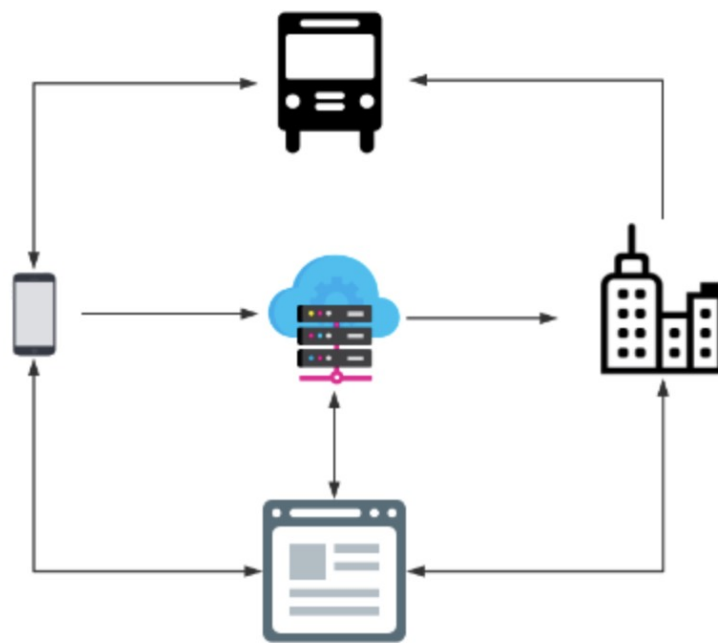


Introduction

- Based upon the current scheduling system and NextBus API
- Android App, Firebase Database, Web App
- Three Teams for Three Platforms
- Addresses common complaints with transit services
- Crowd-Sourced Data Gathering
- User-friendly UI and Robustness

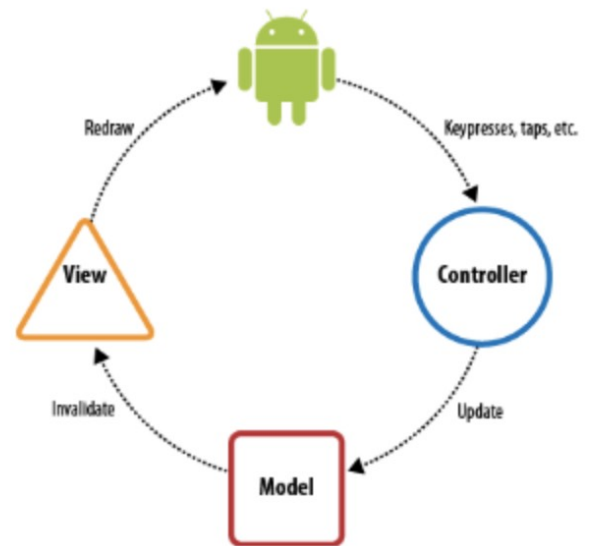


System Architecture



Android

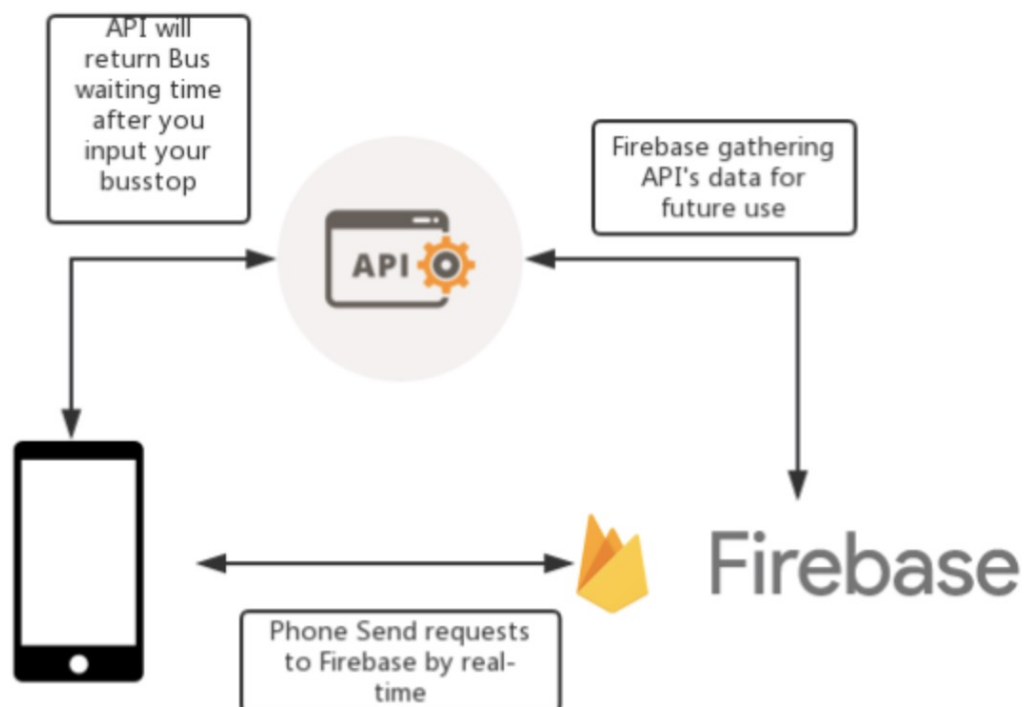
- Modified MVC Architecture
- Android 8.1 Oreo, SDK 27
- Google Maps & Play Integration
- User Feedback Collection via JSON Firebase sync



UID



Firestore Integration



Database

json

CSV

```
{
  "RUID": 117016396,
  "CurrentBusStop": "Quads",
  "DestinationBusStop": "Werblin Back Entrance",
  "Route": "B",
  "Date": "03/21/2017",
  "Time": "15:49"
},
```

RUID	CurrentBusStop	DestinationBusStop	Route	Date	Time
117016396	Quads	Werblin Back Entrance	B	03/21/2017	15:49
195828016	Werblin Back Entrance	Livingston F B	B	03/21/2017	17:34
175032734	Hill Center	Livingston S B	B	03/21/2017	18:01
117005647	Science Building	Library of Science Building	B	03/21/2017	10:03
152459992	Library of Science Building	Busch Student Center	B	03/21/2017	13:23
144030104	Busch Student Center	Science Building	B	03/21/2017	1:35
173651758	Busch Student Center	Quads	B	03/21/2017	2:17
119535185	Livingston F B	Busch Student Center	B	03/21/2017	18:47
130774558	Livingston S B	Hill Center	B	03/21/2017	16:51

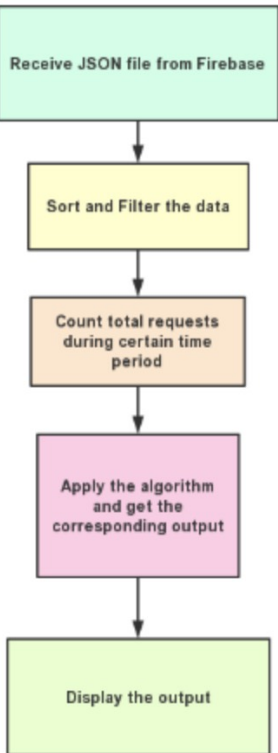
DataStructure: json tree



Algorithm Implementation

Flow

Timeline



April 2nd, 2018, Monday

April 2nd, 2018, Monday

April 2nd, 2018, Monday

April 2nd, 2018 Monday
March 26th, 2018 Monday
March 19th, 2018 Monday
March 12th, 2018 Monday
...

Prediction on:
April 9th, 2018, Monday

Formula:

$$\frac{(\text{Apr 2nd Requests } 10:00 - 10:59) + (\text{Mar 26th Requests } 10:00 - 10:59) + (\text{Mar 19th Requests } 10:00 - 10:59) + (\text{Mar 12th Requests } 10:00 - 10:59)}{4} = \text{Estimated Apr 9th Requests } 10:00 - 10:59$$

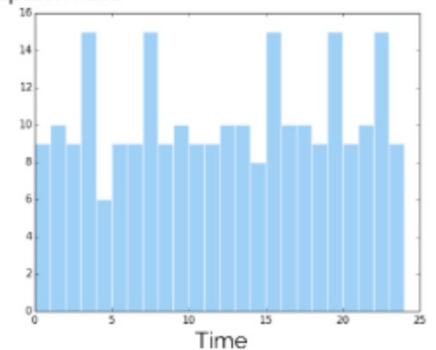
$$\frac{\text{Week } (n) \text{ Time } (X) \text{ Requests} + \text{Week } (n-1) \text{ Time } (X) \text{ Requests} + \text{Week } (n-2) \text{ Time } (X) \text{ Requests} + \dots + \text{Week } (1) \text{ Time } (X) \text{ Requests}}{n} = \text{Estimated Week } (n+1) \text{ Time } (X) \text{ Requests} = K$$

Algorithm Output: (for future use)

Time hour from: 10 : 00 to 10 : 59
Arranged bus dispatch rate : Every 9 minutes
Time hour from: 11 : 00 to 11 : 59
Arranged bus dispatch rate : Every 9 minutes
Time hour from: 12 : 00 to 12 : 59
Arranged bus dispatch rate : Every 10 minutes

Data analysis

BusDispatch Rate



Improvements

Current:

- Implementation of the **map function** in UI, user friendly
- Accomplished implementing code to deal with **requests abuses**

Future:

- **Initiating Firebase/Server integration with the Web** so we can reach a complete circle of our project
- **Improve** app map function, automatically acquire current bus stop
- **Imitating** more student requests from different route/time/stop locations, **improve algorithm** for data visualization and analyzation
- **Involving Machine Learning** to analyze data

You can also look our Slides on Prezi:
<https://prezi.com/p/-xvwqnhrqhh/>