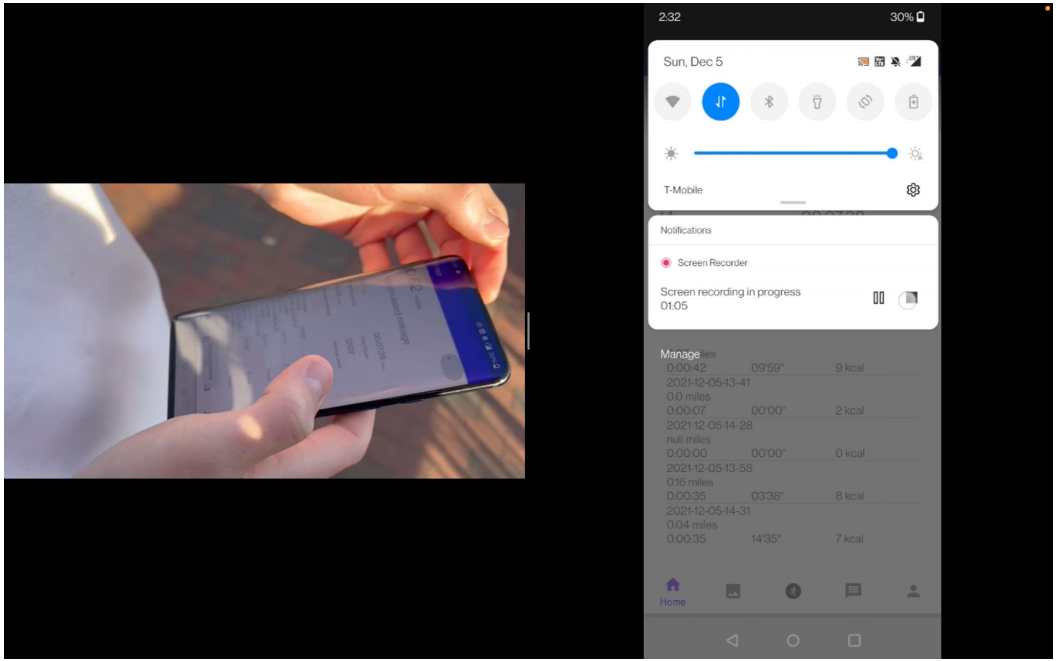
# Joggr

## Team Yellow

**Team Member:**
**Tianze Shan, Michael Andronov, Jinghao Ye, Austin Wang**

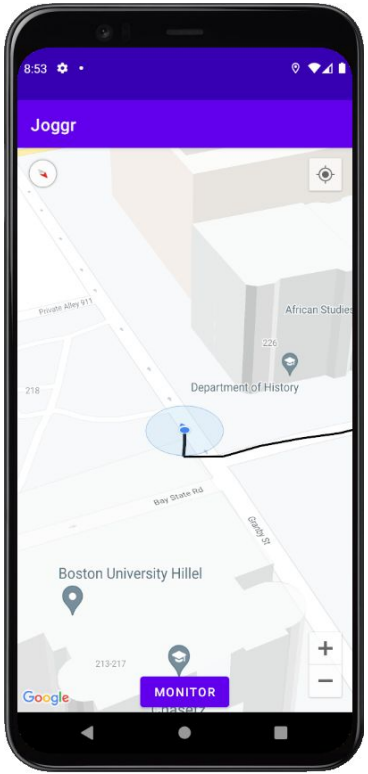
# User Case

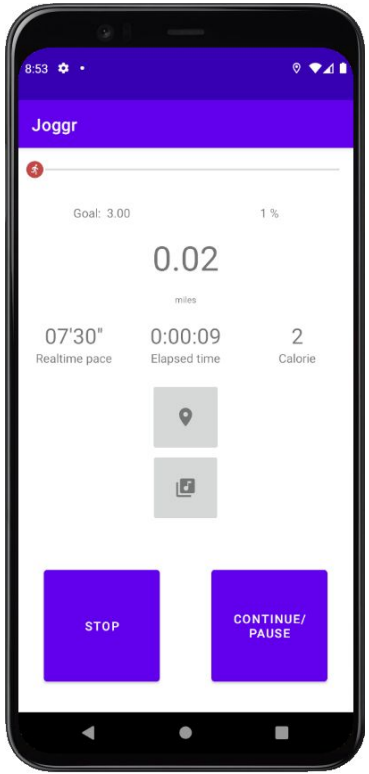

2:32
30%
Sun, Dec 5
T-Mobile
Notifications
Screen Recorder
Screen recording in progress
01:05
Manage
Home

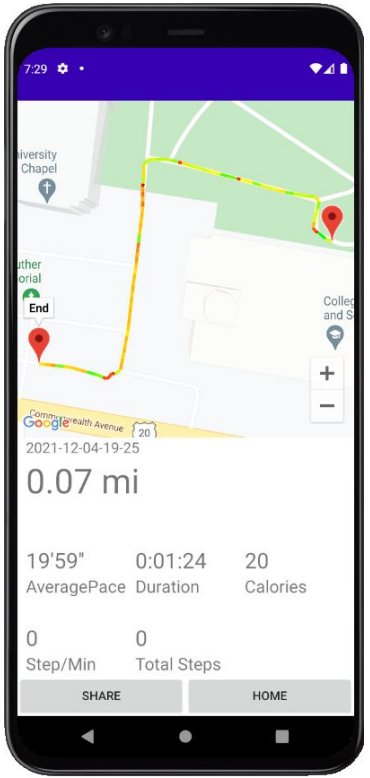
# App Introduction

- Run
- Home
- Posts
- Message
- Setting

# Run


8:53
Joggr
Department of History
Boston University Hillel
MONITOR
Google

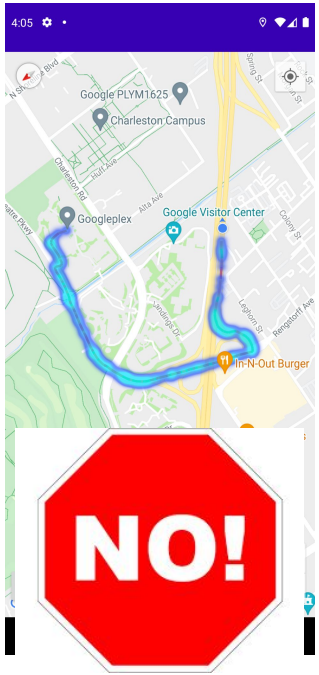

8:53
Joggr
Goal: 3.00
1 %
0.02
miles
07'30"
Realtime pace
0:00:09
Elapsed time
2
Calorie
STOP
CONTINUE/
PAUSE


7:29
End
2021-12-04-19-25
0.07 mi
19'59"
AveragePace
0:01:24
Duration
20
Calories
0
Step/Min
0
Total Steps
SHARE
HOME

# Map

- Get user’s location every second
- Keep camera follow user in a fixed direction and in a good zoom level.
- Every second, draw a line on map between location in this second and location in last second
- Every second, store location in a List for drawing a colorful route in report page.
- Every second, update real-time information such as duration, pace, and etc.

# Roadblock




## Route Heatmap Logic

LocationList = {(lat,lon),(lat,lon),(lat,long),.....,(lat,lon)} ***Every second, we get one more (lat,lon) in the LocationList.***

speed = {Distance between two (lat,long)} / 1 (meter/sec) ----------> speedList[]

Weight (percentage) = (current speed- min speed)/(max speed - min speed)

Color of speed = ColorList[weight * ColorList.length]

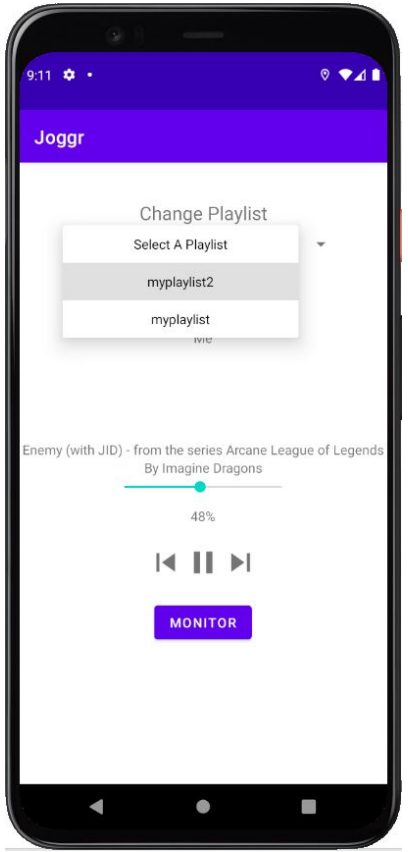
Min speed

Max speed

ColorList = { (255,0,0), (255,1,0),.....,(255,255,0),(254,255,0),..... (0,255,0)}

# Spotify

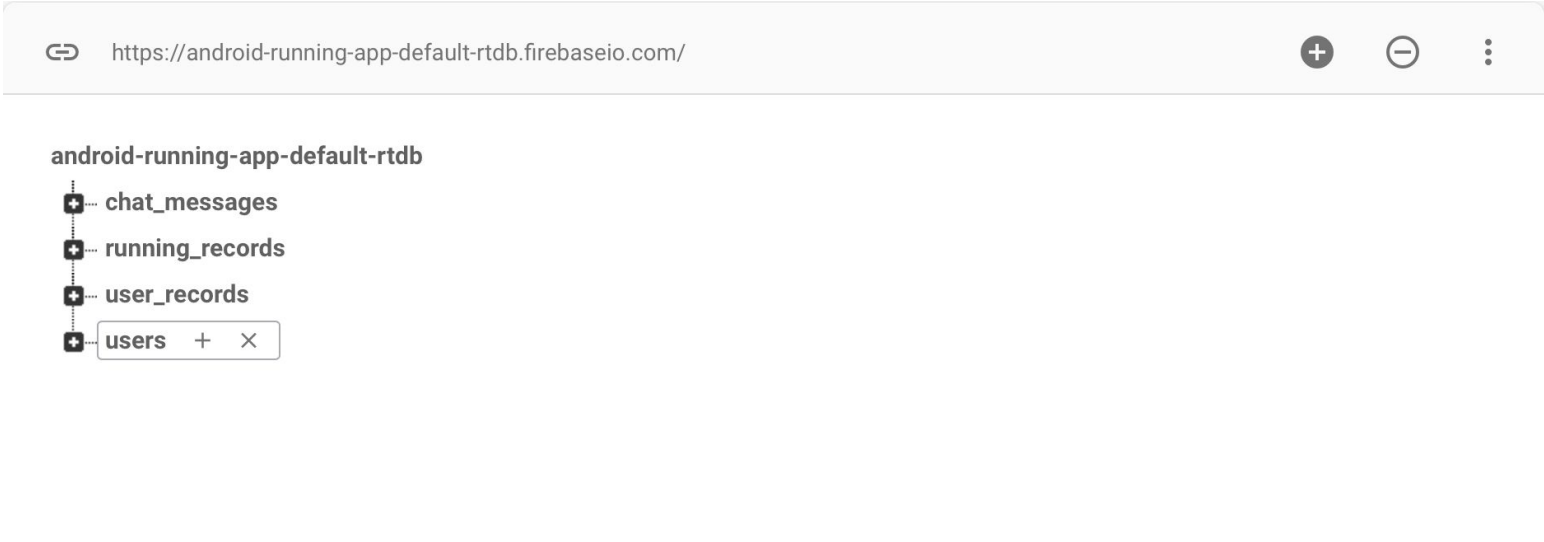
- **Spotify App Remote Library**
  - Connect to Spotify player
- **Spotify Authentication Library**
  - Get Access token
- **Spotify Web Api**
  - Get User’s data



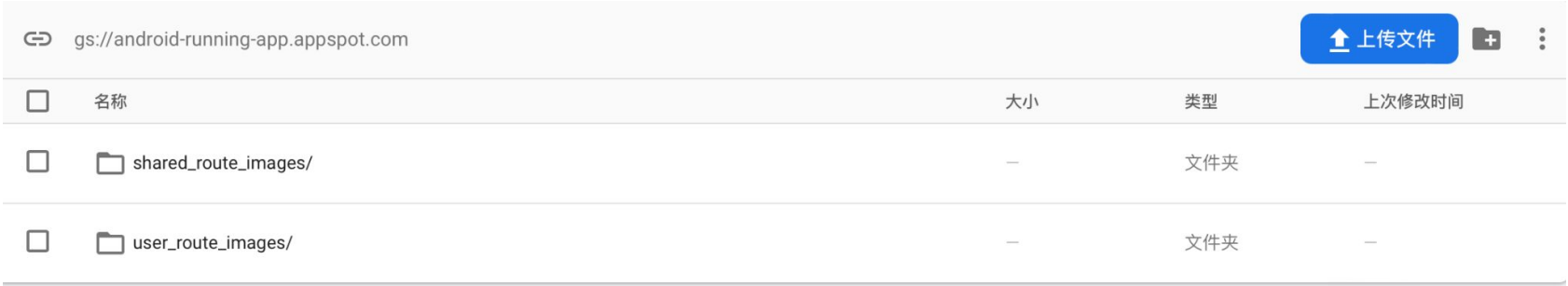
# Firebase Library

- Firebase Auth
- Firebase User
- Firebase Database
- Firebase Storage

# Firebase Database Structure



# Firebase Storage Structure



# Struggles

- Premium account is required to play/stop/select playlists
- Spotify API has an user and access limit - with more users, neet to apply for extension

From the spotify website:

## Building your app with rate limits in mind

Every app is different and you'll want to plan your app architecture and user experience with rate limits in mind. Here are a few techniques that can help you design an app that works well with Spotify's Web API rate limits:

### Apply for extended quota mode

If your app is meant to be used by many Spotify users at the same time then you should apply for extended quota mode. Apps in this mode have a rate limit that is much higher than apps in development mode, the default mode for new apps. You can apply for a quota extension by opening your app detail page in the Developer Dashboard and tapping on the Request Extension link.

# Future Work

- Better sorting filter in different ways (trends, like)
- Animation of route playback
- Professional statistic analysis
  - Distance ran for a marathon
  - Goal system
- Refined tracking features
  - Increase location accuracy
- Enhance social features
- Styling
- Commercial :
  - Work with sport brand to advertising their product
  - Let user register their sneaker type so that system can rate what is the popular sneaker for which type of running goal
  - Let user who achieve certain mileage to get discount on premium spotify membership in the next month

# Reference

- Google Map API https://developers.google.com/maps/documentation/android-sdk/overview
- Spotify SDKs https://developer.spotify.com/documentation/android/
- https://github.com/kaaes/spotify-web-api-android
- FireBase