Team Leader:
Jeremy Galang (jjgalang@iastate.edu)

Team members:
Gabrielle Johnston
Jason Neville
Jorden Lee
Ethan McGill
Lorenzo Zenitsky

Team: sddec20-20
Team Email: sddec20-20@iastate.edu

PARKOUR

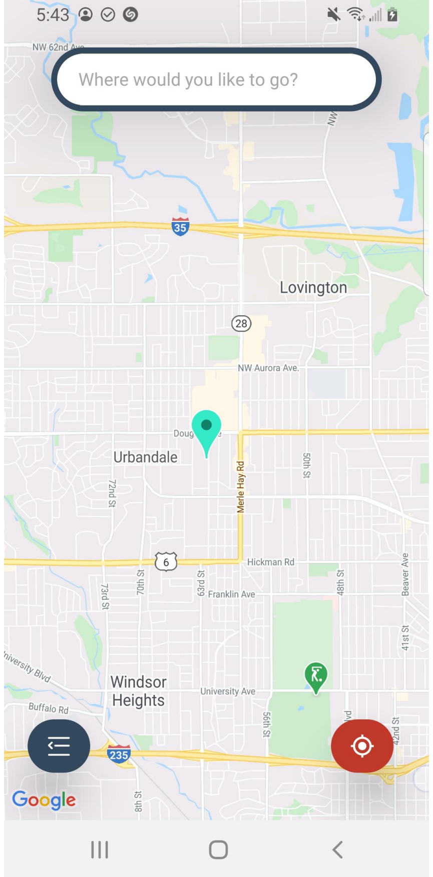
A PARKING-SHARING SYSTEM

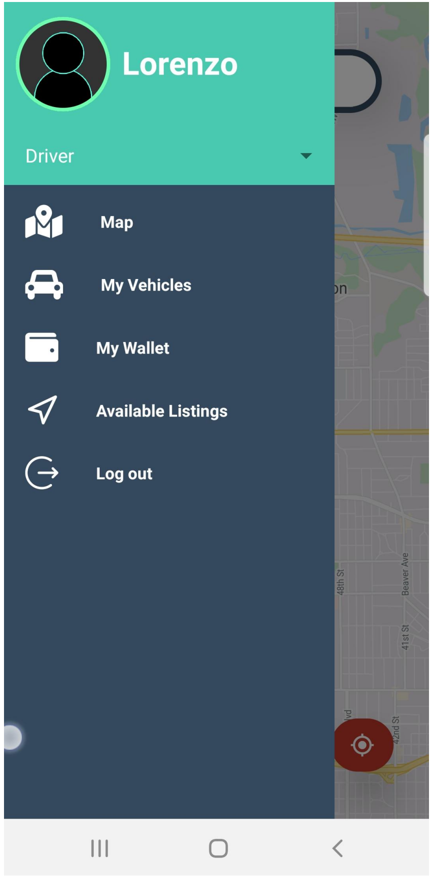
Problem Statement

- Parking is tough to find/expensive when big events take place.
- Example: Iowa State Fair
  - Fairground parking fills up
  - Neighbors rent out spaces to guests
  - No form of organization, drivers mostly wander around looking for spaces
  - Causes traffic around the area
- Some people have extra spaces available for parking
- Can be difficult for drivers to find parking, and hosts to advertise and compete

Project Solution

- Our app makes it possible for users to rent their extra space on their property, making parking cheaper and more accessible to other guests.
- Available for iOS and Android users
- Similar to AirBnB, except for parking.
- Crowdsourced by users/groups
    - Users can host and advertise their available parking
    - Guests can easily find and reserve parking





Requirements

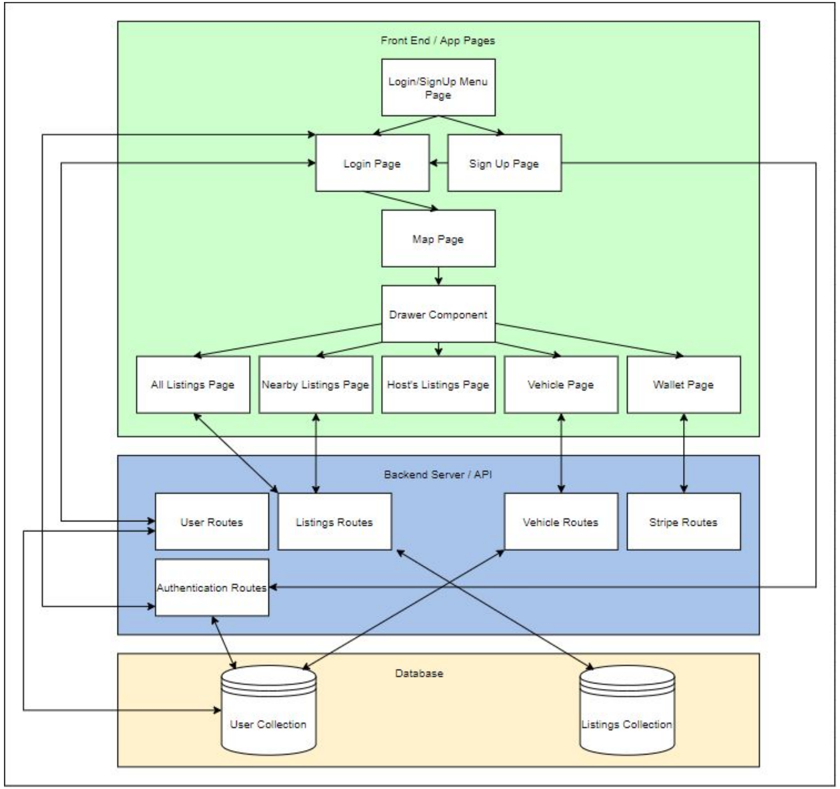
Functional:

- List and/or reserve parking spot(s)
- Individual profile view
- Secure wallet with payment information
- Save user’s vehicle information

Nonfunctional:

- Less than 3 second loading screen
- Easy to navigate for new users
- Render in different size phones
- Encrypt all sensitive information

Technical Overview



Technical Details

- We will be using the MERN stack
  - MongoDB, Express, React, Node.js
- MongoDB is a data-oriented language that stores data in JSON-like files
- Express is a JavaScript server framework
- React (Native) is a used as the frontend framework
- NodeJS allows for JavaScript to be executed as a server
- The Stripe API will be used for processing payments between users
- Two teams: Frontend + Backend

Frontend

- Other technologies/utilities we used outside of the MERN stack
  - Axios
    - Used for API communications (GET, POST, DELETE, PUT)
    - Communicates with the database
  - Async Storage
    - Enables a feature for auto-login
  - Emulators
    - XCode/Android Studio
  - GooglePlaces API
    - Search and query specific locations
  - Redux
    - State management

android

React Redux

Official React bindings for Redux

Frontend Challenges

- Learning new frameworks and languages:
  - Java/TypeScript, Node.js, React Native, React Native Google Place Autocomplete
- Setting up React Native environment
- Stripe Elements vs React Native Components

Backend

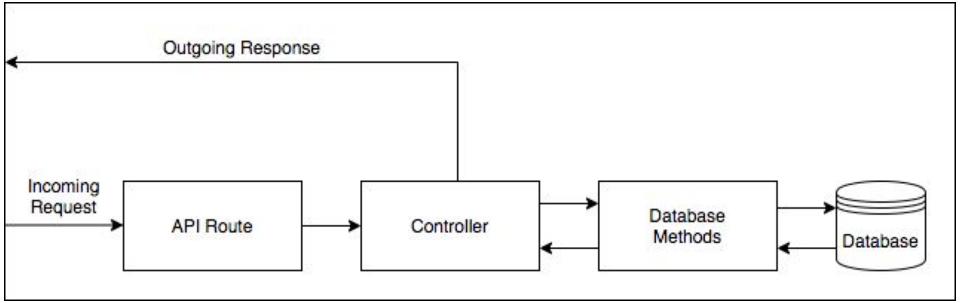
- Node.js for environment
- Used Mongoose.js library to interface with MongoDB
- Express.js library handled routing and request/response objects from HTTP requests
- Used session ids to handle user sessions and request authentication
- Modular server design where each module handles own task
- Hashes and salts sensitive user information before storing in DB (bcrypt)

Backend Challenges

Automated Testing

Backend architecture interfered with Jest Testing framework

- No issues with testing Database Methods
- Mocking response and request data in Controller methods did not work
  - Responses with error code statuses worked as expected
  - Responses with OK code status (200) did not work as expected
  - Our data collections got wiped twice trying to test controller functions



Administrative Challenges

- Covid
  - Virtual meetings
  - No scalability testing
- Didn’t get application up on the app store
- Admin features

Lessons Learned

- How to adapt to unexpected circumstances
- Working together to pool knowledge
- How to learn new technologies
- How to implement new technologies into a program
- How to debug personal environments

Future Development Opportunities

- Fix Bugs through more extensive testing
- Provide routing on Maps once user reserves listing
- Chat feature to allow communication between host and guest
- View/Interact with previous listings
- Rate hosts and area around parking spot
- Add features dependent upon user feedback through mass testing

Demo