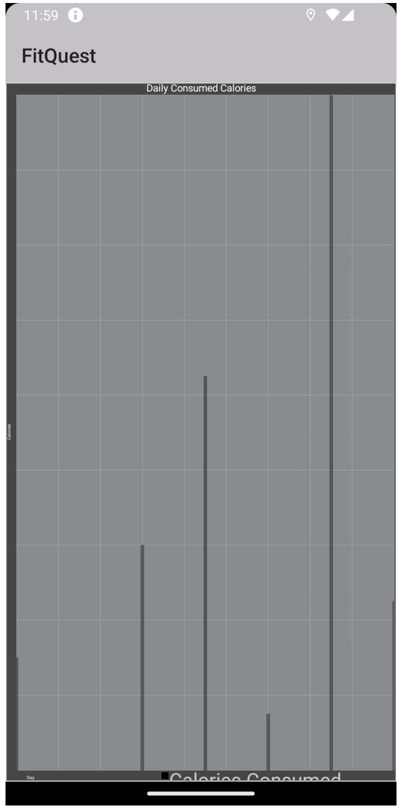
Calorie Counter Chart

Android Concepts:

- Chart/Graph
  - Will chart the daily consumed calories by user
- Sample Data

Challenges:

- Display
  - Displaying the graph correctly
- Firebase
  - Due to time constraints, we weren't able to connect our graph to firebase



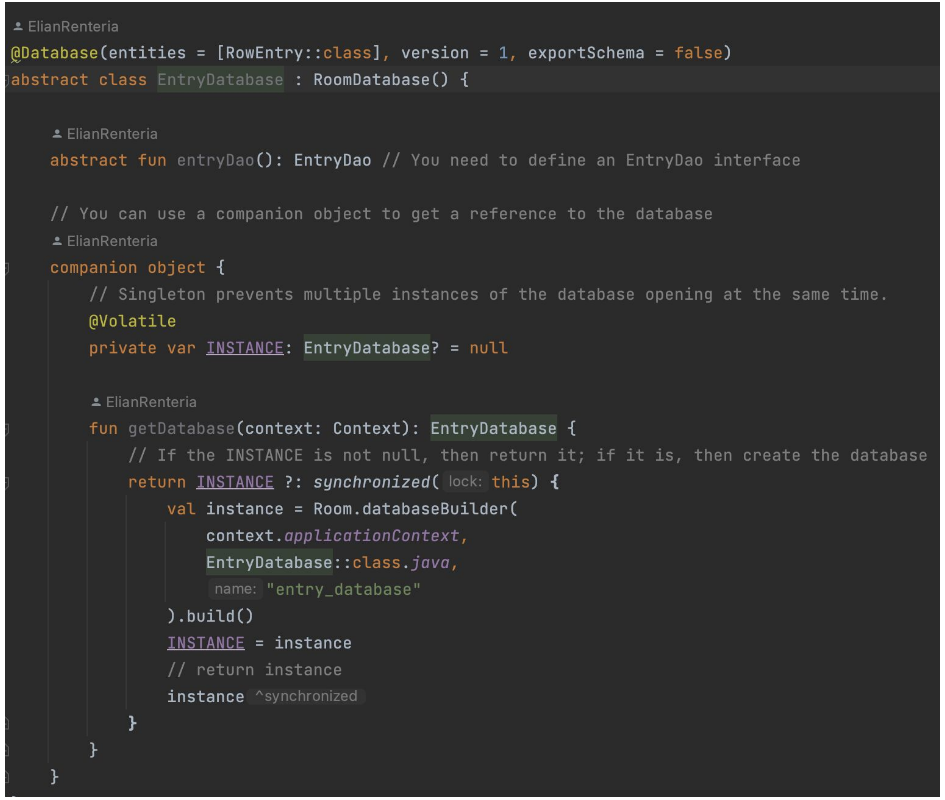
Room Database

Android Concepts:

- Room
  - Used to save calorie counter data across each tab

Challenges:

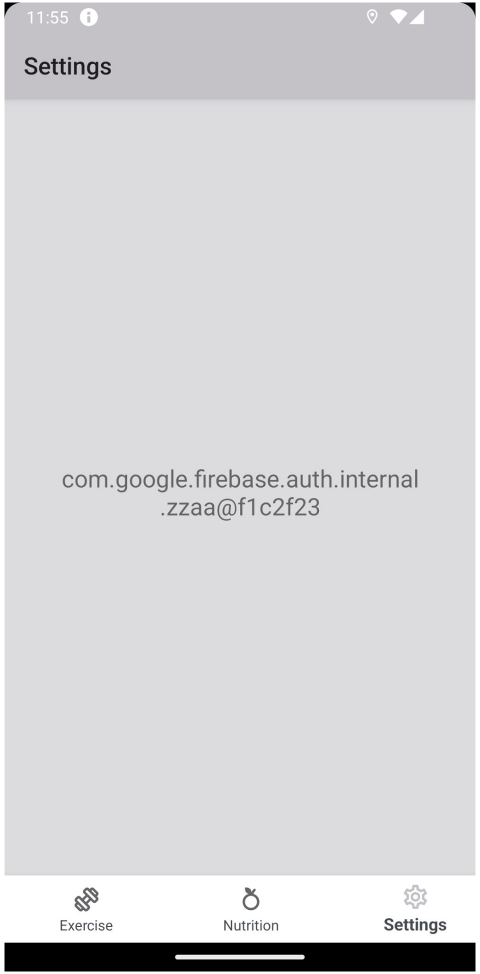
- Setup
  - Setting up the database and linking each row of the calorie list recycle view
  - Synchronizing each instance



```
ElianRenteria
@Database(entities = [RowEntry::class], version = 1, exportSchema = false)
abstract class EntryDatabase : RoomDatabase() {

    ElianRenteria
    abstract fun entryDao(): EntryDao // You need to define an EntryDao interface

    // You can use a companion object to get a reference to the database
    ElianRenteria
    companion object {
        // Singleton prevents multiple instances of the database opening at the same time.
        @Volatile
        private var INSTANCE: EntryDatabase? = null

        ElianRenteria
        fun getDatabase(context: Context): EntryDatabase {
            // If the INSTANCE is not null, then return it; if it is, then create the database
            return INSTANCE ?: synchronized( lock: this) {
                val instance = Room.databaseBuilder(
                    context.applicationContext,
                    EntryDatabase::class.java,
                    name: "entry_database"
                ).build()
                INSTANCE = instance
                // return instance
                instance ^synchronized
            }
        }
    }
}
```

Firebase

- Automatic anonymous login
- Exercise data backed up to firebase
- When more authentication providers added user data will be linked and preserved



com.google.firebase.auth.internal
.zzaa@f1c2f23