Team GVBandwagon

**(Uberish)**

Final Report

Prepared by:  
Ryan Korteway

Nicolas Heady

Michael Christensen

\*Sprint 4 code in “RyanAndNicks” branch

\*Demo video links in “Master” branch

**Abstract**

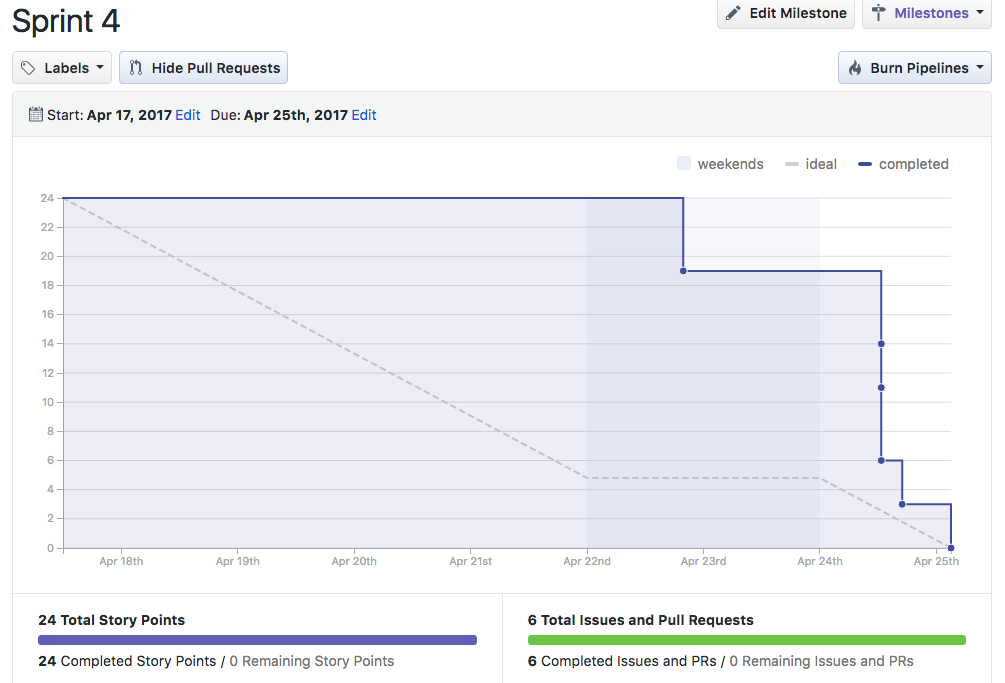
"Grand Valley Bandwagon" is a ridesharing app for GVSU students. Users create accounts using their GVSU email addresses, and can immediately post their ride requests in an instant or recurring format. Other registered users can view these requests and make offers to drive those students. Once both parties accept, information is exchanged and the users can track each other in real time on a map for the duration of the ride. The app was built in Swift and uses Firebase, Google Maps, and Google Places.

**Introduction**

Grand Valley Bandwagon allows users to request rides, for small or no fees, from other users. This idea came from a demand for more or at least simpler, transportation between Grand Valley State University campuses. At peak hours of the day, the Rapid buses that travel between the campuses can reach capacity and cause a student to wait for two or more bus cycles before the are able to get on. There also exist a number of students who drive between campuses. We suspect that many students, if given the opportunity, would rather carpool than take the bus. If they’re even willing to pay a couple of dollars to do so, that may be enough to entice drivers-already headed that direction-to take them along.

The app uses Google Maps to show “drivers” the locations and requests of “riders” looking for rides. The drivers can see what the rider is willing to pay and where they are going, and choose to accept any of these requests. Multiple drivers can accept a single offer. The rider is then able to see on their map, the locations of all the drivers who have accepted their request. They then choose one to accept a ride from and this completes the request-accept cycle.

**Burndown Chart**



**Firebase**

We used Firebase as our way of authenticating that riders and drivers are a Grand Valley State University Student by only allowing people with a GVSU email account to register. We also use Firebase as our backend database for storing information after registration including name, location, requests, rating, car information, etc. We store users locations here which allows users to see other users on the map so they can accept rides. We integrated this into our app via cocoapods.

**Google Maps**

We use Google Maps to gather location data and then display maps. Users can center their map over their location and riders can see drivers that in the “active drivers” state meaning that they are available to give a ride and drivers can see riders that have requested a ride. We created a custom pin for drivers so they appear as cars instead of markers. User locations are updated to firebase and then pull by other users so it is displayed on their map every time a timer goes off. We integrated this into our app via cocoapods.

**Google Places**

We use Google Places in our app when a rider is requesting a ride to choose their location. This allows users to pick latitudes and longitudes that are not actual addresses. This also allows users to enter locations like “Grand Valley State” and it will pull the latitude and longitude of that place from Google Places. We integrated this into our app via cocoapods.

**Requesting a Ride: Start to Finish**

When requesting a ride a rider:

1. Hits the request ride button on the “Ride” tab. This pulls up the request view. The rider then enters their destination, pickup location, rate, and if the ride is reoccuring or not.The rider then waits for a driver to accept their request
2. After a driver has accepted, the rider can accept a ride from a driver. At this point the rider will see the driver on their map in the form of a car and the driver will pick them up.

A driver will:

1. When a driver is ready to give rides they open the left menu item on the “Drive” tab. They then can go online and enter active drivers mode meaning that they are ready to give rides.
2. The Driver can then see any ride requests on their map and can touch them to accept or decline rides.
3. After accepting a ride, the driver waits for acceptance from a rider and then will see the rider they are giving a ride too on their map so they can pick them up.

**Scheduled Rides**

When requesting a future ride a rider:

1. Hits the request ride button on the “Ride” tab. This pulls up the request view. The rider then enters their destination, pickup location, if the ride is reoccuring or not, the time and date of the ride.
2. The rider then waits for a driver to accept their request
3. After a driver has accepted, the rider can accept a ride from the driver at which point they are notified by an alert.

A driver will:

1. When a driver is ready to give rides they open the left menu item on the “Drive” tab. They can then hit the “Future Rides” tab which pulls up a list of all the future scheduled rides.
2. The Driver can then accept these rides by clicking on the ride.
3. After accepting a ride, the driver waits for acceptance from a rider and then they will be notified via an alert.

**Other Features**

We have implemented a number of extra features. These include trip history where a rider or driver can view the rides they have taken. Another feature is the ride summary page which users can get to by hitting their left menu item on the “Ride” or “Drive” view. This view shows them information about the other person they are riding with like their rating, rate, or destination. We store other information in a user’s profile such as their picture which users can view in the accounts page. We chose to implement a floating drawer UI layout which gave us a better looking UI. This floating drawer layout slides a view to the side when a user in the right or left menu.

**Software Engineering Code of Ethics and Professional Practice**

*1.1 Contribute to society and human well-being.*

While creating this app, we like to think we had the best interest of the users in mind. Our goal was to benefit the student body of GVSU by encouraging carpooling and freeing up space on the Rapid buses.

*2.2 Acquire and maintain professional competence.*

Our team exhibited this through consistence meetings and discussion of features and how to implement them. We also learned new frameworks and cross-trained in our assigned areas.

*3.01. Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.*

We learned that we could publish the app in this short time frame and still achieve a high quality product. It took a while to get our time and accomplishment estimates right, but for the last couple of sprints we’ve been more realistic about what we can complete in given time frames.

*3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work.*

We tested our app in pairs to find bugs in our app. We reviewed each other's code contributions. This ensured that we had adequate testing for our app.

*7.08. In situations outside of their own areas of competence, call upon the opinions of other professionals who have competence in that area.*

We started out by asking professor Engelsma where we should start for legal help with our app. We got in contact with the small business development office and then Linda Chamberlain to gain some help from legal professionals

*8.01. Further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process.*

We planned our development in each sprint so that we could ensure that the specification and design of our app went as we wanted.

*8.02. Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time.*

We have a fully functional app with around four hundred hours of work put into it. We were not able to launch the app within the semester because of legal setbacks but we created quality software.

**Teamwork Reflection**

**Nick**

I have learned that working on a team of software developers requires a lot of up front communication and prep work. Defining key tasks and separating what can be done individually from the team is critical to a smooth development process. Discovering each team member’s strengths can also be helpful in delegating tasks. Meeting frequently also seemed to benefit us, especially when viewed in contrast to many teams that had scheduling issues.

**Ryan**

Working with Nick and Mike felt very much like working professionally on a project because we were always very present and very prompt with responses and meetings and questions. It seemed like we were always more or less in sync to get things done and to designate tasks between the three of us. Something I know I have grown on/into as a part of this project was learning how to really phrase my questions and statements to make my teammates see what I mean and the things that are happening that I can understand that they may need a little nudge to grasp.

**Mike**

I already knew Ryan from the GVSU CIS switzerland summer exchange where we both took mobile applications. I think that our project turned out well because Nick was in mobile applications with professor Engelsma during this semester and Ryan and I had already taken mobile applications with him. This gave us a good foundation for making the app. Our schedules turned out to be pretty aligned so meeting up was not a problem. We used google hangouts and other tools to communicate when we could not meet up. I learned a lot about iOS development and teamwork. I also learned that even though we were working on a coding project, a bulk of the work is not coding.

**Conclusions**

We have build out this project more than we initially planned. It started out as a simple “ride now” app that evolved into a “scheduled rides” app and all the complexity that comes with both. We have not completed it to a publishable point, but it is completely functional. What we’ve learned the most is in regard to teamwork and project management. We’ve already discussed teamwork above. All of us “managed” this project is some way. We each had our primary areas/features that we worked on, and had coordinate with the other team members when it came time to implement them together. Managing time, achievement estimates, and dedication to hard work all played equal roles. We would like to finish this app completely, but that depends in part on working around legal and liability issues. Specifically, whether our service would be considered a Transportation Network Company and whether providing the match-making platform can make us liable for damage to persons or property. It seems the next step will be seeking true, perhaps paid, legal advice.