Carpool

University of Illinois at Urbana Champaign

CS428 Software Engineering

Prof. Darko Marinov

Spring 2014

Rishi Bajekal (bajekal1)
Martin Wozniewicz (wozniew1)
Kevin Shu (kshu2)
Shil Bhattacharyya (sbhttch2)
Stephen Lee (slee117)
Won Jung Yang (wyang15)
Gary Li (gli24)

Table of Contents

Description

Motivation

Currently, Illinois students without cars are forced to buy expensive bus tickets if they want to return to Chicago. Carpooling is difficult to organize, and there is no existing solution to facilitate this. Our application will match student drivers with riders and allow both to benefit (driver makes money, rider pays less to get home). Our application would make carpooling much simpler for students.

As students, we have faced the issue of getting to Chicago numerous times. Bus companies have bad service, high prices, and rigid schedules. Carpooling is done via a collection of scattered Facebook pages and Craigslist ads.

Solution

We built a web application for students who want to carpool. Drivers can post rides, and riders can sign up for them as long as there are seats available. We tried to keep it as simple as possible, not requiring accounts or handling payments through our site. We use email verification to confirm actions by users in lieu of logging in, and let them handle payments on their own a la Craigslist.

Language and Platform Choice

We are using Node.js (JavaScript) as our primary back-end language, and AngularJS (also JavaScript) for the client side of the application. Node is a popular web programming platform with plenty of community and library support. We want to make the site responsive and interactive for the users.

Jasmine is a JavaScript testing framework that we can use for both front- and back-end unit testing. It's a popular choice among node users so we'll have plenty of community support as well.

Process

The method we followed for this project is the general framework of XP/Scrum. This method included pair programming, unit tests of all code avoiding programming of features until they were needed, code reviews, and simplicity and clarity in all code. Weekly meetings were held, in order to get our requirements for the iteration ready. The weekly meetings discussed the topics of where we were at in our project, what needed to be done for the next iteration, and how the work would be split. During this time, we would assign the pairs and the user stories that a pair would work on for the iteration. Since everyone was working with new technology, our weekly meetings also provided us a time to talk about struggles that we had in the previous week. We also used Trello to divide work every week. Trello allowed our group to divide and assign specific tasks for each user story easily. Also, Trello let everyone in the group know what the others were working on for the iteration. So, we could resolve any conflict and collaborate with each other.

After pairs had been decided for the iteration, the pairs would the break off and find time that they had available to work on it. When a pair finishes a user story, they would ask to have their code be reviewed. After it has been reviewed, the code would then be merged onto the main branch. Oftentimes, we would

have an extra group meeting to finalize the work that was done and to make sure that combining everyone's user stories did not break our application.

The issue with iterative development was dividing the user cases workload to meet the goal for the iterations. Some weeks, we planned well and we estimated the time it would take well, but other times it took us longer than we had originally planned. When this happened, everyone had to stay and work longer to meet the iteration requirement.

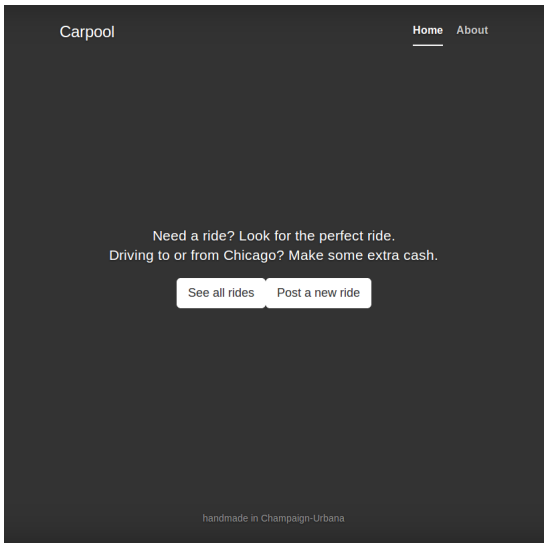
Collaborative development led our group to experience a few issues. Sometimes, a pair was assigned to do work that another pair had originally been working on. Oftentimes, each iteration pair had to switch from working on the front end to working on the back end. This means we had to review what the previous pair had done before we could begin working on our part. And sometimes it was hard to refactor or test new functions that were related to previous work done by another pair.

Requirements & Specifications

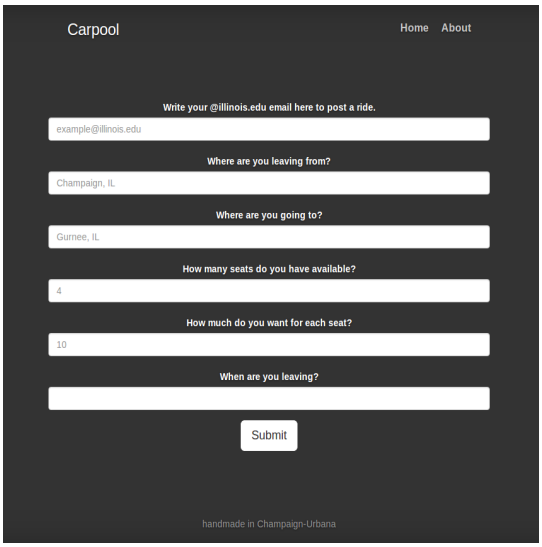
Carpool is an application that focuses entirely on the concept of ridesharing. With that in mind, all of the requirements and specifications have to do with only various activities that are involved with the ride or the people who are involved in the ride.

Ride Creation

With a ridesharing application, creating a ride is clearly the most important requirement as it makes for the core of the entire application. Upon accessing the application's home page, the users are greeted with two buttons that offer functionality to either create a ride or look for all current available rides. For ride creation, a user would click on the button that says "Post a new ride" to be brought to a page where they can enter in the information relevant to their ridesharing offer.



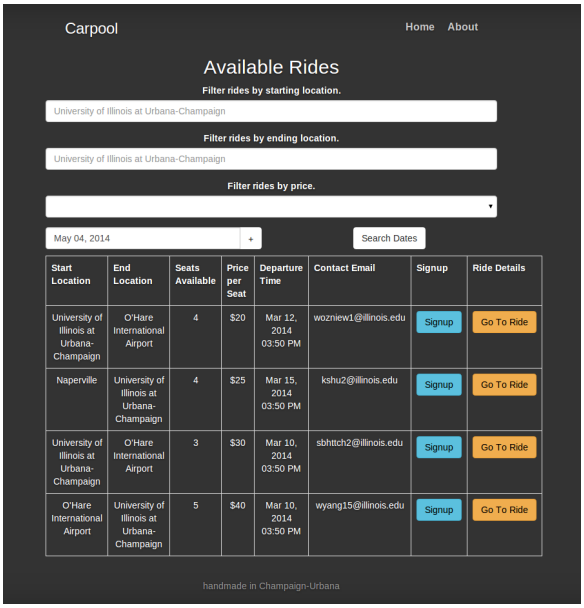
Carpool Homepage



Ride Creation Page

Ride Lookup

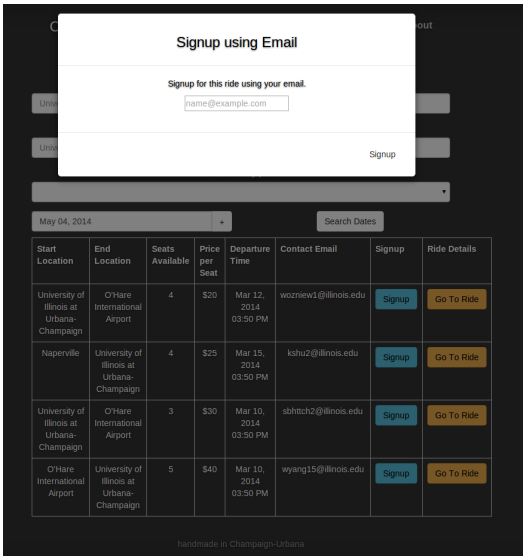
Reasonably, it follows that the next core requirement of a ridesharing application is to be able to look for rides and to be able to filter through rides that match your criteria. The ride lookup aspect of the Carpool application allows the users to filter through rides that users post by various fields such as the starting location, ending location, price, and date of departure. Because the nature of our Carpool application is currently to support only travel between Champaign-Urbana and the Chicagoland Suburbs, we decided that having another filter by date of arrival would be unnecessary.



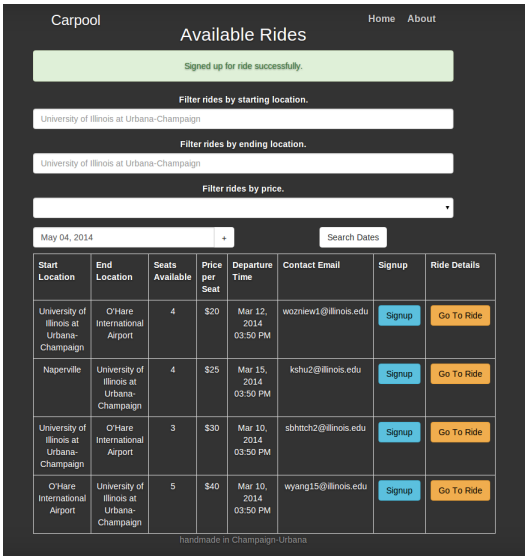
Ride Lookup Page

Ride Signup

After a user finds a ride that he or she is looking for, the next step would be to sign up for it. This is done by clicking on the “Signup” button that is present next to each ride result on the list of rides provided by the ride lookup. Clicking on the signup button brings up a modal window that accepts an email address and associates it with the ride. After entering a valid email, a modeless notification box will show up near the top of the screen that alerts the user to either a successful sign up or a failed signup (if there is a server error, for instance).



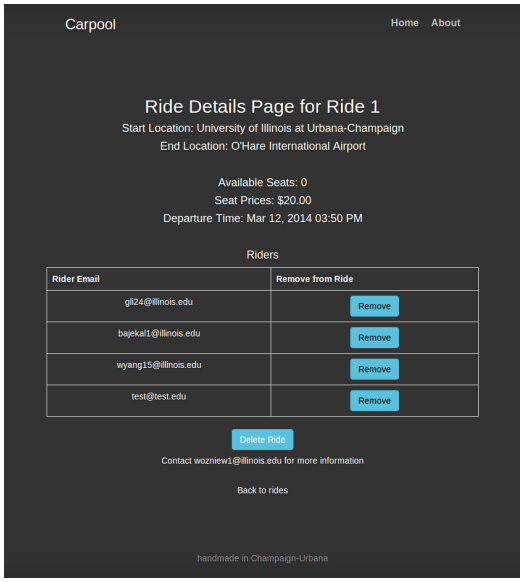
Ride Signup Modal Window



Ride Signup Success Modeless Notification

Ride Alteration

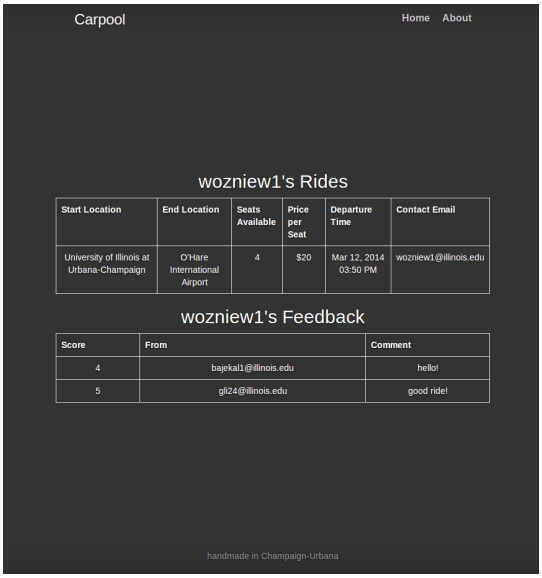
Decidedly, a ride will have to face changes sometimes as people change their minds regarding whether they want to partake in a ride anymore. To get around the issue of people who decide to no longer be part of a ride, there is an option to remove individual riders from a ride or to remove the ride in its entirety. These options are provided on the ride details page that is accessible from the ride lookup page.



Ride Details Page

Feedback and History

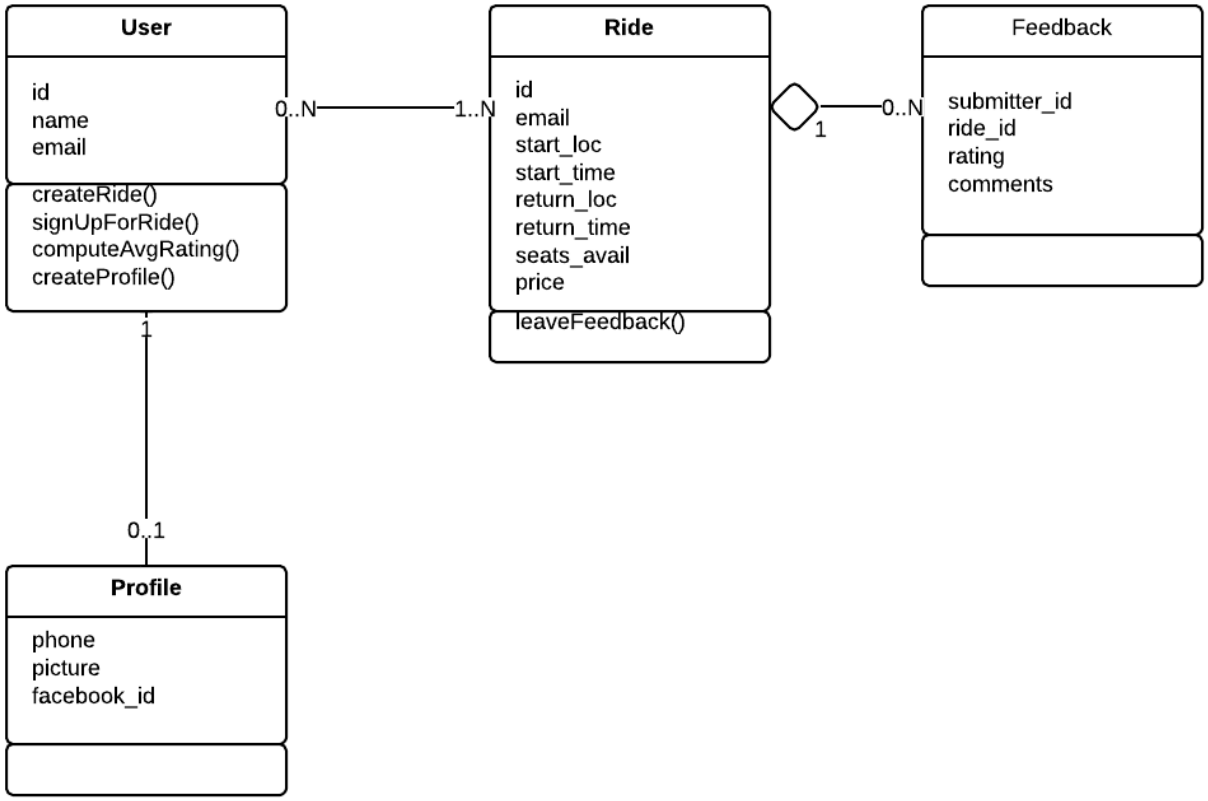
When signing up for rides, one of the most important things to know is the credibility, reliability, and safety of the person who will be offering the ride. Because this type of feedback is so important, each person who has partaken of a ride in the past will have a profile page that has various other users' feedback and a list of the rides that the person has offered in the past. From here, the idea is that the page will allow for other users to know about whether a particular individual is someone they would want to have in their car while ridesharing.



User Feedback Page

Architecture & Design

Our web application has a clear separation of technologies between the front-end, client side portion and the back-end, server side portion. The front-end serves as the portion of the application that a future user would use to interact with our application. Our core technology for designing the front-end was AngularJS and Twitter's Bootstrap. Bootstrap is a framework that provides all of the necessary components for designing the interfacing aspects of the web page. AngularJS utilizes JavaScript to expand on traditional web page design, which primarily uses static web pages, and allows for more dynamic uses. The back-end of our application used Node.js for functionality, and MySQL for storing data. Node.js is a web framework written in JavaScript, designed to handle all of the core server functionality. MySQL is a standard SQL database used to store information reliably.



UML Diagram of Application

The UML Diagram above shows the primary objects and methods of the application. The remainder of this section will provide an explanation of the main application functionality that this diagram depicts.

The core functionality of Carpool is to connect drivers and riders together. The primary means of connecting a driver to prospective riders is for the driver to post information about his or her ride. Riders then can view information about many driver's rides, and they can choose which ride is most appropriate for them.

Any prospective drivers can post their ride information through our web page. The web page asks drivers to fill out a form that includes important details, such as price of the ride as well as location from which the ride will start and stop. This information, as well as their contact email, is sent via an AJAX call to our backend server. The server stores the information into the database, and then sends a confirmation email back to the driver. This ensures that the driver's contact email is valid.

A rider can access a separate page on the website which lists all of the rides that drivers have posted. The page requests the information from the Node backend, which delivers the rides in the database. This information is sent to the AngularJS front end, which can modify the front-end web page to place the information and sort/filter the information according the the rider's needs. After the rider has selected the ride, their selection will be noted by the server, which stores in the database the information about the rider and his/her association with the ride. The driver will be emailed about the new rider for his/her ride by the server once the server stores the data into the MySQL database.

Each interaction by a user with our webpage which causes the user to send their contact information will store their information to the database. This allows the server to implicitly store information about each user, by assuming each user is associated with one email. Our database correlates the user with all of the rides they have created, been a part of, and any feedback they have received. When a user looks for information about another user, the server collects all of the information about the target user and sends it to the front end, where it will be formatted and displayed using AngularJS.

Future Plans

At the moment, our group does not have plans to continue working on the project. Should development continue, we have some future goals we would like to meet. Even though the core functionality of the application is operational, there are certain aspects of Carpool which still need to be addressed.

1. Using a mapping tool or API, such as Google Maps, could aid any drivers or riders about their location. As a visual tool and a means of determining distance, implementing such a tool into our web application could simplify the user experience.
2. The front end design of our application could be designed to better communicate the information to the client. The group's primary goal throughout development was to create a functional application. This led to some negligence on the design elements of the front end. This portion of the application can be reworked to better highlight the tools we implemented.
3. The application still is not being hosted online. The group has looked into free hosting possibilities, but the primary host we looked at, Heroku, had costs associated with implementing the MySQL database. Hosting the site would be a necessary step in introducing the application to the future customer base.

Apart from these goals, we could look into adding more functionality to the application, as well as potentially looking at advertising the application. A large initial user base would be almost required for this application to be useful to many users, so advertising could be considered a major long-term goal. This could require us to partner with other groups or find information to meet this end.

Personal Reflections from Members

Rishi Bajekal

Working on this application was a great opportunity to learn new technology and meet some great people. I have developed many web applications in the past but had not used the particular stack we utilized. In fact, my use of JavaScript was limited to front-end work and did not consist of nearly as much functionality as our project. As a result, I was able to learn a lot about asynchronous programming, callbacks, and other features of the language through this project. Additionally, I now feel quite comfortable using both NodeJS and AngularJS. NodeJS, along with the Express web framework, provided a great stack for quickly building web API's. I loved using AngularJS as a JavaScript front-end framework so much so that I ended up using it for another project for a different class as a result. I feel I will use both these technologies in the future. In addition to technological benefits, I also met some great people through working on this project. I enjoyed working with them over the semester and also became friends in the process.

Shil Bhattacharyya

This project has certainly been an interesting and worthwhile experience for me. Prior to Carpool, I had no experience with web development nor with the technologies with which we worked. Specifically, this semester was my first exposure to working with JavaScript and the frameworks of AngularJS and NodeJS. I found learning how to properly utilize these technologies to be quite challenging, but once I became more familiar with them, I grew an appreciation for the versatility and benefits they had to offer.

In regards to working with the group, I was very pleased with the manner with which my group members communicated with each other and completed their assignments. Everyone behaved politely and respectfully and we had no problem jelling. Moreover, the group was very proactive in helping each other whenever possible so the project moved along smoothly. Working on Carpool has been a positive and informative experience for me and I'm glad I was able to be a part of this project.

Stephen Lee

I learned a lot of new technologies working on Carpool. Most of the technologies used for this project were things I have never used before, such as AngularJS and Node.Js. Furthermore, my javascript skills were not very strong before working on this project. Now that I have worked on Carpool for the duration of this semester, I feel very comfortable using these new technologies in future endeavors.

The group was also easy to work with. Everybody seemed kind enough to help anybody that had problems. Having worked with the majority of the group for most of the code, each person has helped me whenever I had some problem with implementation. There were never any conflicts of personalities either; everything was positive. The group attitude certainly helped me to work comfortably for this project.

Gary Li

I really enjoyed working on Carpool. I was able to learn a lot of new things that I probably would not have learned if not for this project. It really opened my eyes to some of the technologies out there. AngularJS and NodeJS are both things that I learned to use from this project. While learning them was pretty difficult because everything was so new, I had a great time learning them because everyone in the group was so helpful. They definitely made learning and using these technologies very fun and easy for me. Also, everybody was very approachable and was very helpful whenever I was stuck on a problem.

Kevin Shu

Prior to working on the Carpool application, I only had a little bit of experience working primarily with JavaScript technologies like Angular and Node. After having worked more with these technologies to do this project, I think that I have garnered enough experience with these tools to be able to be comfortable working with them on my own now and in the future.

Furthermore, working with the group I had turned out to be quite fun as, typically, everyone was on the same page of wanting to get the work completed while also trying to have a good time doing so. The mood was generally quite light, but the group nevertheless always managed to take things seriously when it needed to. It certainly helped to have a group of fairly like-minded individuals as this contributed to the overall jelling of the team.

Martin Wozniewicz

I've never had an opportunity to work with a large team of peers on a project until now, and definitely learned a lot doing it. In the past, the largest team I've been involved with was only four others directly working on the code. Sharing the workload of a brand-new project with this many people brought its own challenges, but was very rewarding. While I'm not new to web programming, I still learned a lot about unit testing in Node, and client-side programming with Angular. Learning these new technologies was a worthwhile investment, and I'll definitely be using them in my projects from now on. As far as our process goes, I liked that we gradually settled into a rhythm that worked for us. It's impossible to predict exactly how a given group of people will work together, but one of the best things about our team is that we adapted to each others' needs.

Won Jung Yang

Working on Carpool was challenging because of all the new technologies that were used for this project. I had never used JavaScript, Angular JS and Node.Js before. So, learning and getting used to these new technologies was hard. However, all the members in the group were willing to help whenever I had questions or needed help, so the overall process was very positive and productive for me. The group was very organized and diligent in getting things done. And, everyone was very pleasant to work with. I definitely enjoyed working on Carpool. It was very positive experience.