Michael Small

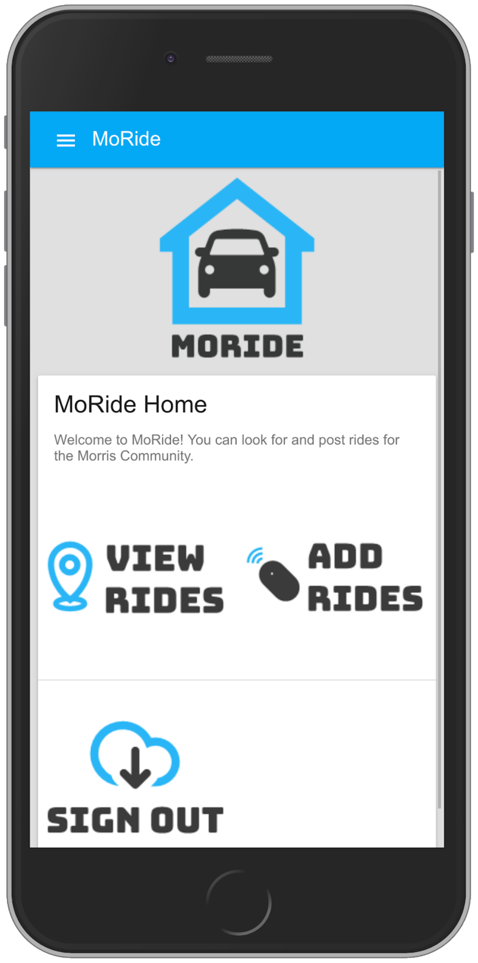
MoRide - Morris Ride Sharing

Overview

MoRide is a ride sharing web application for the the town of Morris, Minnesota. Commissioned by the student government of the University of Minnesota Morris, MoRide fills a niche in the Morris community while promoting eco friendly transportation.

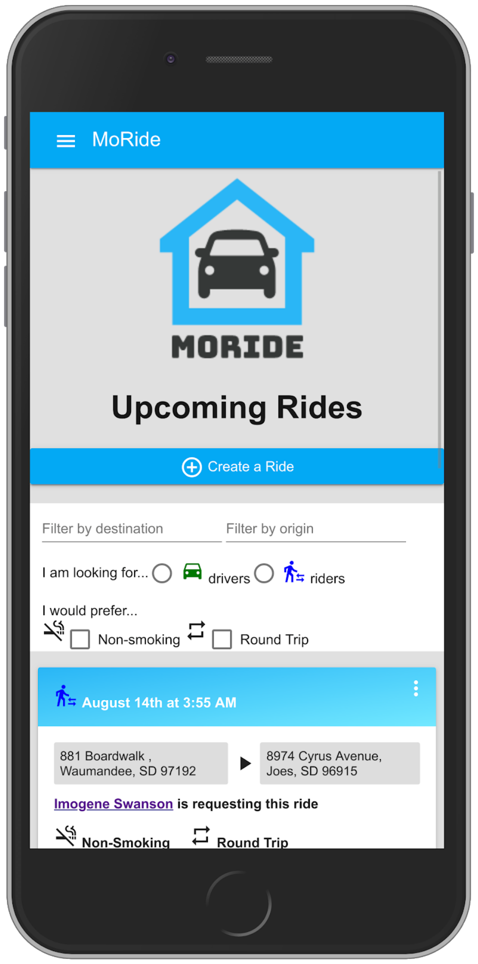
To use MoRide a user must sign in with a Google account. All University of Minnesota students and staff have a university issued Google account, but any Morris resident with a Google account can sign in and use the application as well. Once a user is signed in, they are free to use any of MoRide's functionality.

Any user can request a ride that they wish to receive from whoever in the Morris community could meet the requester's ride criteria. Any user can also post a ride that they are are willing to offer.



MoRide Home

Welcome to MoRide! You can look for and post rides for the Morris Community.

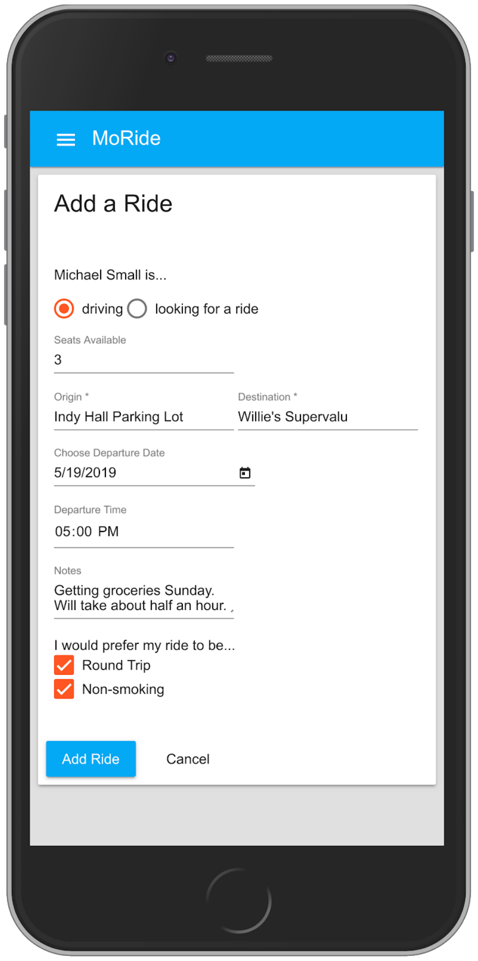


August 14th at 3:55 AM

881 Boardwalk , Waumandee, SD 97192 ▶ 8974 Cyrus Avenue, Joes, SD 96915

Imogene Swanson is requesting this ride

Non-Smoking Round Trip



Michael Small is...

[✓ driving] looking for a ride

Seats Available: 3

Origin *: Indy Hall Parking Lot

Destination *: Willie's Supervalu

Choose Departure Date: 5/19/2019

Departure Time: 05:00 PM

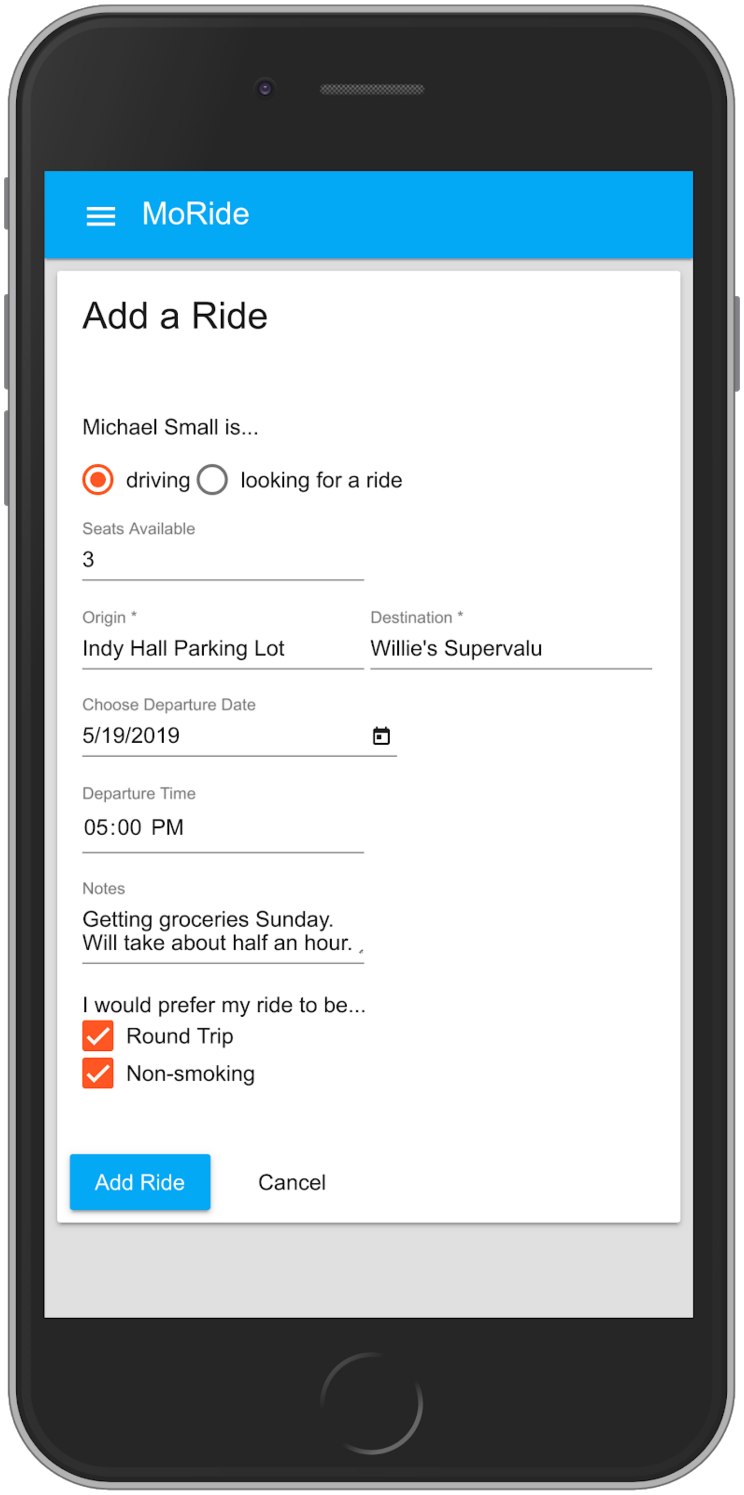
Notes: Getting groceries Sunday. Will take about half an hour.

I would prefer my ride to be...

[✓] Round Trip

[✓] Non-smoking

Michael Small



MoRide

Add a Ride

Michael Small is...

[✓ driving] looking for a ride

Seats Available
3

Origin *
Indy Hall Parking Lot

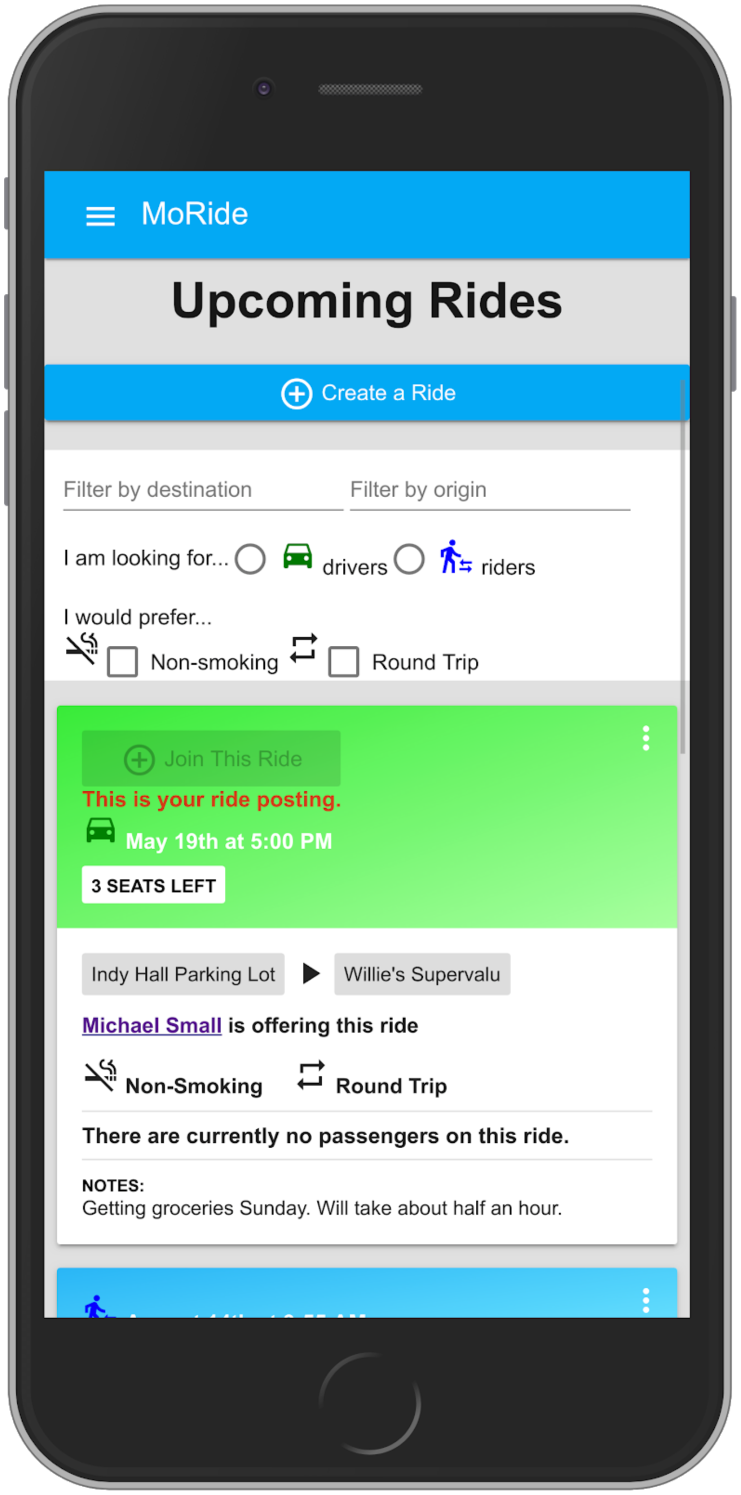
Destination *
Willie's Supervalu

Choose Departure Date
5/19/2019

Departure Time
05:00 PM

Notes
Getting groceries Sunday.
Will take about half an hour.

I would prefer my ride to be...
[✓] Round Trip
[✓] Non-smoking



MoRide

Upcoming Rides

Filter by destination

Filter by origin

I am looking for... drivers riders

I would prefer...
Non-smoking
Round Trip

This is your ride posting.

May 19th at 5:00 PM

3 SEATS LEFT

Indy Hall Parking Lot ▶ Willie's Supervalu

Michael Small is offering this ride

Non-Smoking Round Trip

There are currently no passengers on this ride.

NOTES:
Getting groceries Sunday. Will take about half an hour.

[Left] Filling in details for a ride a user could offer.
[Right] What an offered ride looks like on the "Upcoming Rides" page.

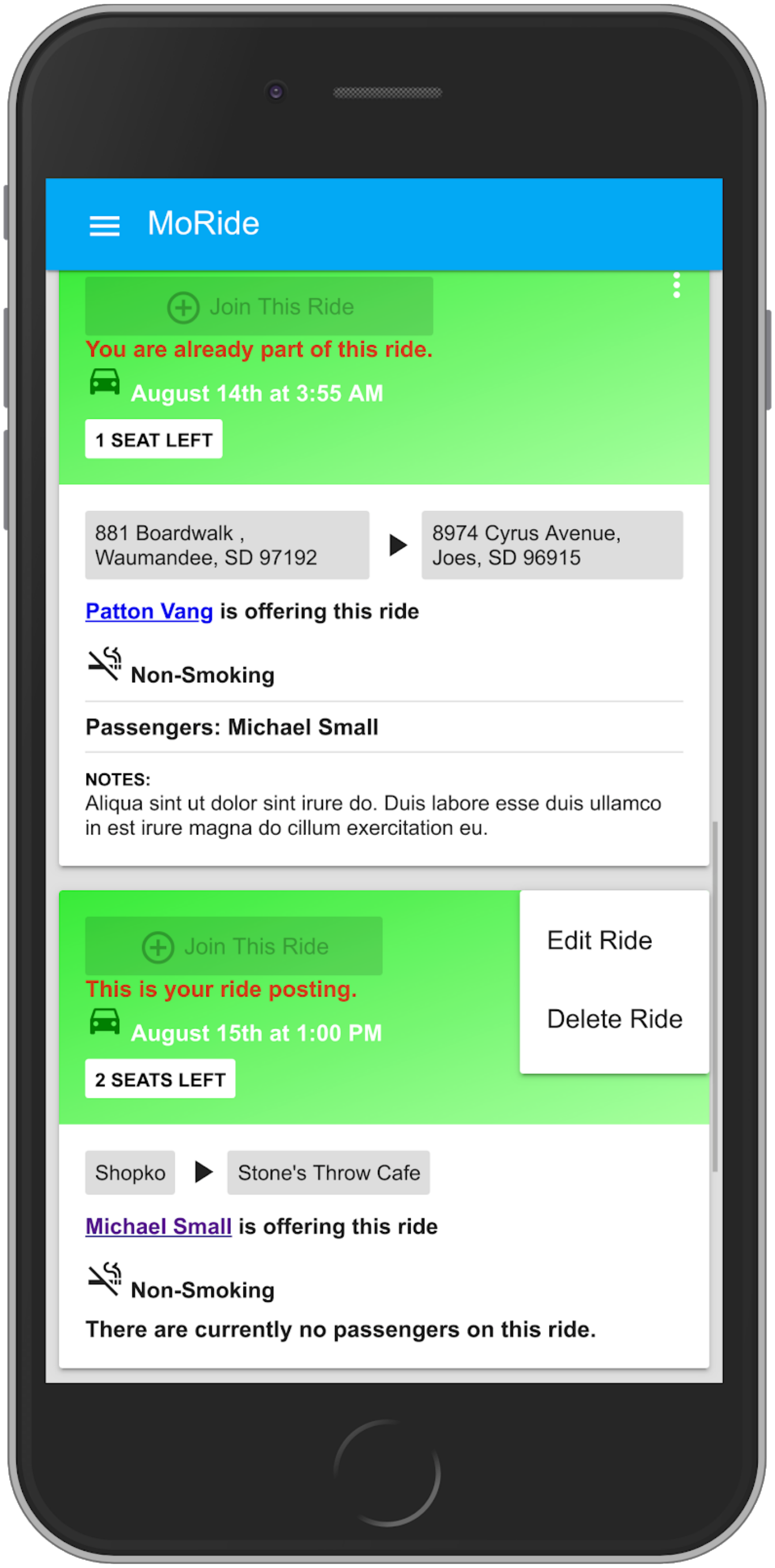
Michael Small

On the “Add a Ride” page, both offered rides and requested rides can specify ride Origin, Destination, Departure Date, Departure Time, and Notes. The postings can also be tagged with tags such as “round trip” or “non-smoking”. The Seats Available field is only shown if the user marks themselves as a driver. If a user is offering a ride, Seats Available is a required field. Origin, Destination, and designation of whether a ride is offered or requested are required fields for both types of rides while the rest of the fields are optional.

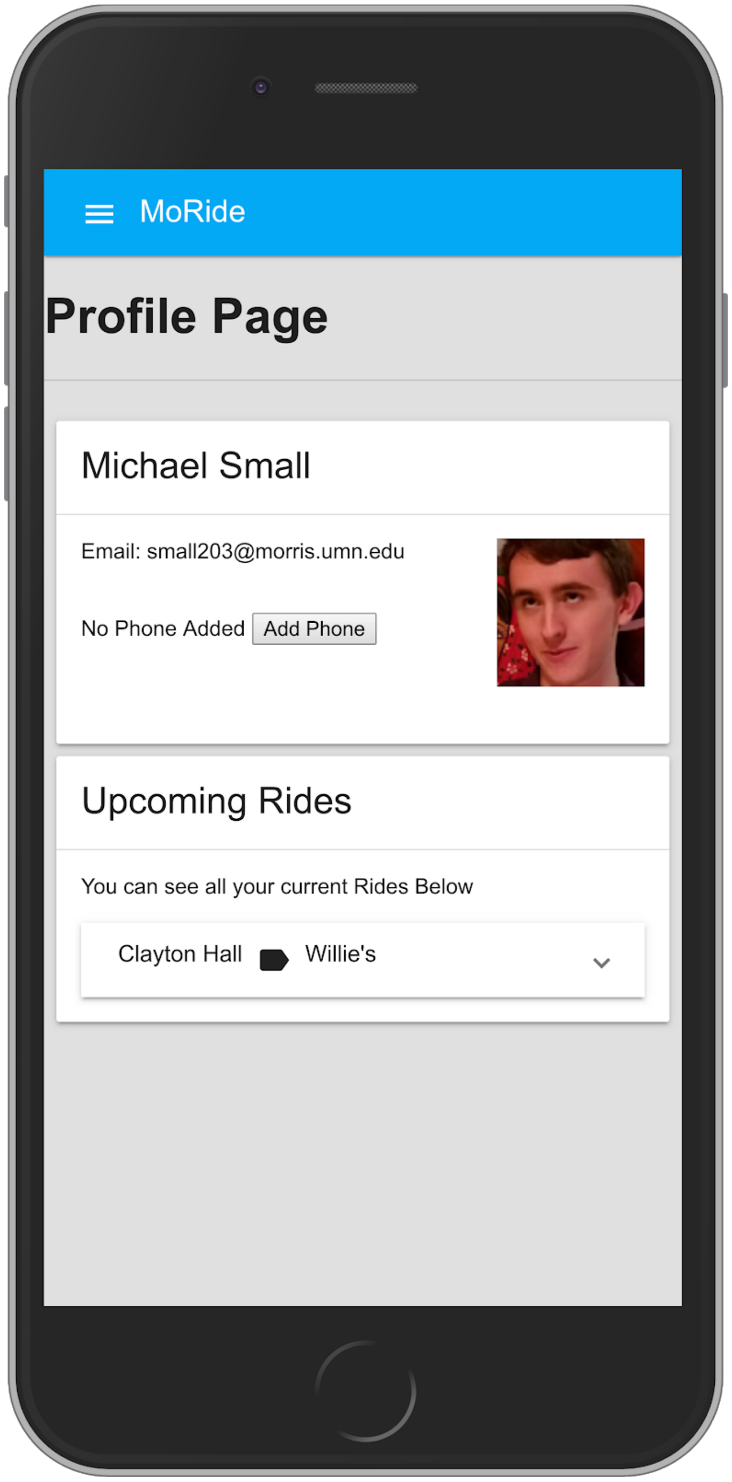
Users can search for and filter ride requests or offers on the “Upcoming Rides” page. Users can filter by Destination, Origin, tags and either type of ride.

If a ride request or offer was created by a user, they have the option to edit details of the ride or delete the ride from a popup.

Should a user want to join a ride that was offered by somebody else, they can request to join that ride using the “Join This Ride” button. Joining the ride lists that user as a passenger and reduces the seats available for that ride.


MoRide
Join This Ride
You are already part of this ride.
August 14th at 3:55 AM
1 SEAT LEFT
881 Boardwalk , Waumandee, SD 97192
8974 Cyrus Avenue, Joes, SD 96915
Patton Vang is offering this ride
Non-Smoking
Passengers: Michael Small
NOTES:
Aliqua sint ut dolor sint irure do. Duis labore esse duis ullamco in est irure magna do cillum exercitation eu.
Join This Ride
This is your ride posting.
August 15th at 1:00 PM
2 SEATS LEFT
Edit Ride
Delete Ride
Shopko
Stone's Throw Cafe
Michael Small is offering this ride
Non-Smoking
There are currently no passengers on this ride.

Michael Small


MoRide

Profile Page

Michael Small

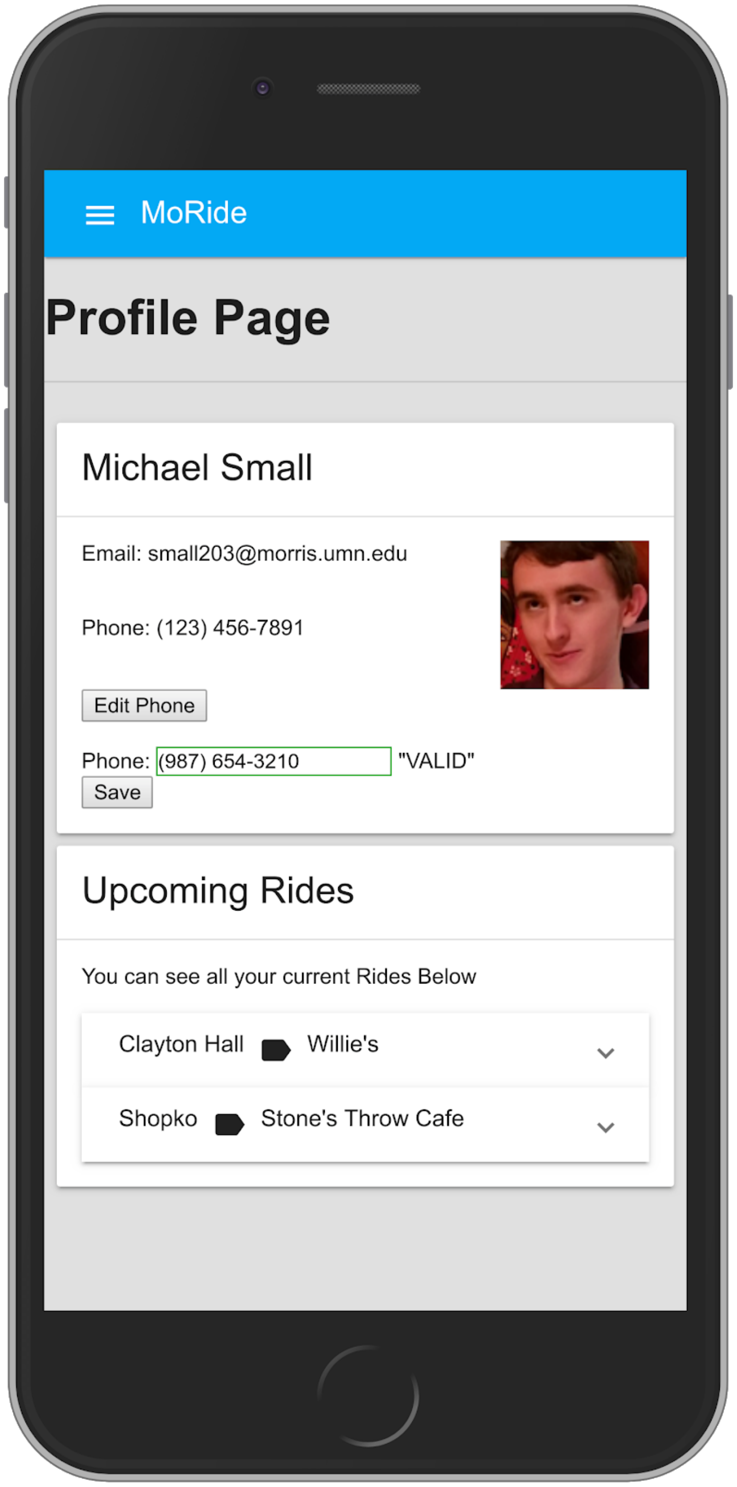
Email: small203@morris.umn.edu

No Phone Added

Upcoming Rides

You can see all your current Rides Below

Clayton Hall Willie's


MoRide

Profile Page

Michael Small

Email: small203@morris.umn.edu

Phone: (123) 456-7891

Phone: (987) 654-3210 "VALID"

Upcoming Rides

You can see all your current Rides Below

Clayton Hall Willie's

Shopko Stone's Throw Cafe

Each user has a profile page containing their name, Google email address, Google profile picture, and a compact list of details about the rides that they created. Users can choose to add a phone number for additional contact purposes.

The end goal of MoRide is to promote ride sharing for sustainable transportation.

Technologies Used

1. Angular 5+ (frontend)
2. Spark Java (backend)
3. MongoDB (database)
4. Angular Material Design (styling)
5. Karma/Jasmine (unit testing)
6. Protractor (end-to-end testing)
7. Digital Ocean (deployment)

Agile Driven

Our Software Design course is driven by an Agile approach. Each semester, customers from our community approach the course instructors and decide on what that semester's group of students will create. The Morris Campus Student Association and prostaff in Student Life and The Office of Sustainability teamed together as our customers.

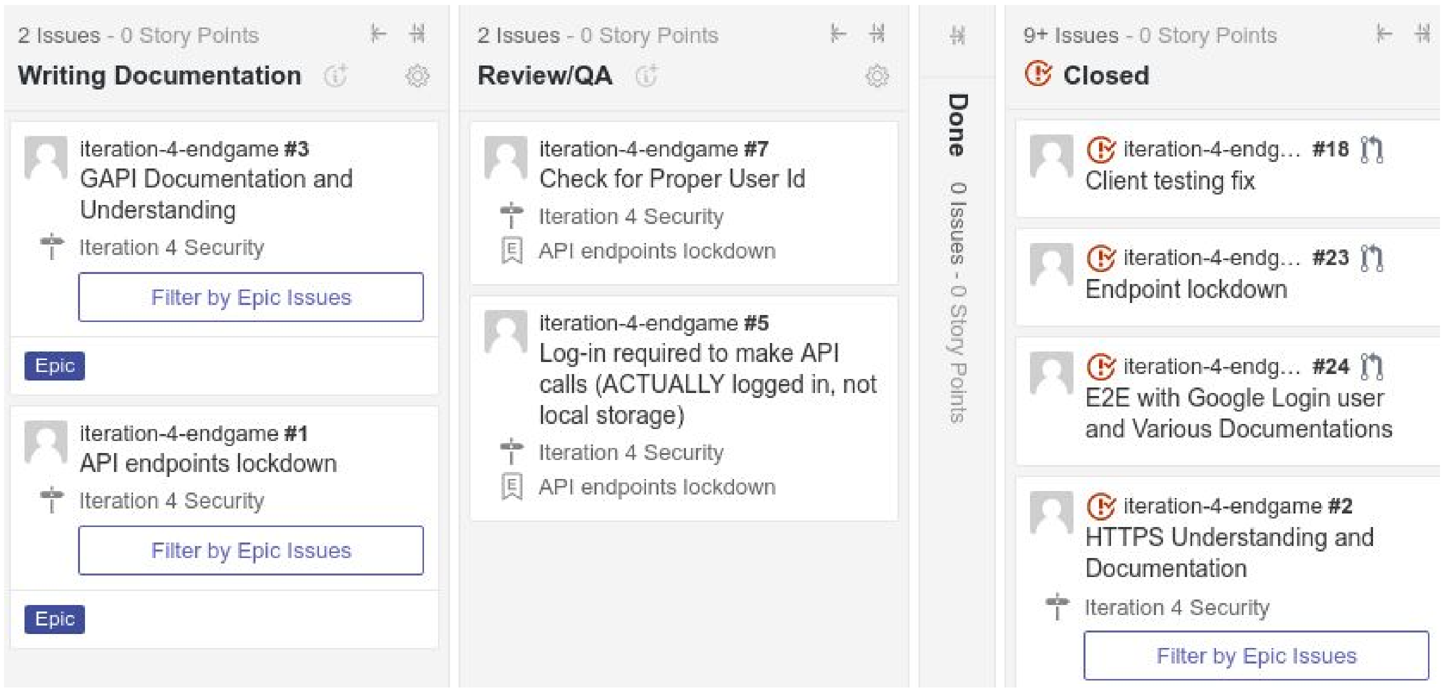
Our developer team and the customers got together at the beginning and brainstormed our expectations for the project. Once we all had a common understanding of the big picture, our developer team broke into small groups that each decided what potential features (user stories) could be offered and sold to the customers.

Each team worked in one-to-two week iterations on whatever stories customers bought from them. Each team used ZenHub as a virtual visual workspace to manage our progress on user story delivery. At the end of each iteration, each group presented a showcase of their completed stories to the customers and received feedback. From there, larger groups were formed and they combined the best stories from their code base together. They sold new stories based on the collective code base and then repeated the same iteration process.

Each team member along the way was responsible for understanding the full stack as well as writing tests (client, server, end-to-end) to ensure the quality of our final product.

We would also have developer-group-to-instructor and individual-student-to-instructor meetings in the middle of and after iterations. These meetings kept the developers on track with their expectations and helped guide them on what to start, stop, and continue doing to ensure a smooth process of user story delivery with an Agile framework.

Michael Small


2 Issues - 0 Story Points

Writing Documentation

iteration-4-endgame **#3**
GAPI Documentation and Understanding
Iteration 4 Security
Filter by Epic Issues
Epic

iteration-4-endgame **#1**
API endpoints lockdown
Iteration 4 Security
Filter by Epic Issues
Epic

2 Issues - 0 Story Points

Review/QA

iteration-4-endgame **#7**
Check for Proper User Id
Iteration 4 Security
API endpoints lockdown

iteration-4-endgame **#5**
Log-in required to make API calls (ACTUALLY logged in, not local storage)
Iteration 4 Security
API endpoints lockdown

Done 0 Issues - 0 Story Points

9+ Issues - 0 Story Points

Closed

iteration-4-endg… **#18**
Client testing fix

iteration-4-endg… **#23**
Endpoint lockdown

iteration-4-endg… **#24**
E2E with Google Login user and Various Documentations

iteration-4-endgame **#2**
HTTPS Understanding and Documentation
Iteration 4 Security
Filter by Epic Issues

A select portion of just three progress pipelines in Iteration 4's ZenHub page. We encouraged reviewing pull requests and documentation before marking issues as completed.

Personal Contributions

MoRide Codebase (Iteration 1-3)

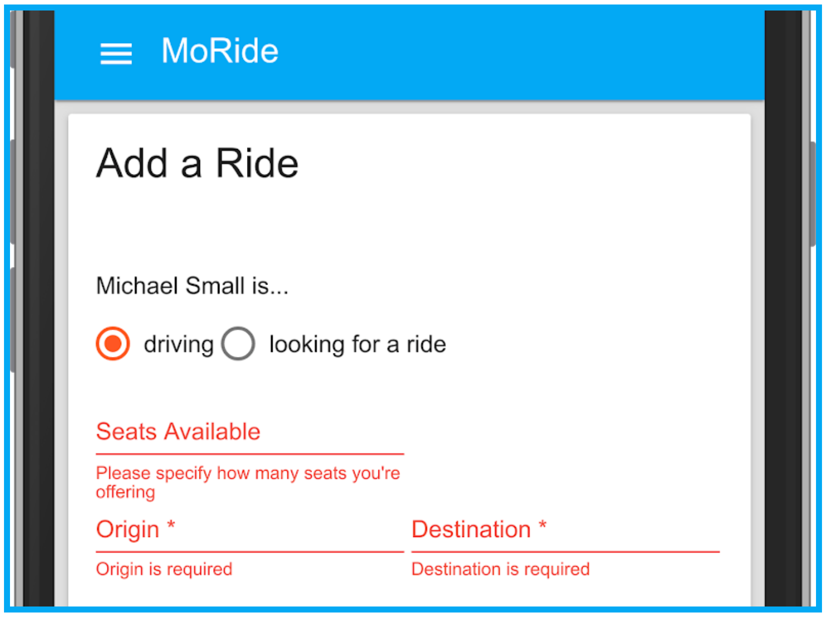
[GitHub repository for Iteration 3]()

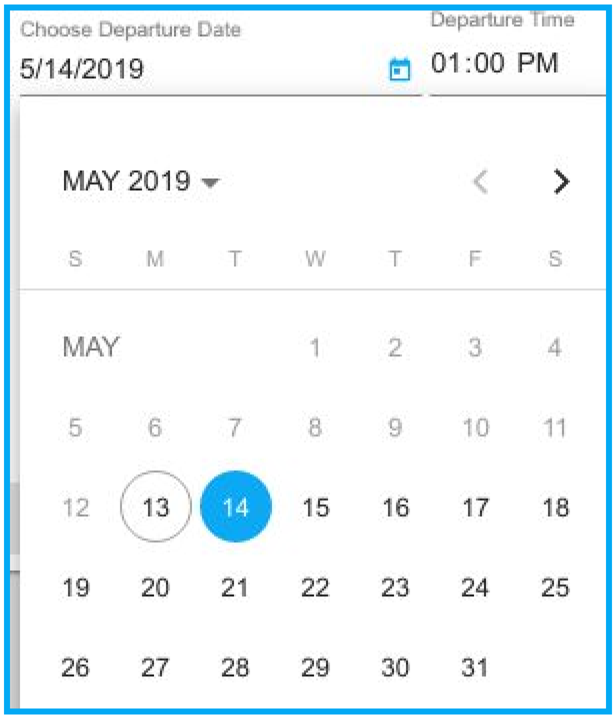
In total, four iterations occured that semester. Iteration 3 was the last iteration that I wrote new features for and worked with our traditional customer group as I was a part of a special team for the last iteration (see next section).

While I didn't create new features for the app in the last iteration, I worked on the same codebase from Iteration 1 through 3. My team's Iteration 3 code base ended up being used as the starting point by the final two normal groups that developed new features for the customers in Iteration 4.

I do not know why each of the final groups used the Iteration 3 code base that I was a part of from its inception, but I believe that it was because of the solid range of tested features that my teams and I saw to fruition over the semester. I had a hand in many of the features in the Iteration 3 code base to varying extents, with hands on experience in our entire stack.

I contributed a great part to the “Add a Ride” page. I saw that essential fields were required to be added on the form and gave meaningful error dialogs to users if they were not completed. I also saw that users could only add meaningful data to those forms, as in a user must type “3” into the Seats Available field instead of something like “i am offering 3 seats.” I also implemented how that data was passed to the backend and stored in the database.





I also saw to it that users could enter the date and time of a ride in a meaningful way. Date is entered in standard American month/day/year. A user can select the date of departure from a calendar popup. A user cannot select a date before the current day. As for time, the time form dictates what the user can put in as standard American twelve hour AM/PM format. While the user sees the data in this familiar format, the work I did in the backend and databases stores the date in ISO-8601 format and time in twenty-four hour format. This allows the data to be easily manipulated in the server while giving users meaningful ride data.

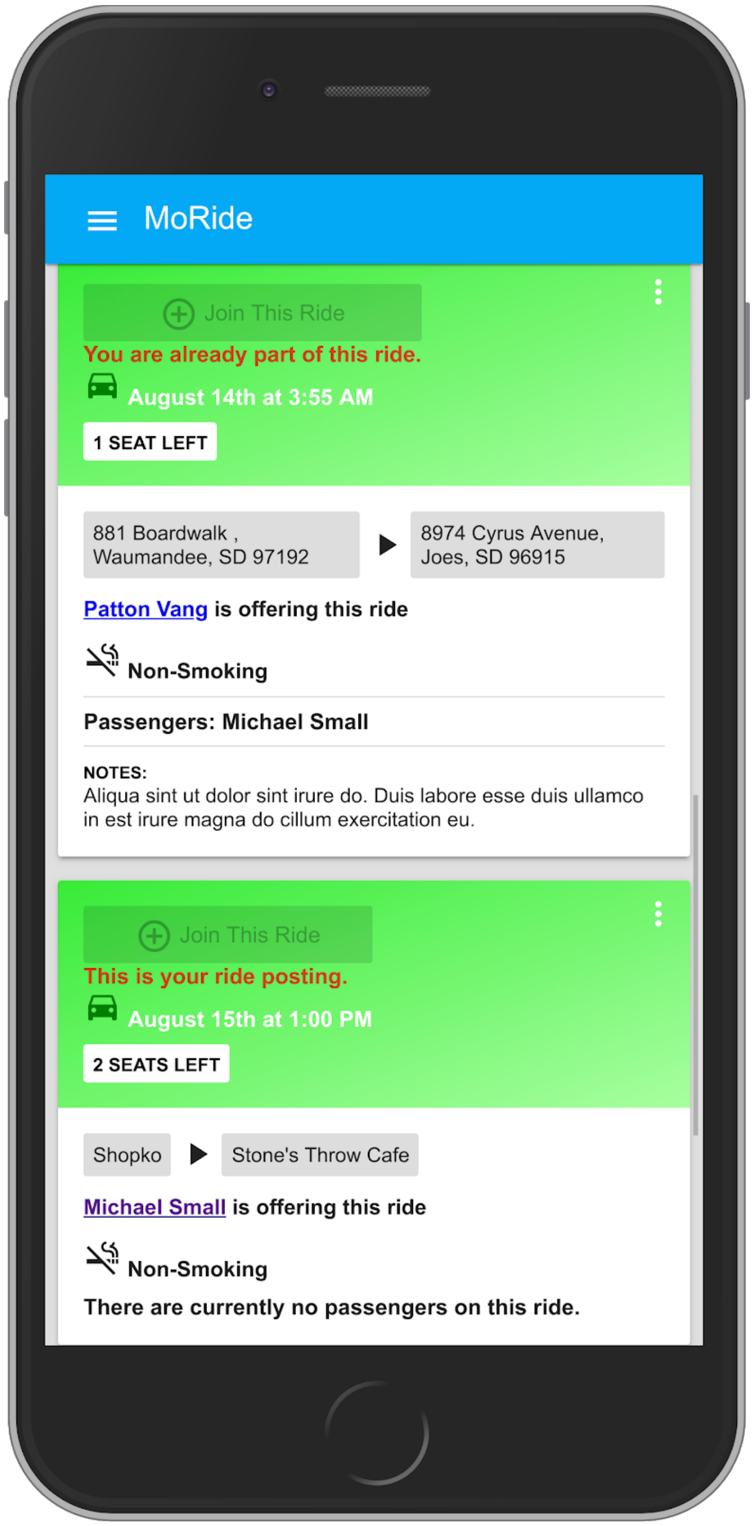
Michael Small

I also played a part in the "Available Rides" page. I helped style and format the ride listing cards.

Using the date and time data in the backend using its logic friendly format that displays in a meaningful format to the user, I ensured that the rides were listed in order of when they were offered. Rides are listed in chronological order from soonest on top to latest on bottom. An August 15th ride comes after an August 14th one, and if they were to be on the same day the one with the earlier time would be on top.

Rides that have a date and time that are past the current date and time are automatically removed from the "Available Rides" page. As per customer request, old rides are not deleted and just simply not shown any longer.

All of these features were tested in various forms. The functions that handled time and date parsing to user readable format were developed by creating a wide range of unit tests written using Jasmine and Karma. End to end testing using Protractor ensured that added rides that are created show up on the page and in the right order by time and date.


MoRide
Join This Ride
You are already part of this ride.
August 14th at 3:55 AM
1 SEAT LEFT
881 Boardwalk , Waumandee, SD 97192
8974 Cyrus Avenue, Joes, SD 96915
Patton Vang is offering this ride
Non-Smoking
Passengers: Michael Small
NOTES:
Aliqua sint ut dolor sint irure do. Duis labore esse duis ullamco in est irure magna do cillum exercitation eu.
Join This Ride
This is your ride posting.
August 15th at 1:00 PM
2 SEATS LEFT
Shopko
Stone's Throw Cafe
Michael Small is offering this ride
Non-Smoking
There are currently no passengers on this ride.

I contributed many other features throughout the three iterations, but those were my most involved ones. As for bug fixing, by Iteration 3 I had fixed a well known but elusive bug that prevented a user from seeing a newly created ride or edited ride until they refreshed their page.

Michael Small

Security (Iteration 4)

GitHub repository for Iteration 4 (security)

Google OAuth had been used in past and present software design course offerings and will likely be used in the future. By the last iteration this semester, every group was using Google OAuth in a way that was essential to use their web apps.

Most Google OAuth work was based on a previous classes' implementation. However, a critical flaw was that their implementation relied on using Local Storage to hold details like the user's name, email, picture, id and so forth. Local Storage can be viewed and manipulated very easily by any logged in user and allowed for rides and users to be spoofed and for sensitive information to be out in the open. Additionally, our API was exposed and didn't require any sort of user token from Google to be called; POST requests could be called by anybody to add, edit or delete ride information from anybody. This was the status quo for multiple years of software design until our instructors tasked a small team on the last iteration to fix the problem. The group I was a part of got rid of Local Storage and made sure any POST requests on rides required a unique user identification token created by Google and effective for only that user's info.

What I had the most impact on in this iteration was client testing and end-to-end testing. Almost all of our client tests failed once my group members began implementing the token based request handling. What I did was help fix those tests by mocking the necessary methods and fields related to authentication in all of our unit tests. I wrote the documentation on how to do so for future groups.

As for end-to-end testing, I did necessary research on how our new token based solution could be tested in the first place. End-to-end tests worked fine using the old Local Storage solution because "isSignedIn" could be flagged in Local Storage as true, but were seemingly impossible using tokens. Nobody could figure out how to sign in a test user or bypass sign in using the token solution. Our instructors who valued end-to-end tests insisted that it was bad practice and against Google's terms of services to have tests log in a test user using a test user's credentials. After gathering research on the issue, I presented to my group and instructors that it is valid and actually encouraged by Google's own OAuth team to use a test user account that the tests use to enter credentials for each test suite. The problem and solution is documented here.

Additionally, I helped the group secure the API by introducing them to Postman. Regarding deployment, I wrote docs on deploying Google OAuth on a Droplet.