

**Project Report**

**of**

**iChappal**

**for**

**CMPE 297**

**By**

Gyanesh Pandey (gyanesh.pandey@sjsu.edu)

Manoj Kumar (manoj.kumar@sjsu.edu)

Shalini Narang (shalini.narang@sjsu.edu)

Yamini Muralidharen(yamini.muralidharen@sjsu.edu)

**Project Advisor**

**Prof.Chandrasekar Vuppulapati**

**Decoember 2018**

[**iChappal** 1](#_gjdgxs)

[**Architecture components** 3](#_1fob9te)

[Db Structure 4](#_3znysh7)

[REST API (non-functional as of now) - 4](#_2et92p0)

[Kafka server setup 4](#_tyjcwt)

[Kafka remote enable 4](#_3dy6vkm)

[Kafka Commands 5](#_1t3h5sf)

[Reference: 19](#_2s8eyo1)

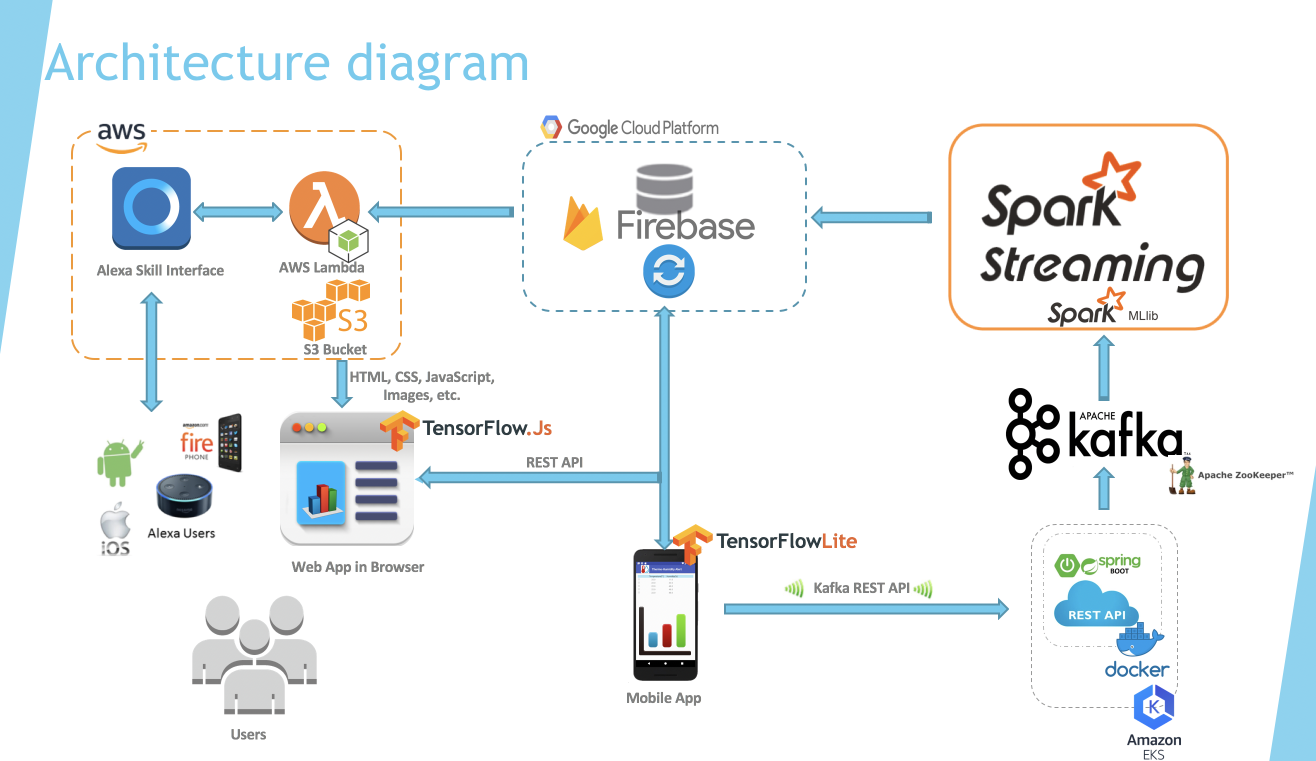
# 

**Introduction**

A pedometer app which used the accelerometer sensor to count the number of steps taken by the user. The step count stream values are fed as a stream to Kafka pipeline through JSON payload. The values are processed by Spark ML which has a trained model with user profile features like age, weight, height, gender etc. to predict the calories burned by the user. As the user walks the calories are predicted and a real-time feed of user progress Mobile App using Firebase

## **Architecture components**

1. Android App
2. Firebase real time db
3. Kafka
4. Spark Stream
5. Dashboard
6. Alexa
7. Web interface and TensorJS Review sentiment
8. TensorFlow Lite



## Db Structure

* kf-step-detail - userid, dateon, steps
* kf-total-steps - userid, dateon, totalminutes, totalcalories, levels, miles
* kf-user-profile - userid, gender, age, weight, height

## REST API (non-functional as of now) -

http://kafkapublisher-env.8tsy628pba.us-east-1.elasticbeanstalk.com/api/publish

## Kafka server setup

Create Ubuntu 16.4 EC2 M-Large Tier in AWS.

sudo apt-get update -y

sudo apt-get upgrade -y

sudo add-apt-repository -y ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer -y

sudo java -version

sudo apt-get install zookeeperd

netstat -ant | grep :2181

wget https://archive.apache.org/dist/kafka/0.10.0.1/kafka\_2.10-0.10.0.1.tgz

sudo mkdir /opt/Kafka

sudo tar -xvf kafka\_2.10-0.10.0.1.tgz -C /opt/Kafka/

sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-server-start.sh /opt/Kafka/kafka\_2.10-0.10.0.1/config/server.properties

sudo nohup /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-server-start.sh /opt/Kafka/kafka\_2.10-0.10.0.1/config/server.properties /tmp/kafka.log 2>&1 &

# Kafka remote enable

**configuring server.properties**  
Open server.properties file in edit mode

sudo nano /opt/Kafka/kafka\_2.10-0.10.0.1/config/server.properties

Uncomment and add public ip

advertised.listeners=PLAINTEXT://ec2-54-191-84-121.us-west-2.compute.amazonaws.com:9092

## Kafka Commands

Start

#cd /opt/Kafka/kafka\_2.10-0.10.0.1/

sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-server-start.sh /opt/Kafka/kafka\_2.10-0.10.0.1/config/server.properties

Stop

sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-server-stop.sh

Create Topic

sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic testing

Delete the Topic

sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-topics.sh --zookeeper localhost:2181 --delete --topic testing

Zookeeper to List Topic

sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-topics.sh --list --zookeeper localhost:2181

Producer

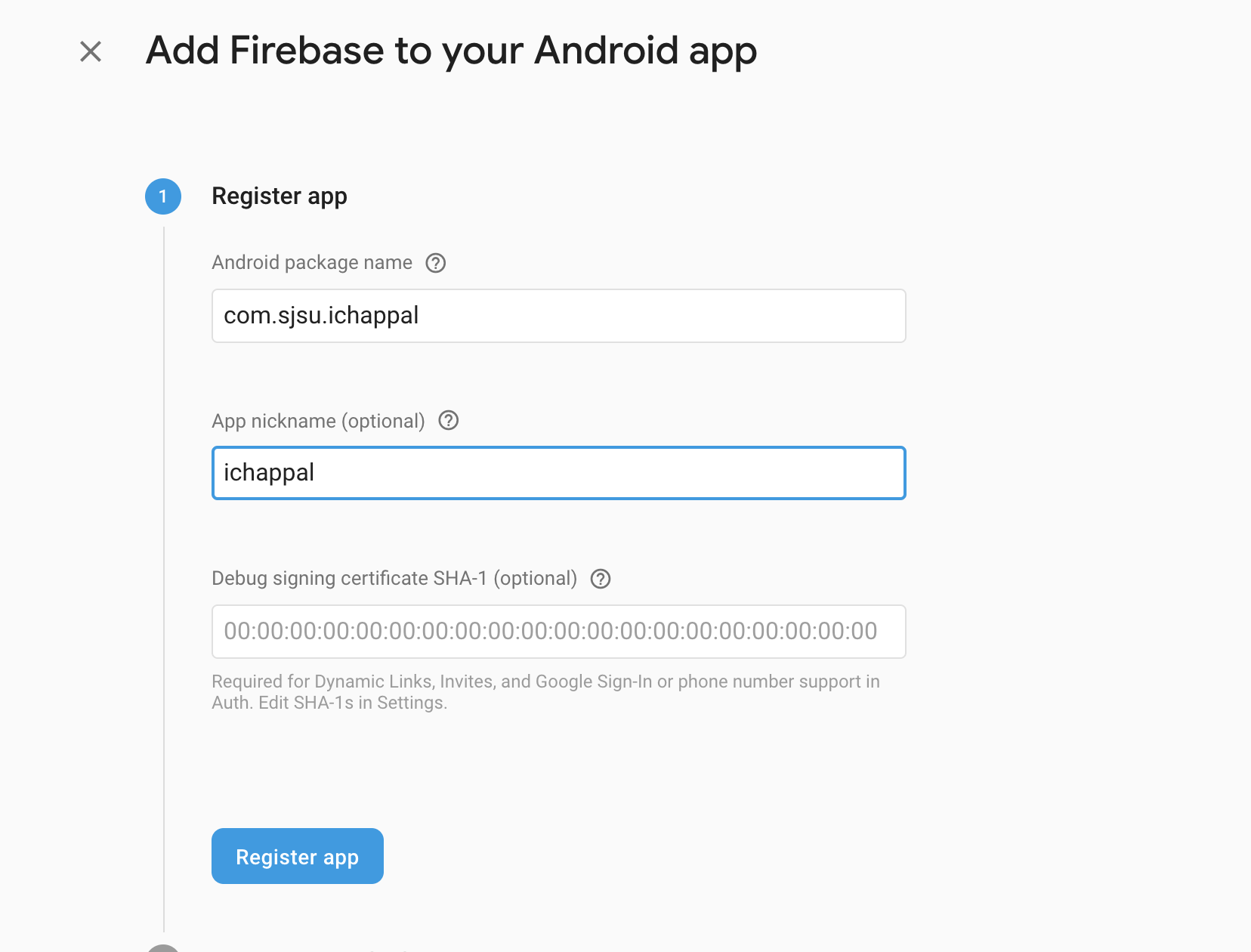
sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic testing

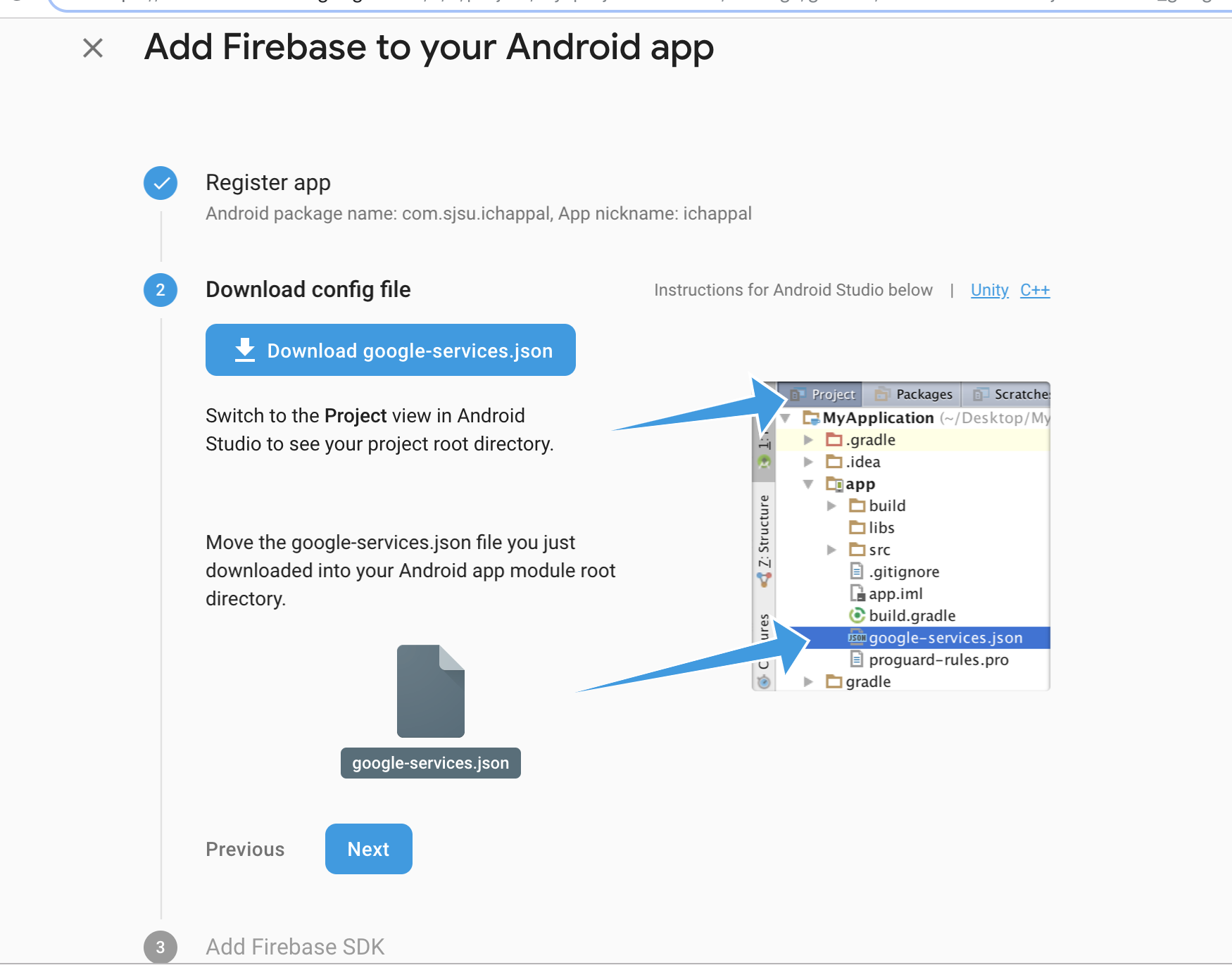
Consumer

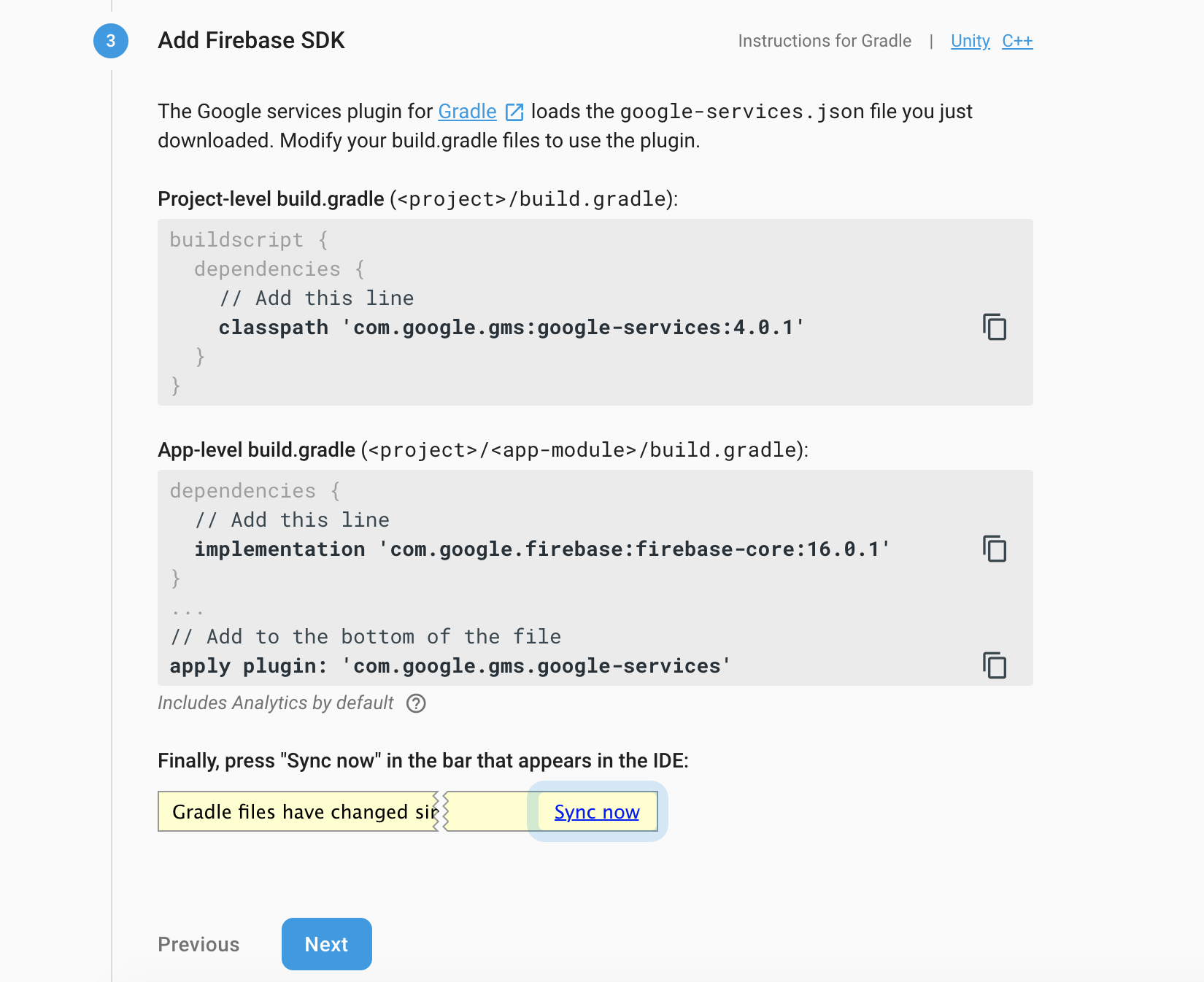
sudo /opt/Kafka/kafka\_2.10-0.10.0.1/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic testing --from-beginning

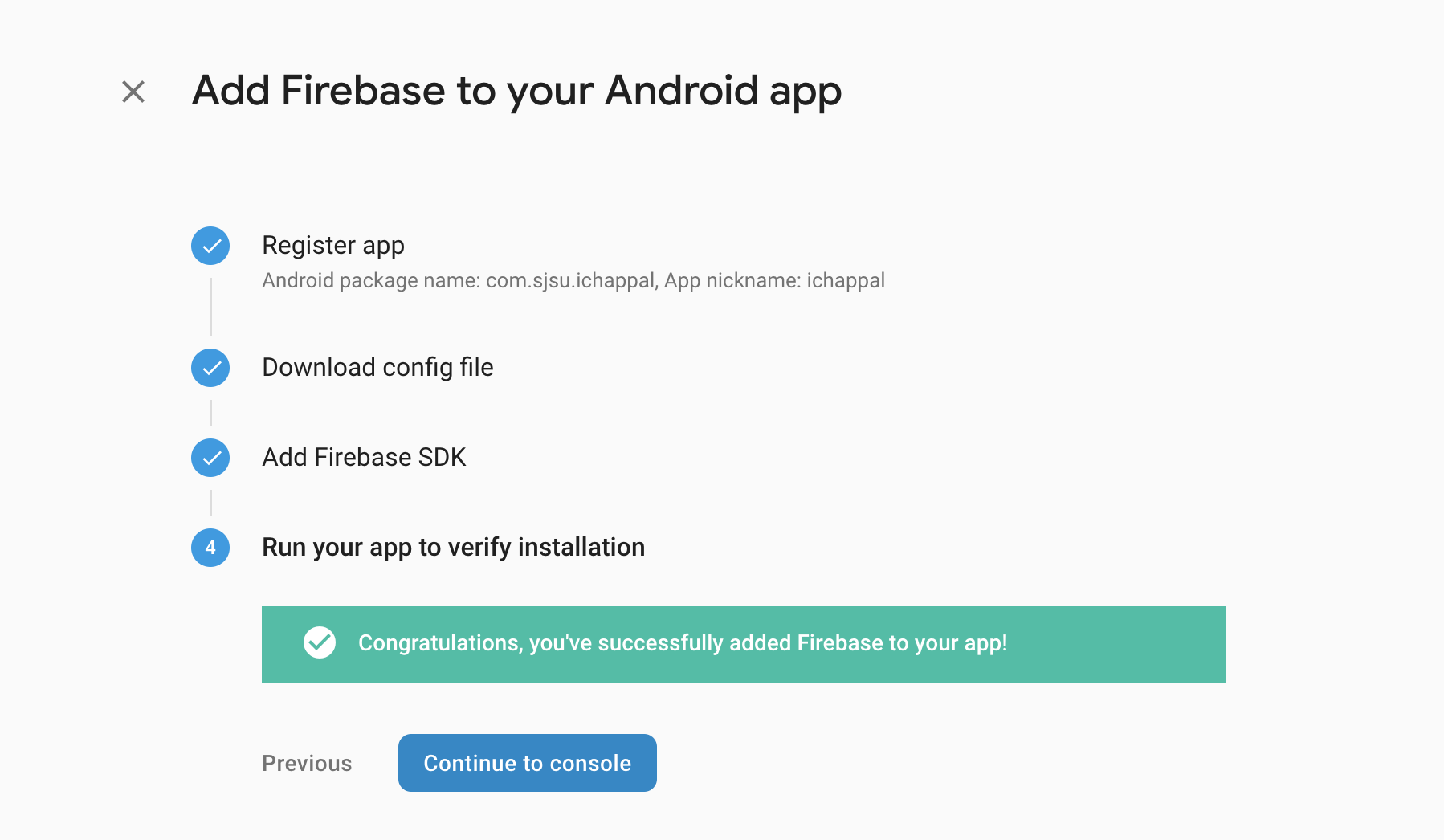
**Firebase Android integration**

Firebase has been used for authentication and fetches Real-time updates from the Firestore database

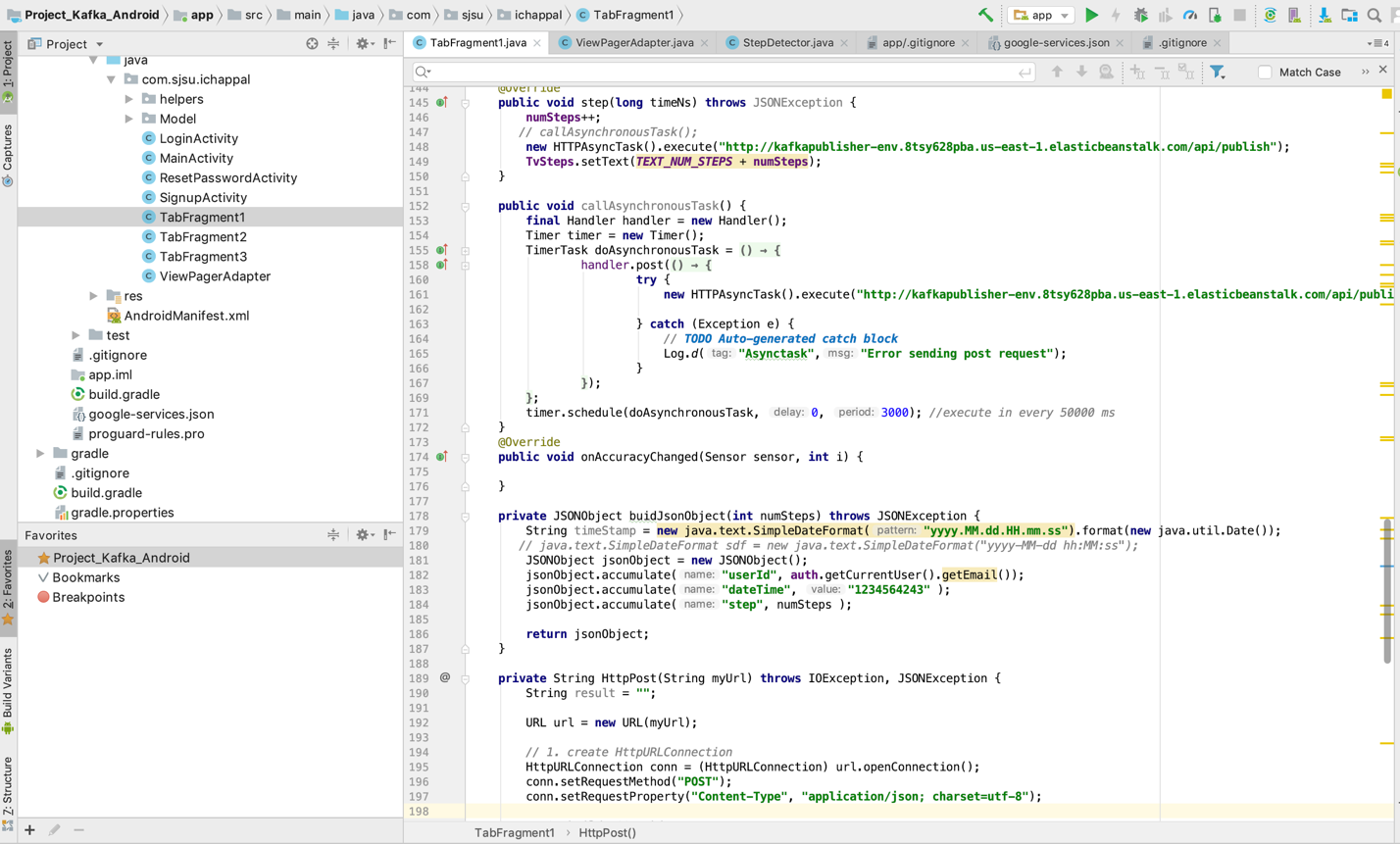


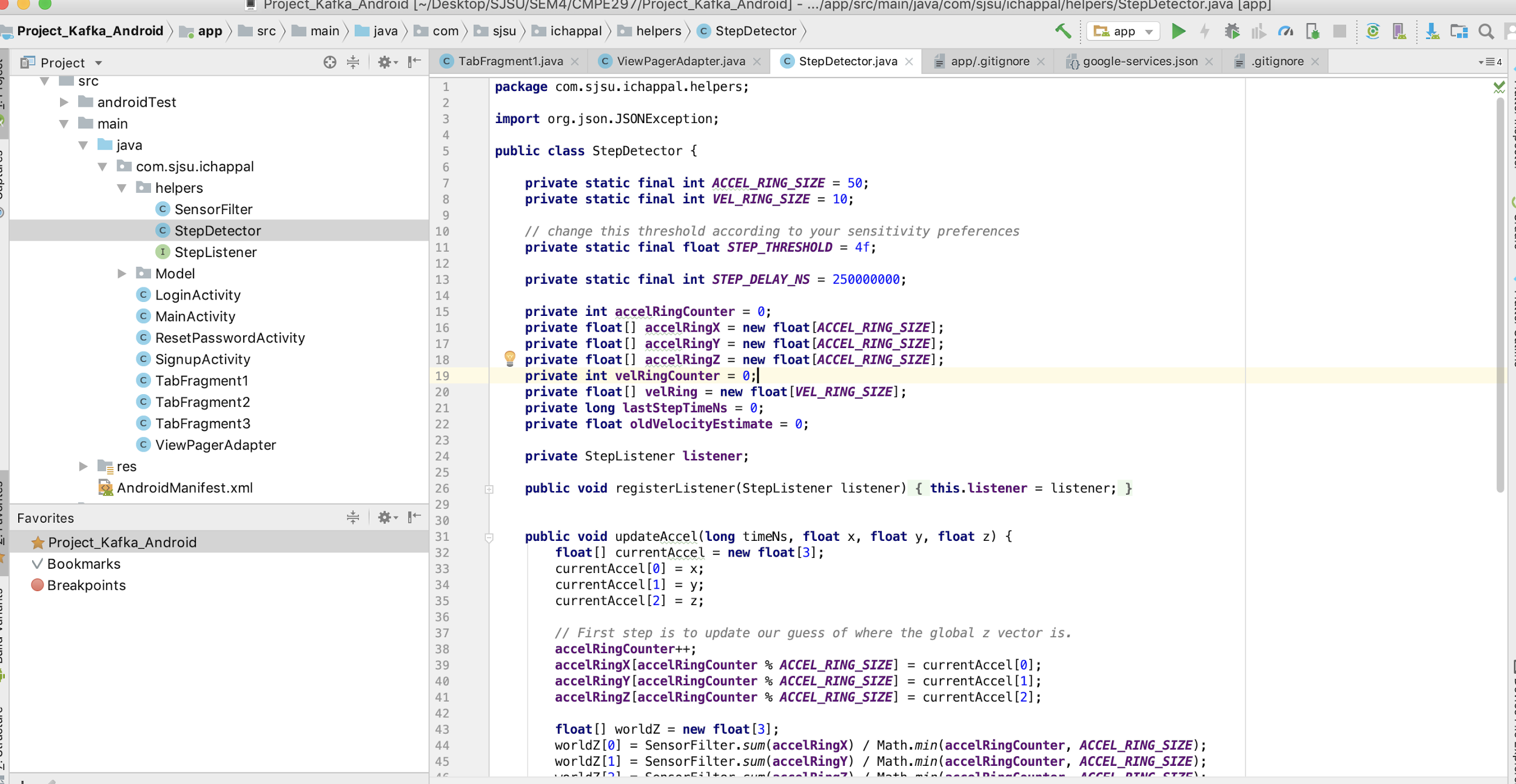


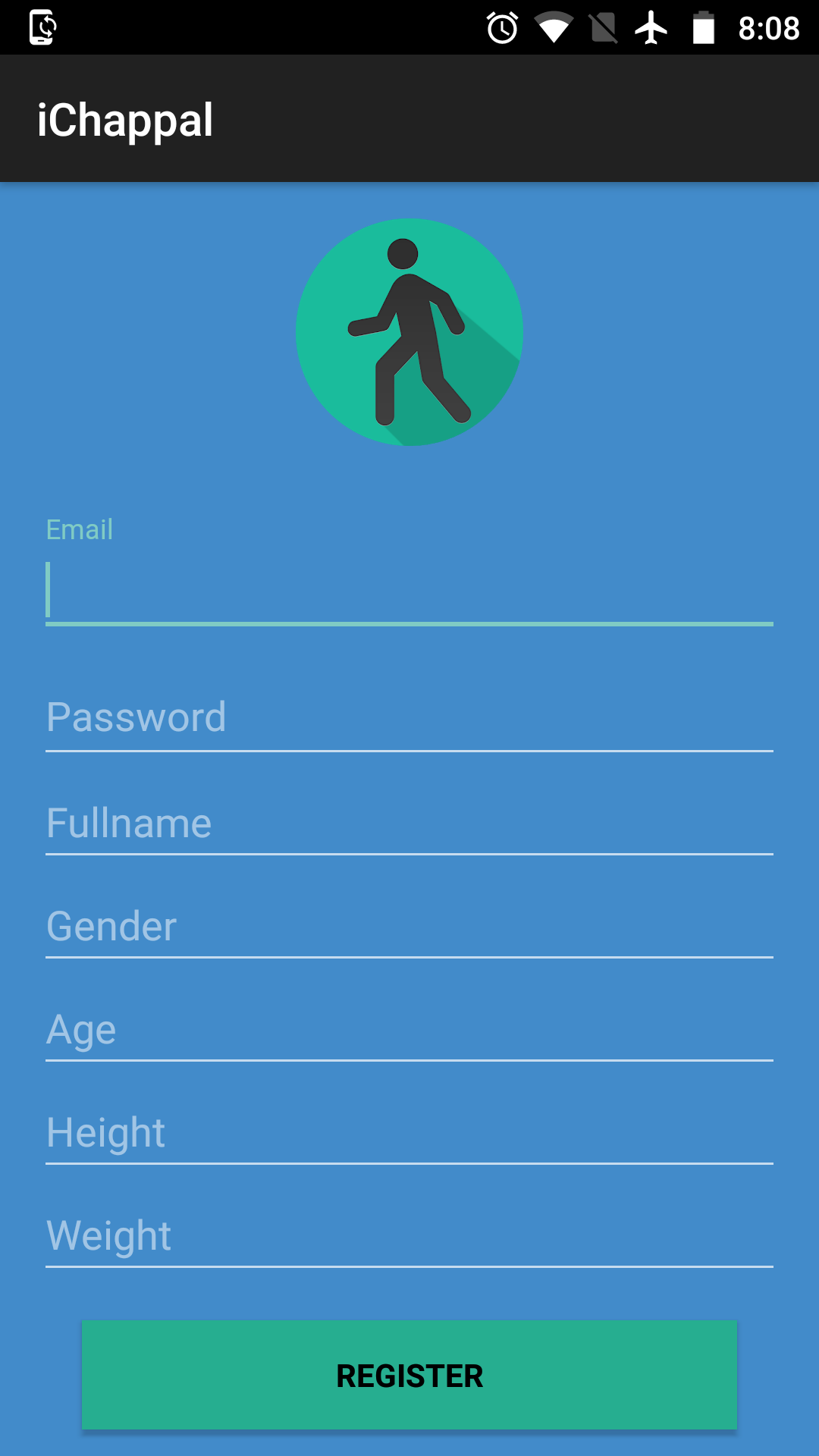


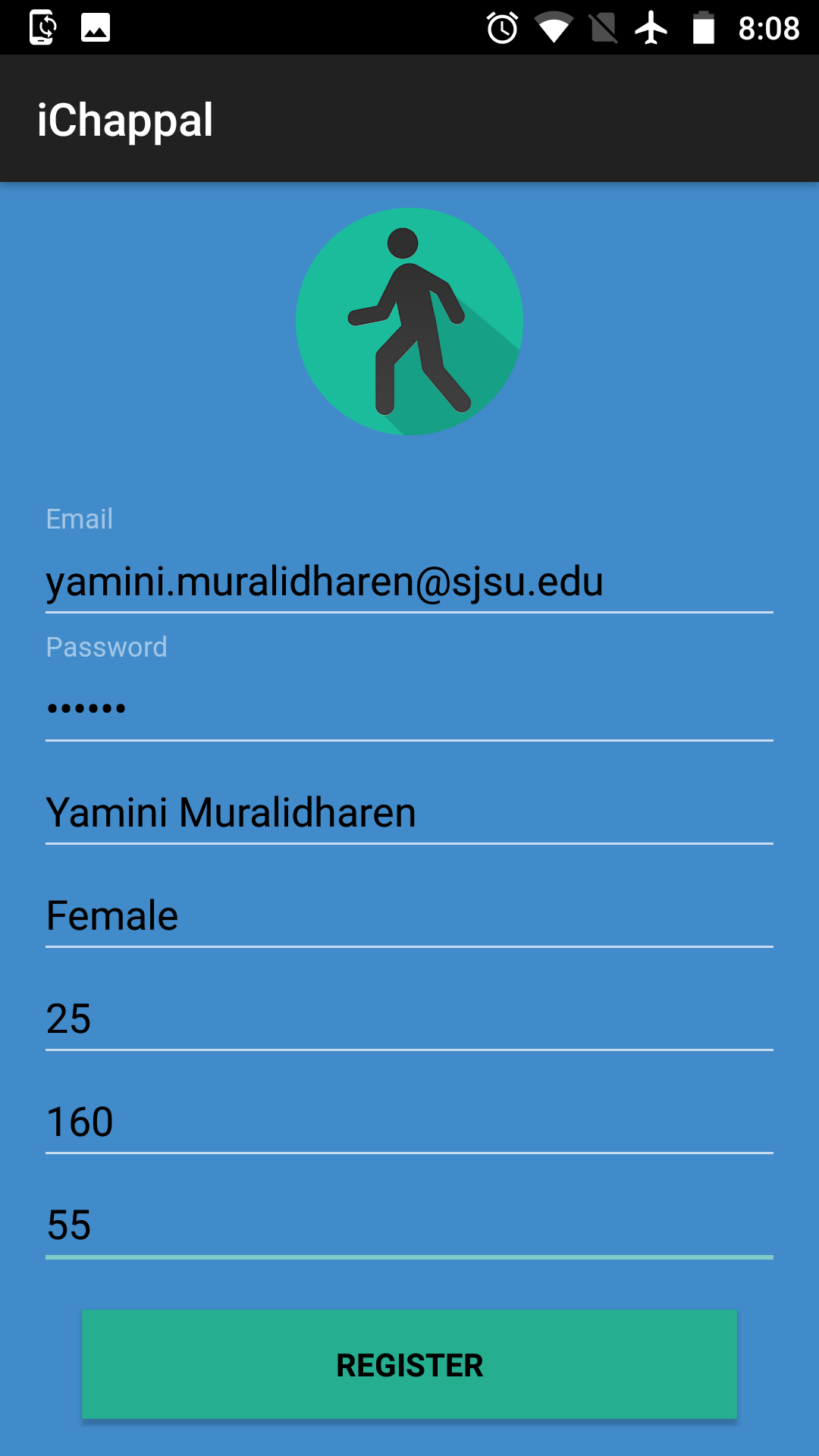


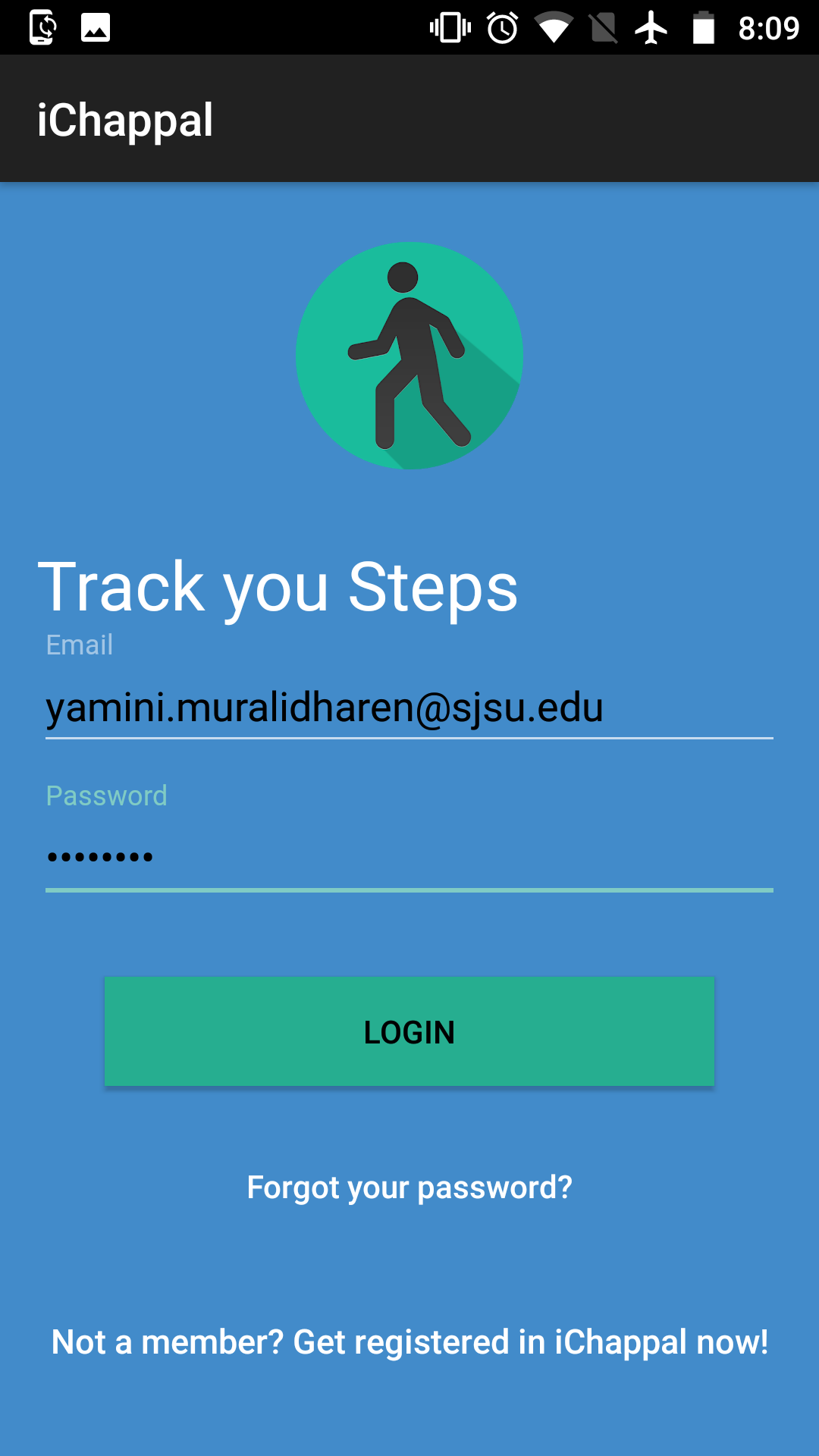


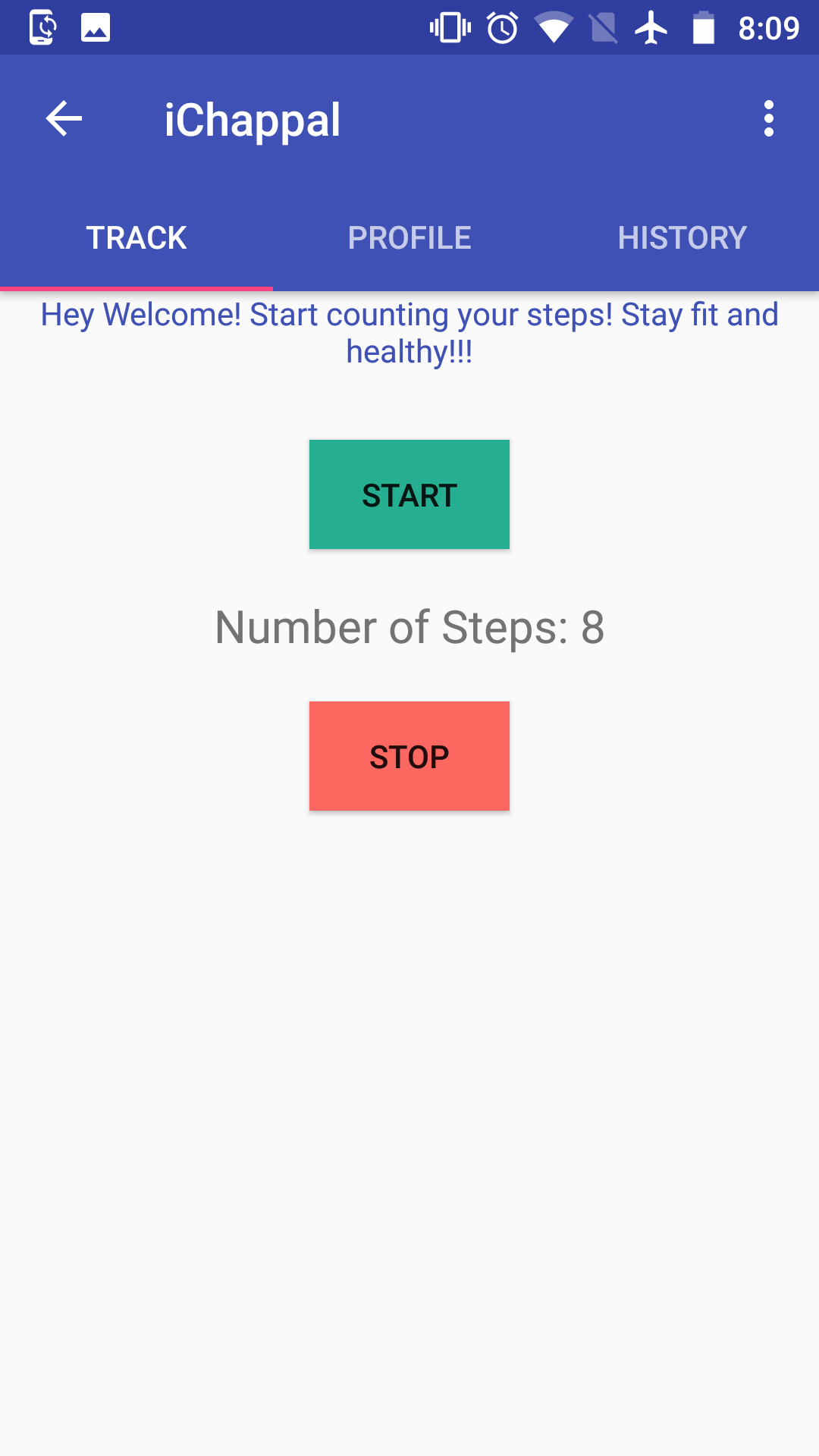
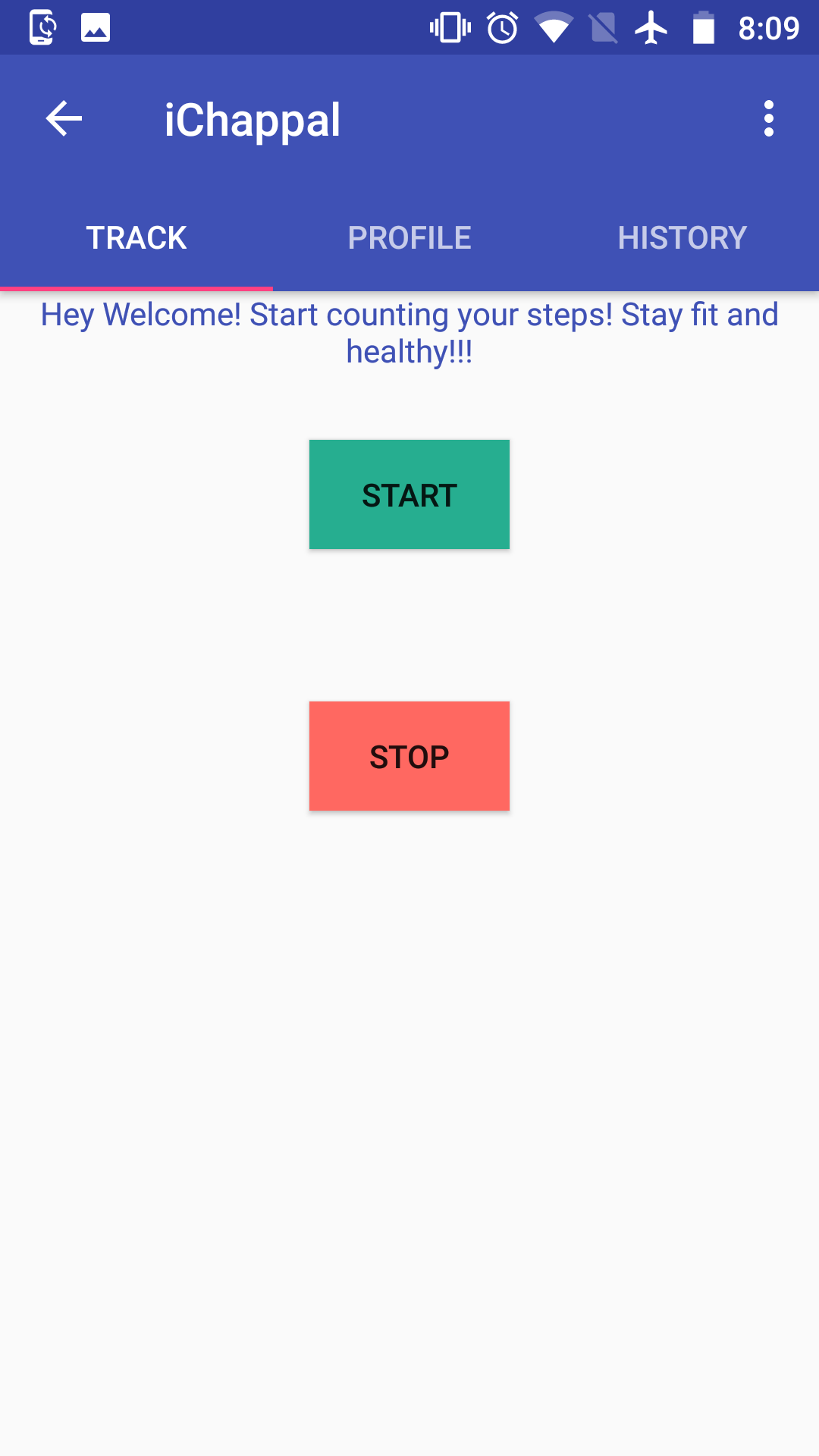
****

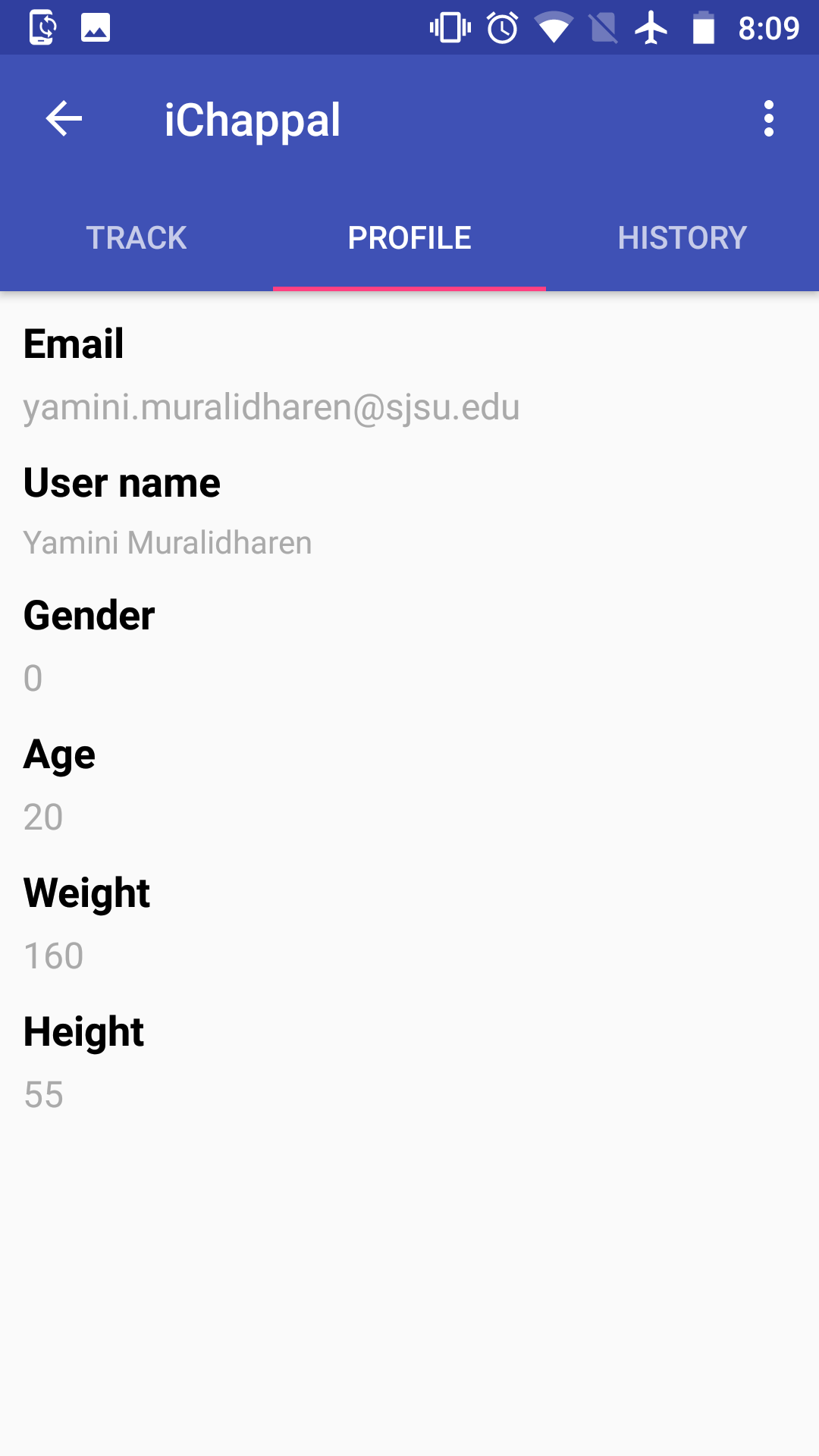
****

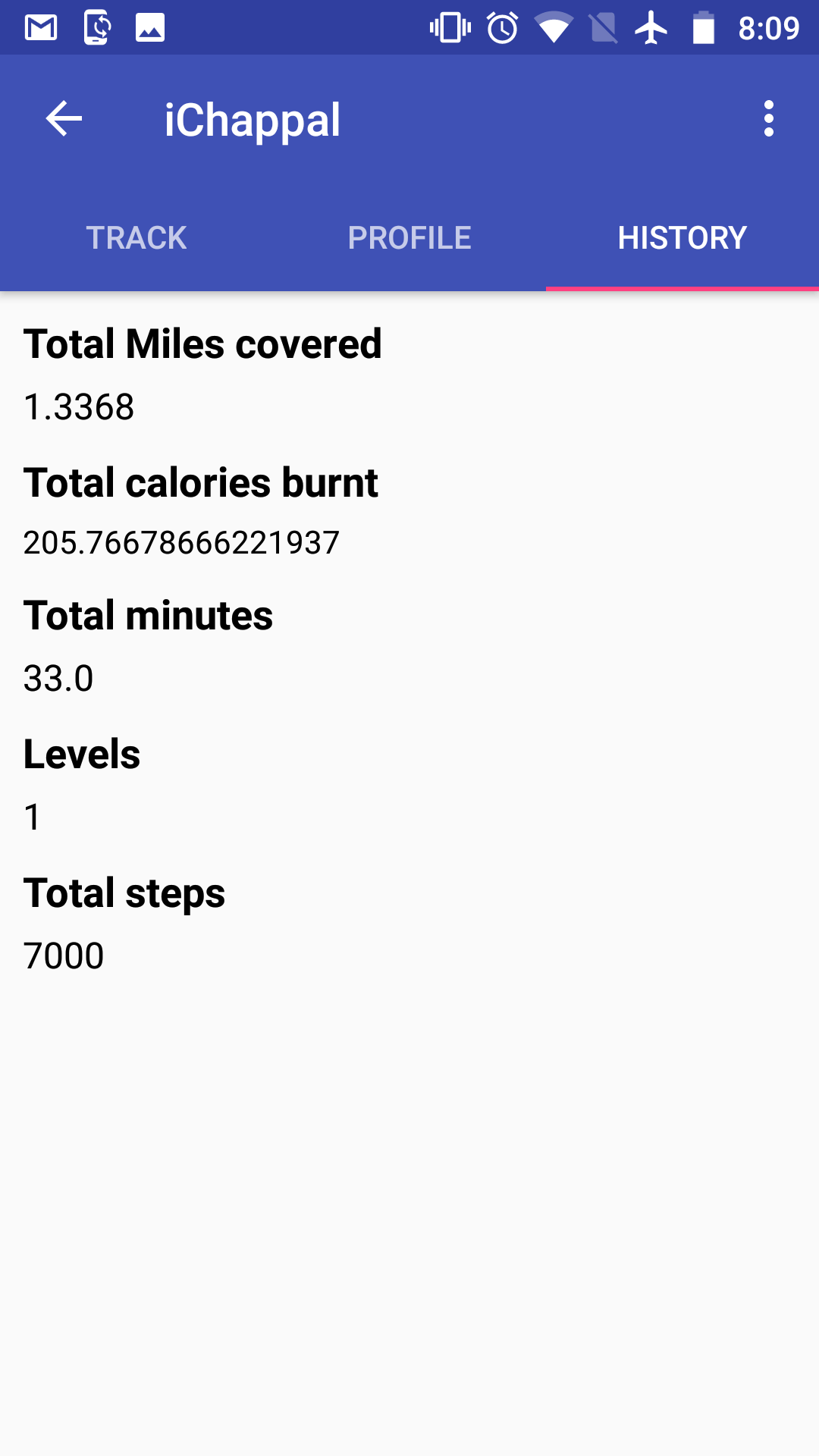


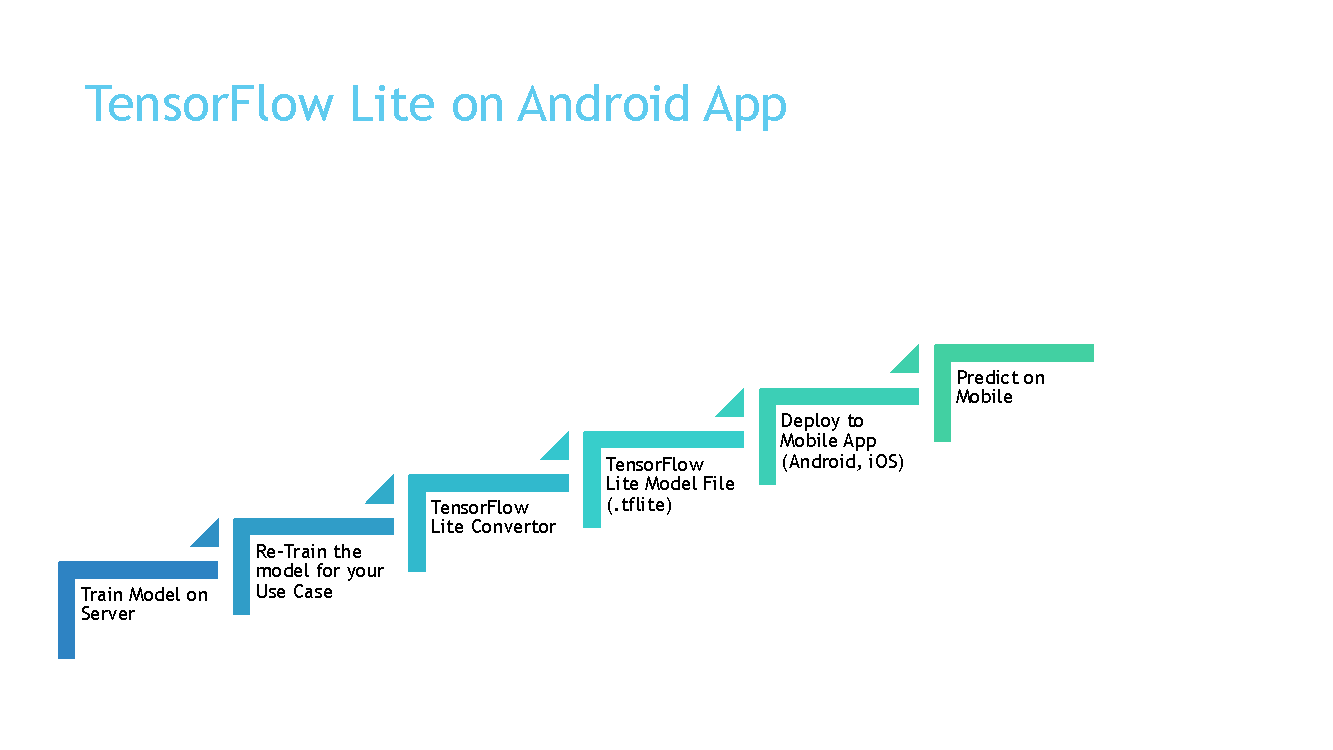


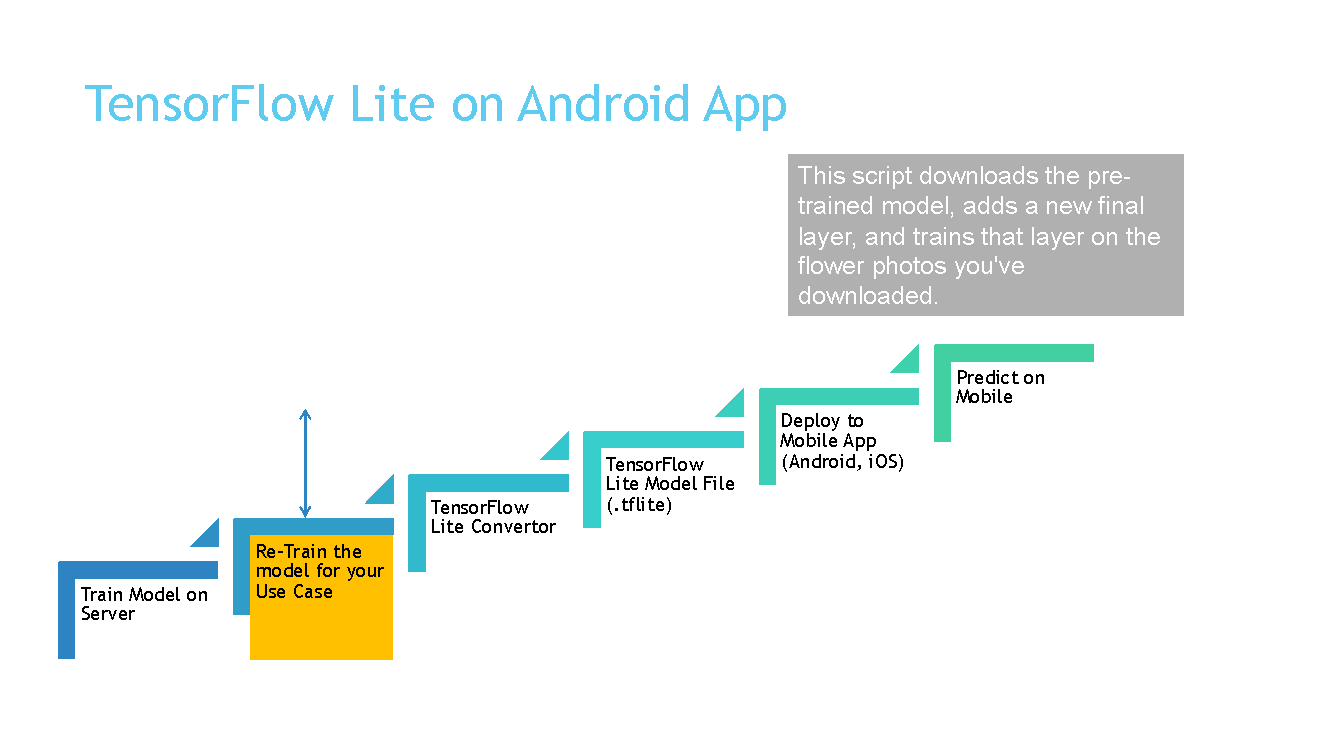


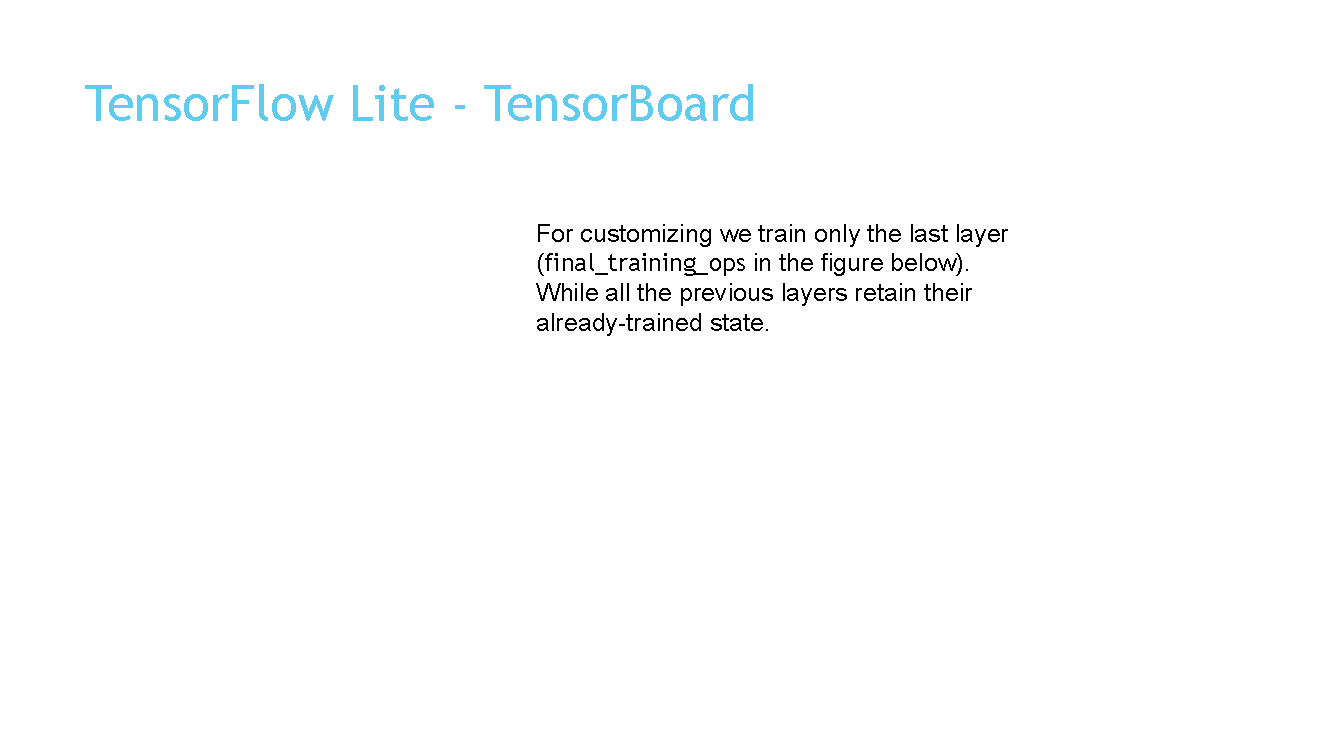


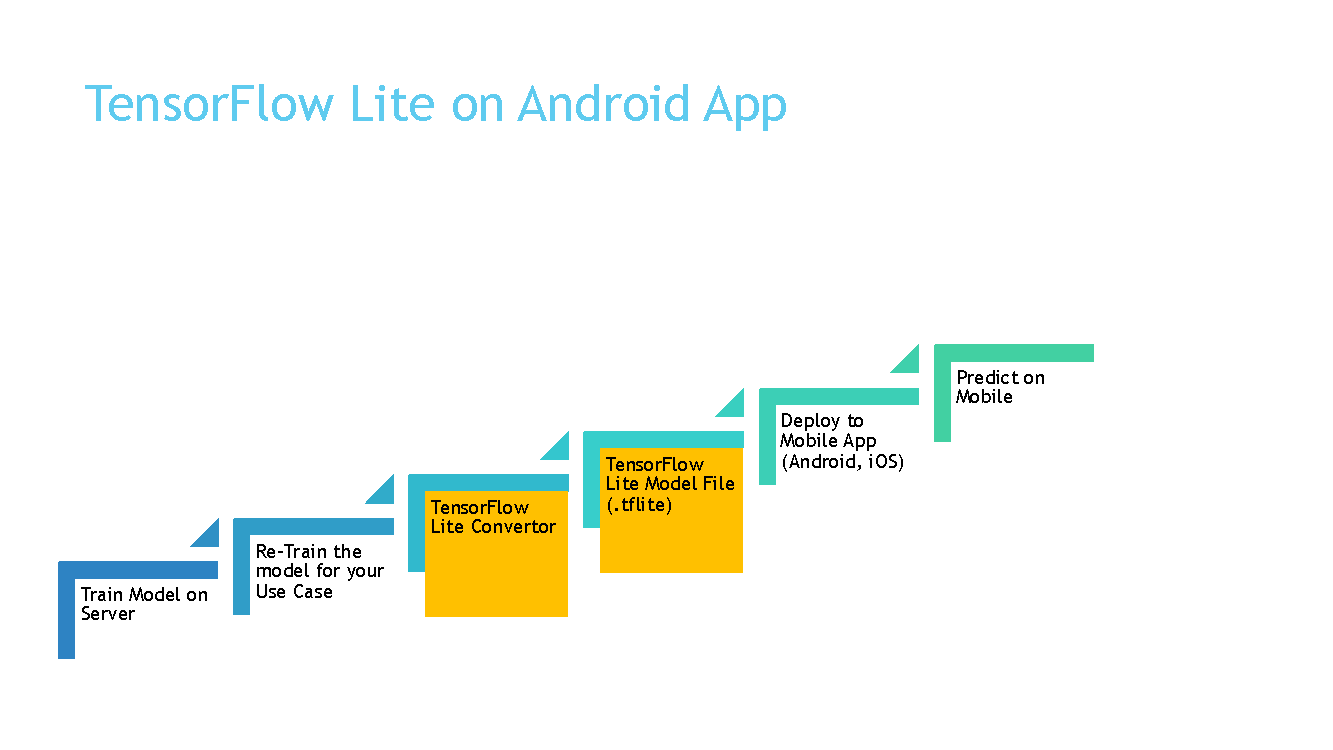


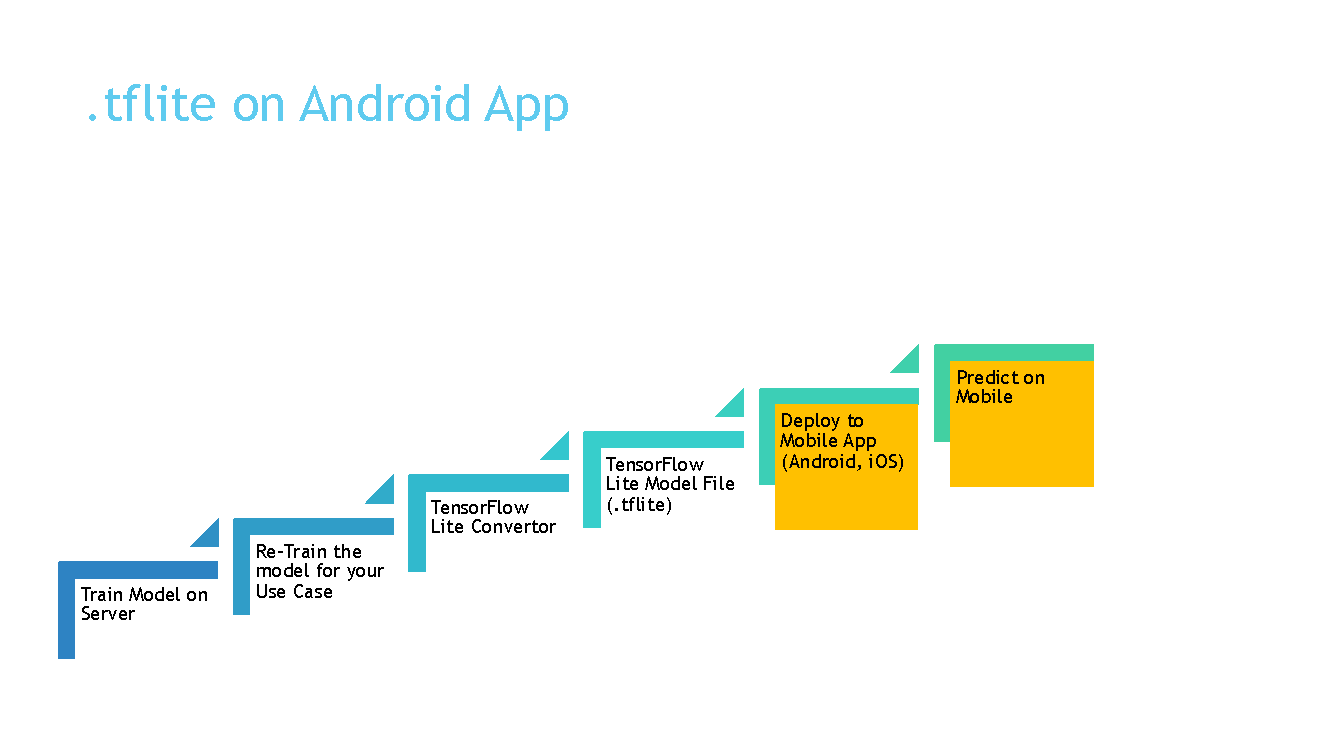


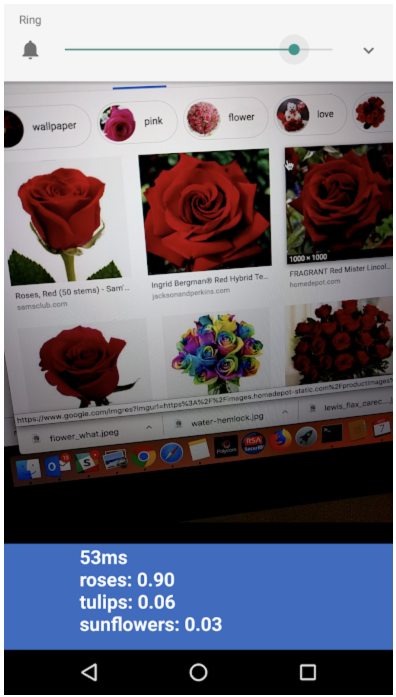


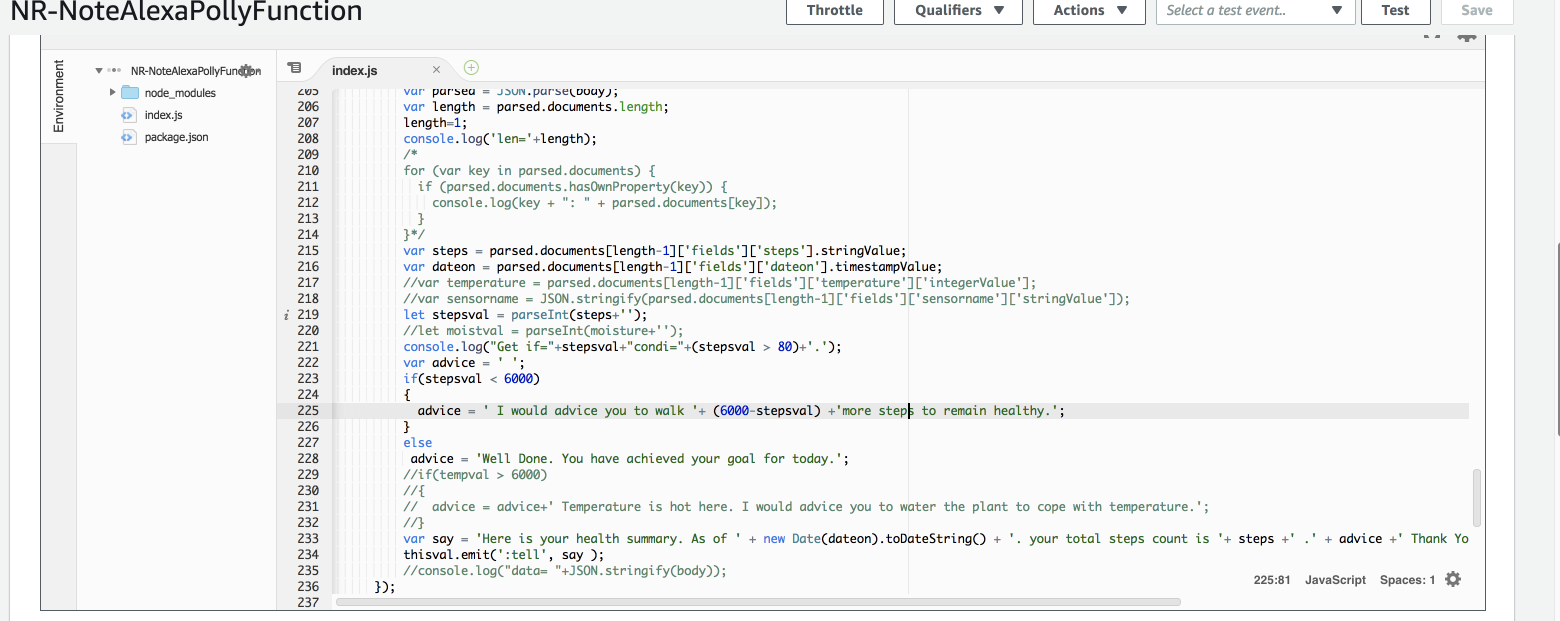


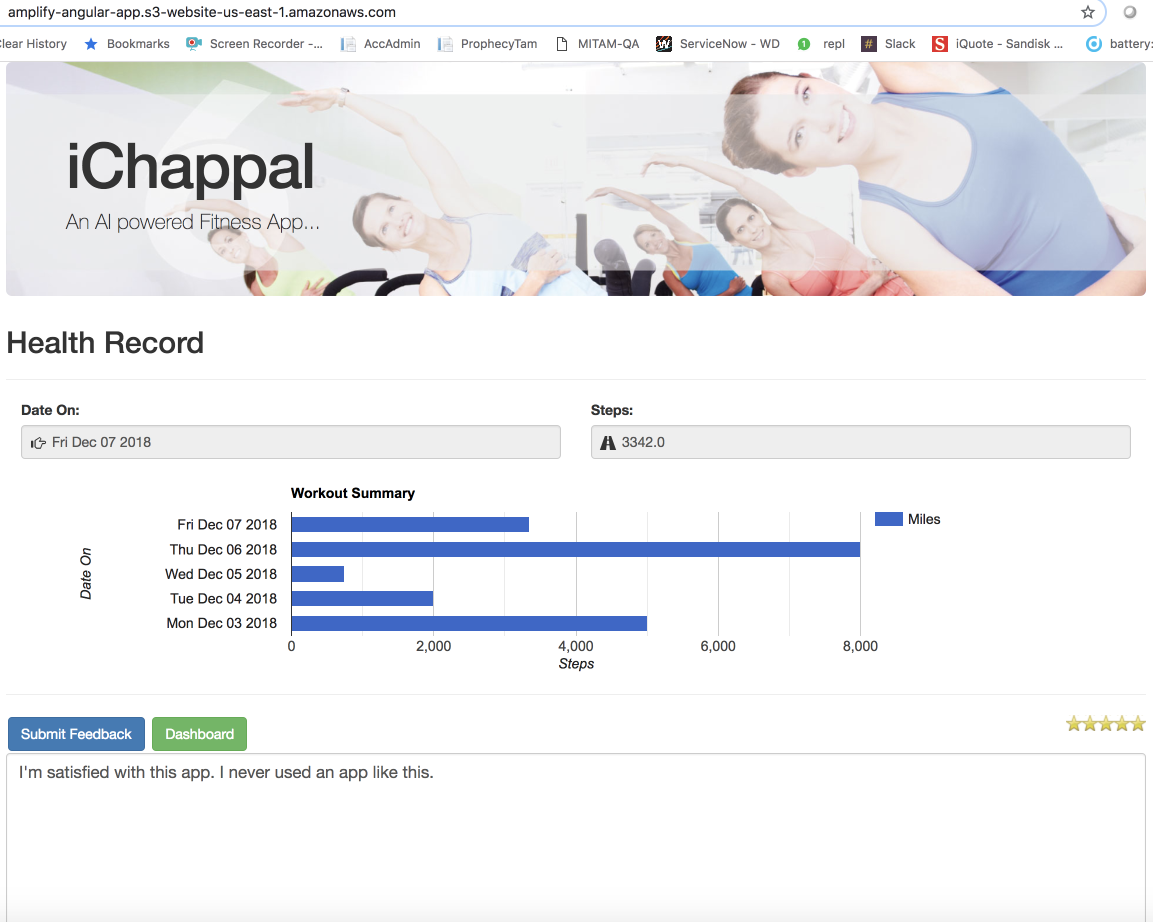


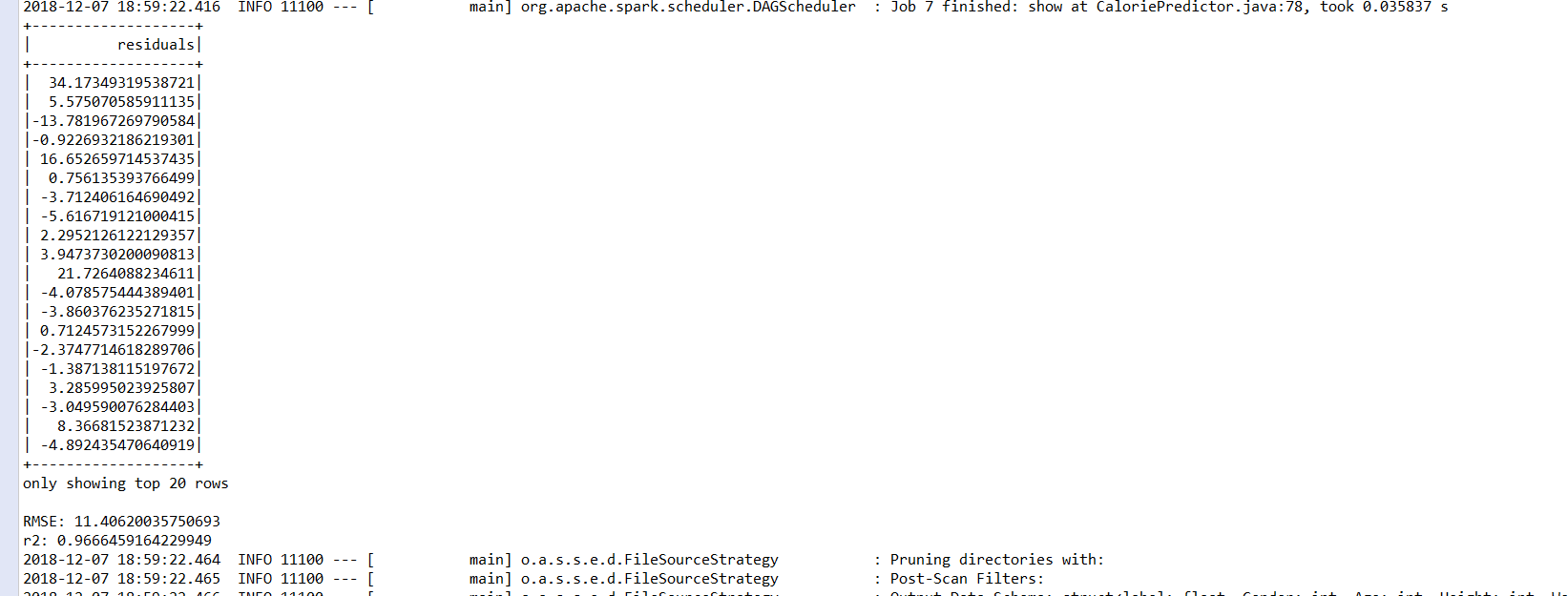


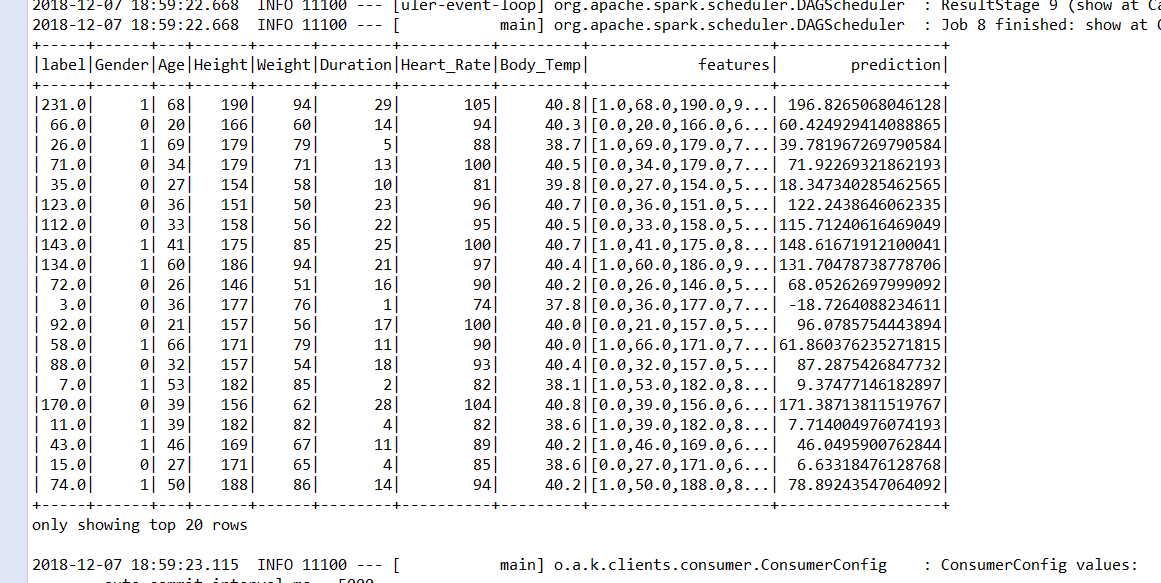


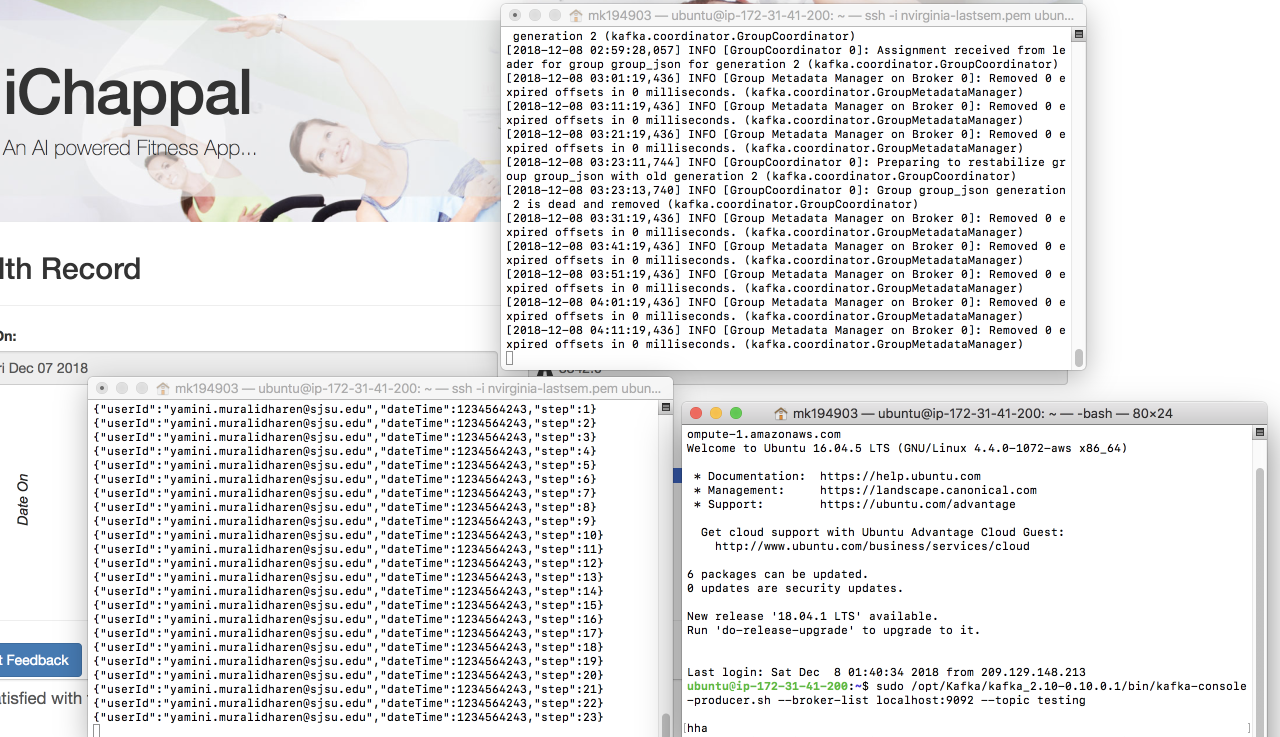












### Reference:

https://www.truiton.com/2015/06/android-tabs-example-fragments-viewpager/

https://dzone.com/articles/cost-oriented-programming-in-the-cloud