

EE/CprE/SE 491 WEEKLY REPORT 2

02/06/24 - 02/14/24

Group number: 22

Project title: CyRide Visualization

Client: Mohammed Soliman

Advisor: Mohamed Selim

Team Members & Role:

Bradon Buckalew: Programmer

Endi Odobasic: Programmer

Evan Schlarmann: Programmer

Andrew McMahon: Programmer

Week Summary

This week's focus was researching the foundation for the project. Research was conducted on which tech stacks would benefit this project and coincide with team members' skills. The team also designed a workflow document that describes how the application responds to user input and other criteria.

Accomplishments

Research was conducted on possible tech stacks for the CyRide Visualization app. Below are possible tech stacks that could be used, each having its own characteristics that the application could benefit from.

MongoDB - Express.js - React - Node (MERN)

This stack utilizes JavaScript as the main programming language and has a NoSQL database. MongoDB is a non-relational database that stores data with the use of JSON formats instead of tables like how MySQL uses. With the help of the Express framework, it would provide a seamless way to create single/multi page web applications. It also provides a simpler way to create the backend model of an application. The Node framework helps in creating a server side web application so that it can handle extensive data loads as it is asynchronous and follows the event-driven model. React framework gives us a component-based build of an application and allows us to create interfaces efficiently and provide real-time updates. This would be an ideal

stack for javascript developers or if we want to maintain one language for most of our codebase.

PostgreSQL - Django - React

PostgreSQL is a relational database known for its powerful functionality and provides inter-compliance between all kinds of important features regarding data security. It is also open to data types such as JSON or XML and is a great help when working on the backend of applications. Django is based on Python and follows the MVT structure (close to MVC). Some sought-after features are that it comes with built-in authentication, routing, and others. ORM is also a provided layer that makes database interactions much easier. Django utilizes Python and can send data to React, which uses JavaScript. This stack has an SQL database that can connect with Django.

PostgreSQL - Django - Flutter

Flutter is a SDK that allows the building of natively compiled applications from a single codebase. It provides a vast amount of different tools to create a customized UI. It's also fast as it uses its own engine. This is a mix of using Django as the backend and Flutter as the front end for the application. It would have a SQL database connected with Django.

Firebase - Flutter or Hive - Flutter

Flutter can be used as frontend and backend and requires the programming language Dart. This would be more unfamiliar to our team. Hive is a SQL database that can connect with Flutter applications. Firebase, on the other hand, is a NoSQL database that connects easily with Flutter and has additional features like hosting a potential server.

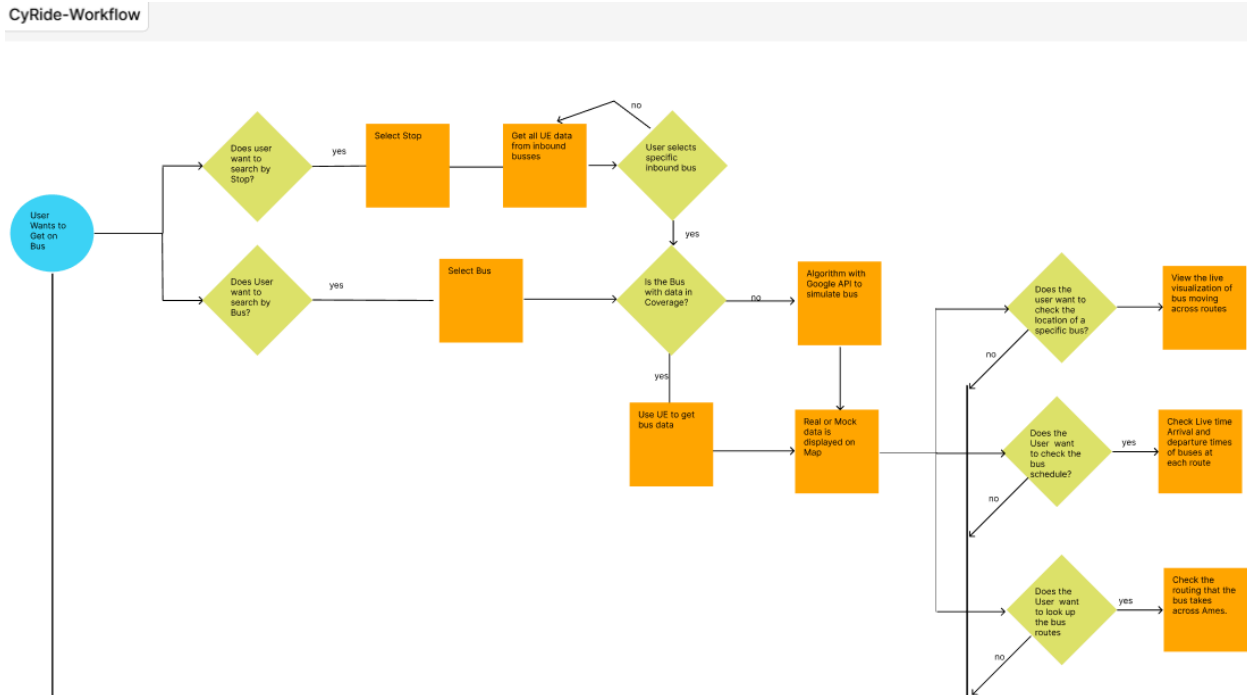
All of these applications can be ported to mobile devices. React must be written using React Native in order to have simultaneous development for web and mobile applications. Flutter can be developed for web and mobile using the same codebase as well. For storing tons of data that is semi-structured, like the locations of all the buses, it may be best to use a NoSQL database. This could result in faster querying and an increase in scalability.

Python within Django is very efficient in computing large datasets and may be the fastest/easiest language when trying to predict entity locations. Javascript is most well known by our team, making MERN a possible easier stack to implement. Flutter is a great framework but may be the hardest to implement as our team lacks skills in the DART programming language.

Current Cyride Bus Tracking

Through research, it was found that CyRide uses GPS to track their buses. The buses communicate directly with their servers and databases giving their location information. All the stored data is used in algorithms to calculate predictions for bus arrival times which are then shown to their users. This is what allows Cyride to provide accurate predictions for bus time arrivals.

Workflow Diagram



The workflow diagram was assigned to us by both our client and advisor, to show we understand the basis of the project, the choices a user has, and to lay the groundwork for the design of our app.

Pending Issues

None

Individual Contributions

<u>NAME</u>	<u>Individual Contributions</u>	<u>Description</u>	<u>Week Hours</u>	<u>Cum. Hours</u>
Evan Schlarmann	1)Researched tech stacks 2)Researched Current Cyride Implementation	1)Researched and found tech stacks that would be suitable for the project giving details on what programming languages and databases they use. Then compiled the benefits that each stack could bring to the project and why they should be considered. 2)Also conducted research into how Cyride tracks their buses and provides arrival times.	3	6
Braden Buckalew	1)Started Workflow diagram 2)Requested Ubuntu VM from ETG	1)Created a figma group for our group to work on, create components for workflow logic, started basic flow if the user wants to select a bus 2)Sent in a request for an Ubuntu system so we could have a virtual environment to run and experiment with the backend	4	6
Endi Odobasic	Tech-stacks and Empathy Map	Looked into and researched more in-depth on some of the possible techstacks our project might utilize. Gave descriptions of what some of the languages, databases, or frameworks were. Also, I took a look at the Empathy map from last week to get a better representation of what our audience might look for in an application and how we could go about making an application that works great. I provided a description of what our audiences' issues were with the current CyRide mobile application. There are some possible things to fix or even add.	3	6
Andrew McMahon	Finished workflow diagram, initialized & finalized report	Modified workflow diagram to distinguish user choice of specific bus view, or view by stop. This update finished the workflow diagram, which represents how our app should react based on user choices. Set format for this report for team members to fill in, and finalized the report to have even pages and no non-symmetric qualities.	3	5

Plans For the Upcoming Week

- Tech-Stack - Braden
 - Due: (02/21/2024)
 - Will get the team together in person to decide the stack that we want to use as a team that's not only best for the project but also one that everyone is comfortable with.
 - Project Setup - Evan
 - Due: (02/21/2024)
 - Will create the first files for the project within GitLab to set the groundwork.
 - Setup Cyride Meeting - Andrew
 - Due: (02/21/2024)
 - Will set up a meeting with the CyRide team so that our group can get a tour of the facility and ask all of our questions about CyRide's tracking system, to better understand some of the logic behind our project.
 - Practice With GitLab - Group
 - Due: (02/21/2024)
 - Everyone will make sure they can pull and commit to the project in GitLab.
-

Weekly Client/Advisor Meeting Summary

Finalized understanding of our project. Talked about the idea of acquiring our own UE for testing outside the bus, necessitating detailed planning with hard deadlines for tasks. Trello will be used more extensively for project management, while GitLab will handle technical requirements and code-based issues. Decision-making will primarily involve the team, with the need to persuade the client on key decisions. An Agile work environment will allow for quick adaptation to changing circumstances. Additionally, we plan to create a workflow, develop an interactive flow showing the bus's location, predict the bus's location using Google API when the UE is out of coverage, and utilize mock data to prepare for real UE usage.