

An Ellensburg taxi app that's
out of this world...



Tyler Lloyd
Junyu Lu
Austin Richardson
Jose Rodriguez
Stephen Stengel
Supervisor: Szilard Vajda
Client: Kenneth Swanson

Rodeo Town Taxi App

From RODEO Speedwagon

Table of Contents

- App Introduction: 3
- Requirements: 7
- Progress: 10
- Analysis: 17
- Design: 21
- Map: 26
- Tools: 28
- Scheduling: 35
- Messaging: 37
- Testing: 39
- Conclusion: 41

App Introduction

Rodeo Town Taxi

- 24-hour service
- Taxi rides, deliveries, car lockouts, jump starts
- First-aid trained, background checked drivers



A New Era

- The services are currently only available by phone call
- Compatible with iOS and Android devices
- In line with 2020 standards



The Competition

- Not common in Ellensburg
- Rodeo Town is a locally owned business



Uber

Requirements

Absolute Requirements

- Book a taxi ride (IP)
- Access other services offered by Rodeo Town Taxi (IP)
- Dispatch button (C)
- Storing driver information (C)
- Administrative control over drivers (C)
- User login (C)
- Driver and customer locations on map (C)
- Compatible with Android and iOS (C)
- One app for all users (C)



Additional Requirements

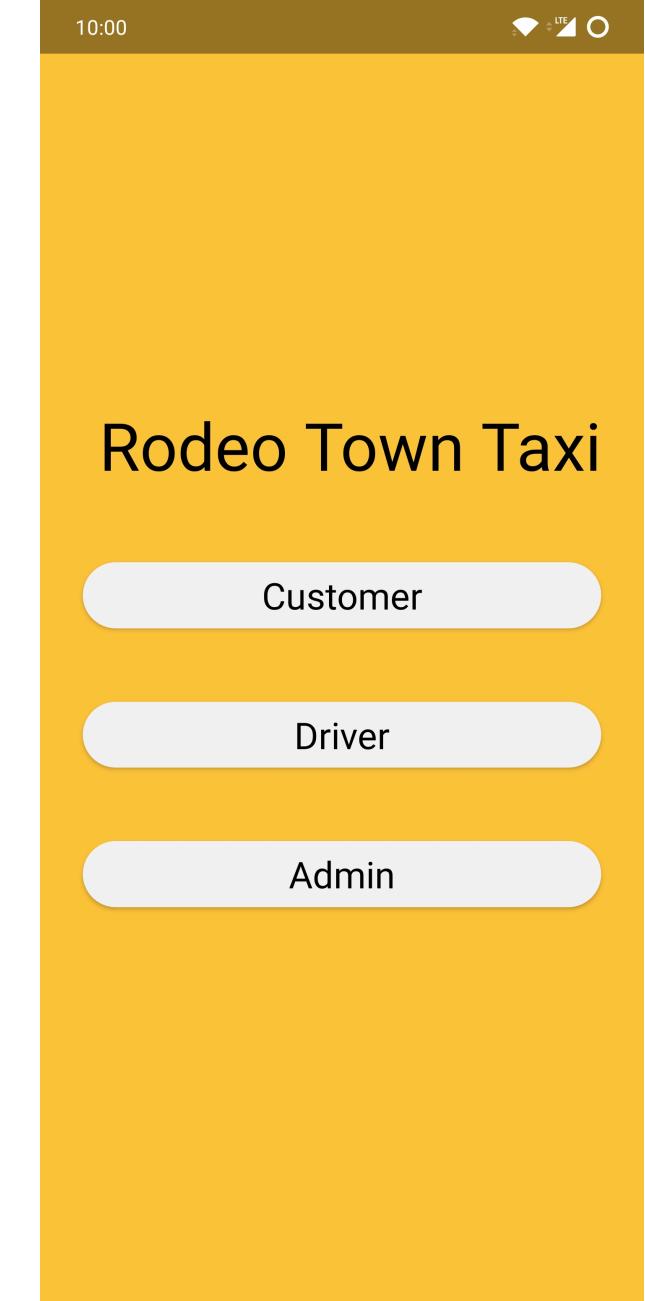
- Desirable:
 - Rewards system and promotions
 - In-app messaging (IP)
 - Additional information section (IP)
 - Calculating wait time and trip duration
- Optional:
 - Rating System



Progress

User Roles and the Database

- Customer Role
- Driver Role
- Admin Role
- Authentication
- Cloud Firestore



User Roles and Database(2)

Customer Login

Name

Email

Phone Number

Log in

Continue as a guest

Home

Driver Login

Email

Taxi Number

Password

Log in

Reset Password

Home

Admin Login

Email

Password

Log in

Reset Password

Home

User Roles and Database(3)

Welcome Admin

Add Driver

Modify Driver

Remove Driver

View Drivers

Sign Out

New Driver

Name

Email

Phone Number

Password

Add

Back

Info of Driver to
Modify

Name

Email

Phone Number

Continue

Back

Remove Driver

Name

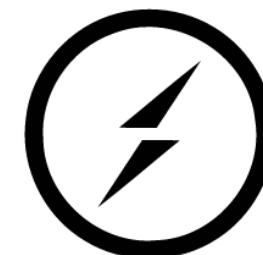
Email

Remove Driver

Done

Web Sockets

- Fast communication between Driver and Customer
- Running on our Google Cloud App Engine
- Structure: Request Type and Request Data

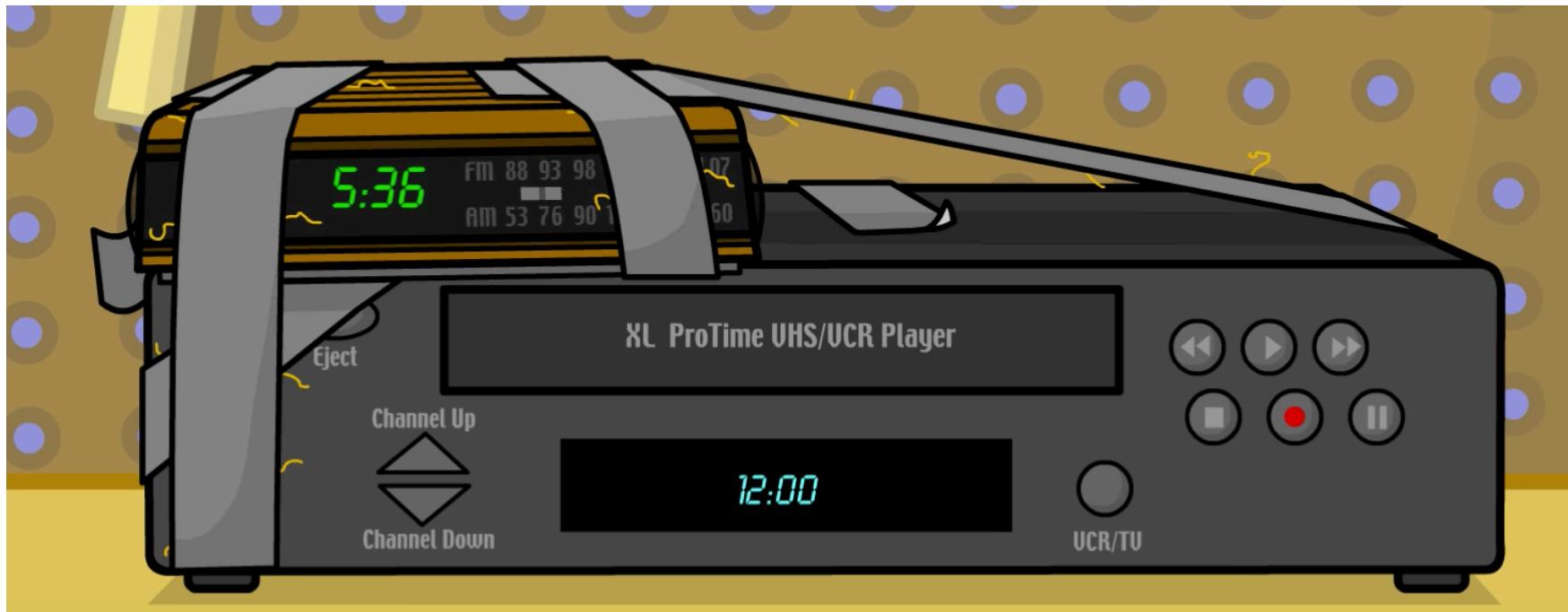


socket.io

<SOCKET DEMO>

Problems Encountered

- No prior experience with React Native or Node JS
- Finding the best way to host server code
- React Native changes constantly



Analysis

Validity

- Requirements valid for the needs of a taxi application.
- Client is a taxi industry professional for over 10 years in the business.
- Information provided correct and accurate with modern taxi application.



Feasibility

- Highest priority requirements can be completed within the provided 10-week timeframe.
- Traditional database connection and functionality.
- Basic map functionality via Google Maps API.

POSSIBLE



IMPOSSIBLE



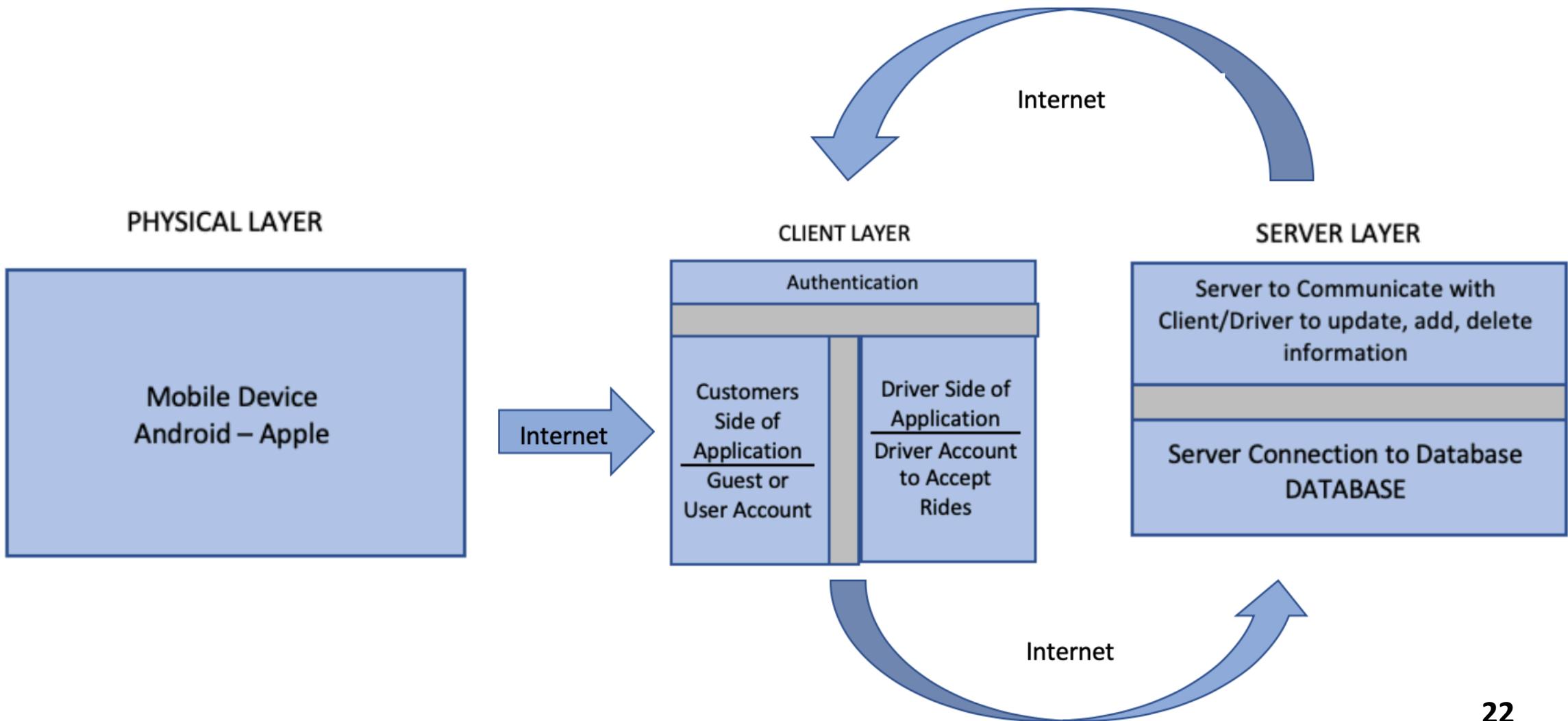
Consistency

- All goals for the application matched with the requirements specified by the client.
- Requirements are not conflicting and are logical for this type of application.

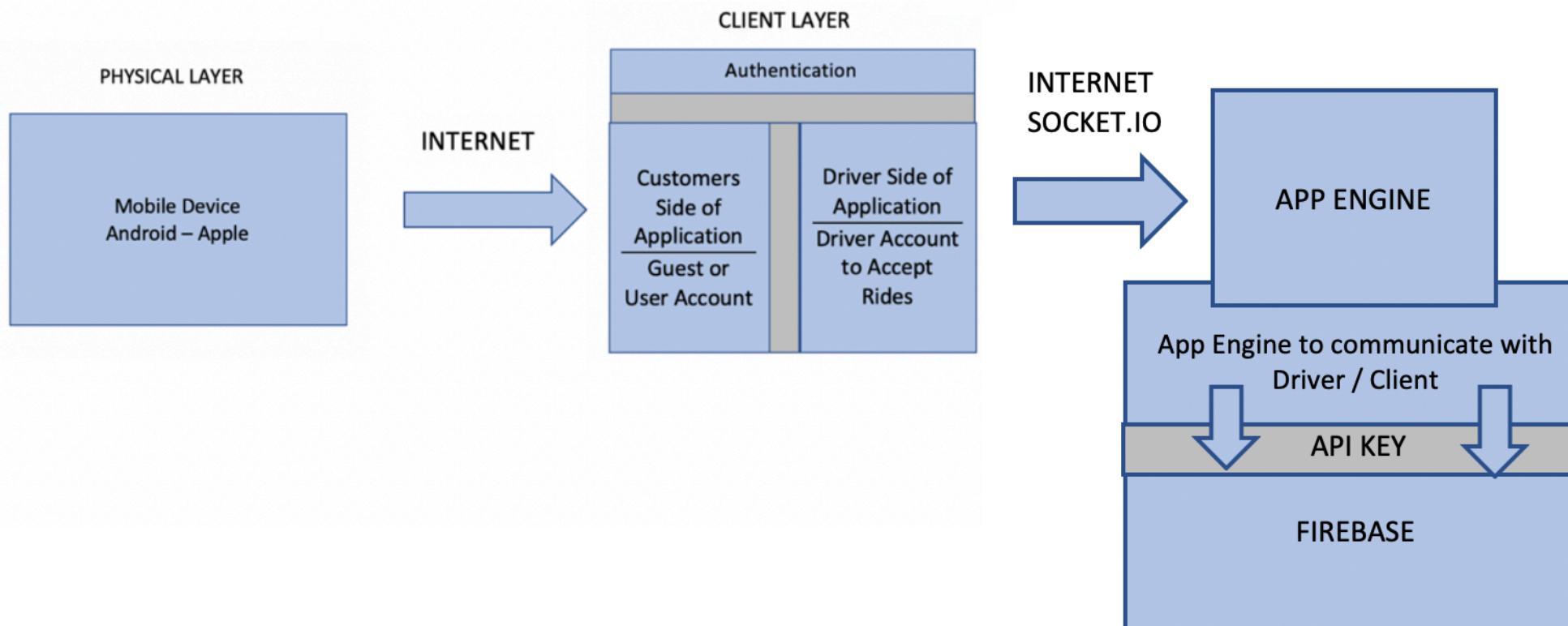


Design

High Level Design (Before)



High Level Design (Now)



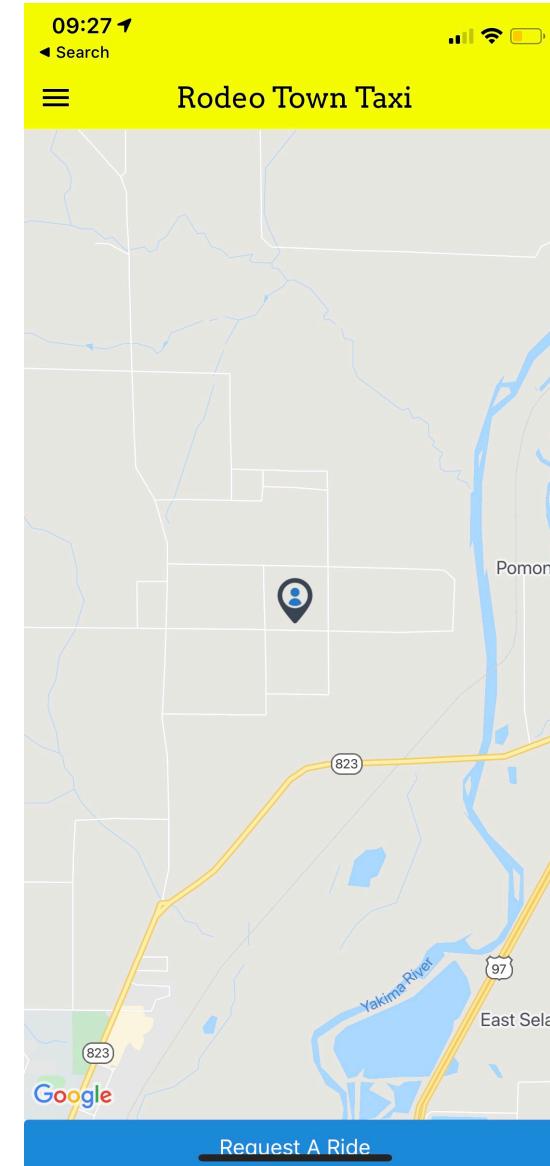
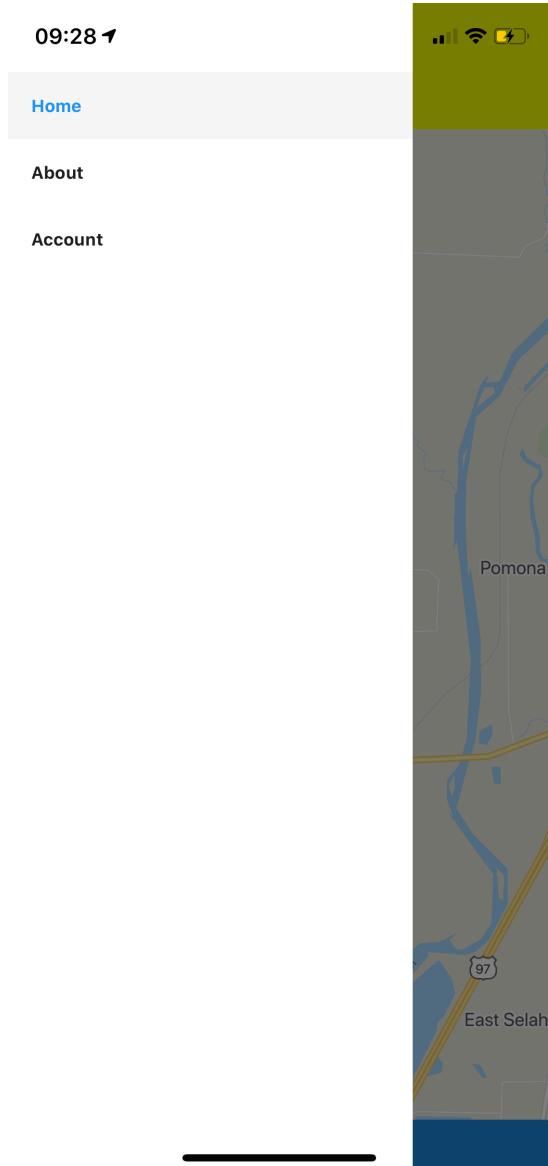
Low Level Design (Before)

- User Side – Driver Side
 - Google Maps
 - Button to call dispatch ~ Button to call ride
 - Request a Ride
 - Display marker on map
- Server Side
 - Login
 - Connection establishment
 - Request date ~ send data
 - Analyze connection
- Database
 - User Table [First, Last, Password, Guest, Email, Rides, Awards]
 - Driver Table [First, Last, Password, Employee ID, Car ID]
- App Main
 - Switch Driver or User
 - Login
 - Forgot Password

Low Level Design (Now)

- User Side – Driver Side [Separate files]
 - Google Maps
 - Button to call dispatch ~ Button to call ride
 - Request a Ride
 - Display marker on map and render while moving
 - Database
 - User Table [Name, Email, Phone Number, Password]
 - Driver Table [Name, Email, Phone Number, Password, Car ID]
 - App Main
 - Choose on how to login
 - Login Screen
 - Redirect to proper home screen; userHome or driverHome
- ❖ AppEngine to host the app and make connections with Firebase using socket.io within specific files such as the userMap and/or driverMap to update their positions accordingly and display necessary information on the map.

Map



Tools

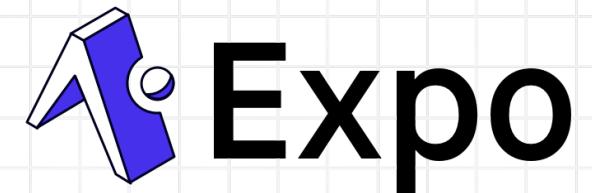
Git / GitHub

- Version control system
- GitHub acts as a free online repository
- Pros so far:
 - Has made it easy to share and run eachother's code.
 - Keeps our work separate
- Cons:
 - Merging branches is currently a problem.



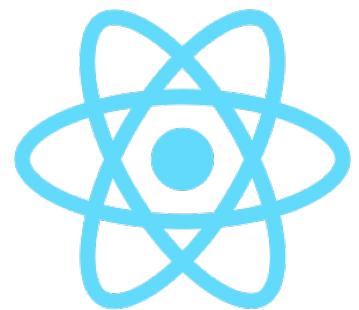
Expo

- Framework for testing apps on physical devices without needing Android Studio or Xcode
- Pros so far:
 - Pushes updates in code to phone seconds after saving
 - Can also push to emulators
- Con:
 - Sending data over the school wifi makes us sad



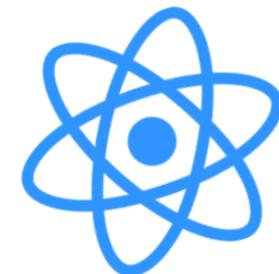
React Native

- React is a JavaScript library
- React Native lets you use React on devices. Natively.
- Pros so far:
 - One code base for iOS and Android with minimal changes
 - Many packages to do specific things
 - Lots of APIs
- Cons:
 - Library and dependency bloat



React Native Packages

- react-native-elements
 - APIs for many UI elements
- react-native-maps
- react-native-maps-directions
- react-navigation
- Firebase
- Socket.io-client



Google Cloud

- Pros so far:
 - Plenty of free credits
 - Using App Engine instead of server
 - Google mapping APIs
- Cons:
 - All too easy



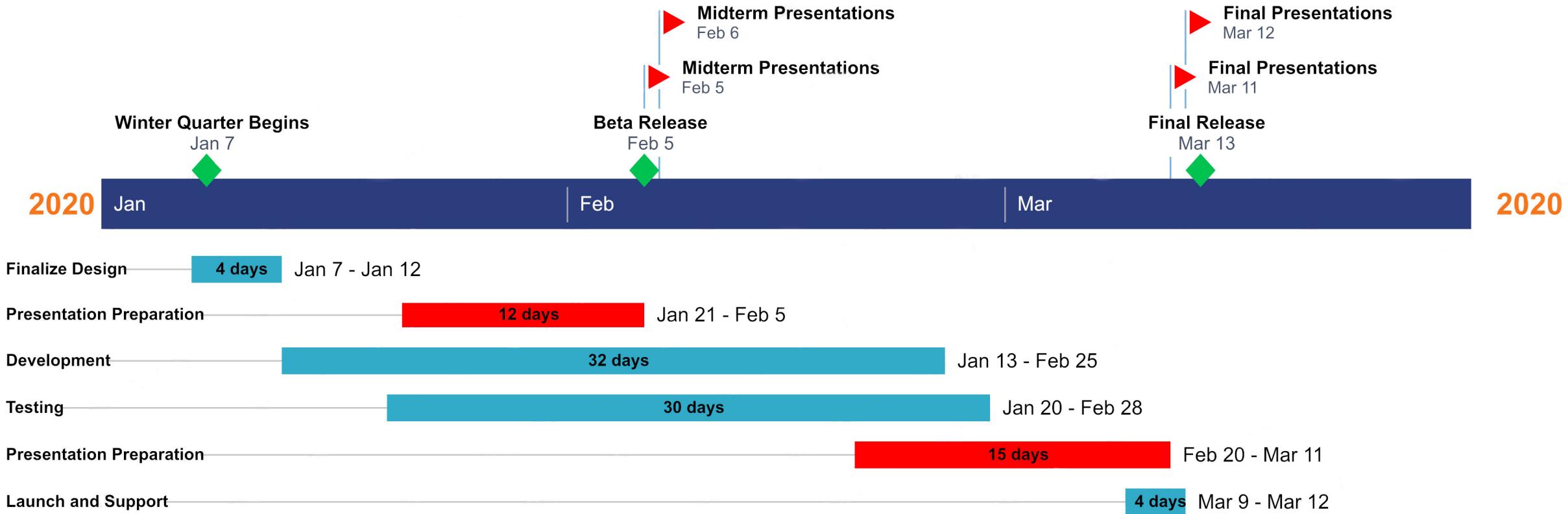
Scrum Meetings

- Pros so far:
 - Share things quickly
 - Informal
- Cons:
 - Share too quickly
 - Informal
 - No snacks



Scheduling

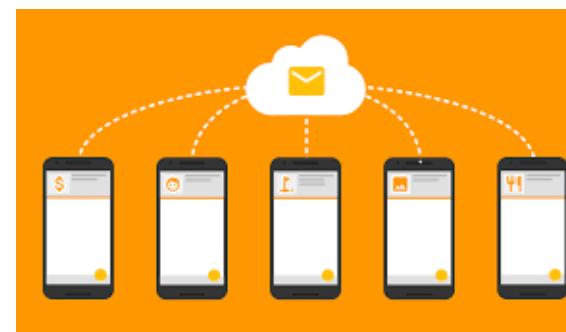
Rodeo Town Taxi Application



Messaging

Messaging

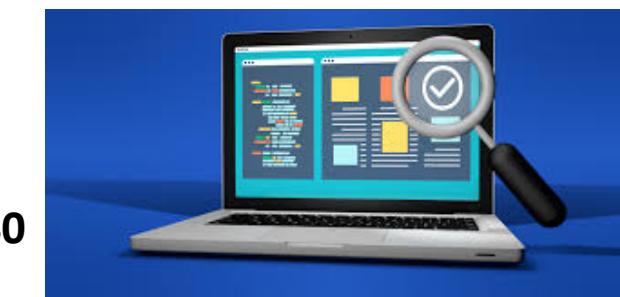
- One to one
 - Communication between drivers and customers
 - Through internet connection
- One to many
 - Announcement by the manager
 - Through firebase



Testing

Testing

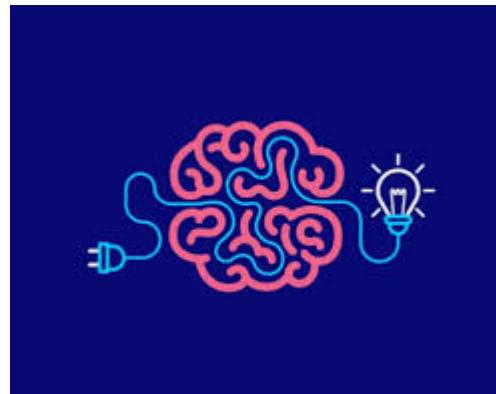
- Testing is important
- Overall testing:
 - Exceptions instead of crashing
 - Observe the driver & customer's side in a practice test
- Detailed testing:
 - Ensure map & routing algorithm works
 - Unit testing of the fragile parts



Conclusion

Conclusion

- Build a server with a database ✓
- React Native as code base ✓
- Connect with Google Maps API ✓
- Customer requests a taxi ✓
- Customer communicate with driver by phone / message
- Administrator controls the drivers
- Cross platform messages
- Testing



A problem has been detected and Windows has been shut down to prevent damage to your computer.

PROCESS_INITIALIZATION_FAILED

If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware is properly installed. If this is a new installation, ask for any Windows updates you might need.

If problems continue, disable or remove any recently installed hardware or software. Disable BIOS memory caching or shadowing. If you need to use Safe Mode to fix your computer, press F8 to select Safe Mode.

Technical information:

*** STOP: 0x00000060 (0xF2)

*** 4FQ.sys - Address

stop error screen,
s again, follow

Hardware is properly installed.
Contact hardware or software manufacturer

installed hardware
caching or shadowing.
components, restart
Options, and then

7,0x00000000)

00, Datestamp 4d5dd88c

Beginning dump of physical memory

Physical memory dump complete

Contact your system administrator or technical support for further assistance.



An Ellensburg taxi app that's
out of this world...



Tyler Lloyd
Junyu Lu
Austin Richardson
Jose Rodriguez
Stephen Stengel
Supervisor: Szilard Vajda
Client: Kenneth Swanson

Rodeo Town Taxi App

From RODEO Speedwagon