

EE/CprE/SE 491 WEEKLY REPORT 10

04/9/24 - 04/16/24

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, we continued to work towards displaying mock data on the Google Maps interface. This time, we tested some larger-scale features like routes and implemented them on the Google Maps UI. We also began to link the frontend and backend by introducing tests and finalizing endpoints via Postman on both sides of the application. The backend also has an API to improve communication between the frontend and backend developers, and provide a readable interface for all that use or inquire about the application.

Accomplishments

Environment Update:

Split the codebase into production and development environments. This allows for building the application with a development database that can be altered without repercussion. The production environment is used when deploying changes to the main branch and ensuring that the application functions correctly.

Backend Updates:

We updated the existing tables to ensure they are set correctly for our project, and also created a new table to hold all stops for the frontend. We needed to ensure the correct keys were used to set and retrieve data. Also, we had to add more fields that we needed that tie into the Machine Learning Model and the particular data it needs. Finally, we created a basic API implementation to ensure the frontend, backend, and future users are all well-informed and understanding of the project.

Bus Route:

Created the bus route we should use to test our application for bus connectivity. The UE will be routed along the brown route, so we created a path within the application to display it as necessary. Along this path, we will display some sort of bus icon from the live web sockets to update bus locations in real time.

Testing Frameworks:

We created testing frameworks on both layers of the application so that when it comes time to test things out and ensure they work entirely how they are supposed to, they are.

Django Websockets – Frontend:

We began implementing the data from the WebSocket – as currently outlined in Django (backend) – in the frontend, once implemented, this creates a basic link between the backend and frontend.

Pending Issues

- Polyline Route
-

Individual Contributions

<u>NAME</u>	<u>Individual Contributions</u>	<u>Description</u>	<u>Week Hours</u>	<u>Cum. Hours</u>
Evan Schlarman	1) Updated the Django models 2) Created different environments 3) Updated bash files	1) The Django models now follow SQL naming standards when migrating and Python naming standards when getting data from a query. I also fixed the models and can now get data from the database. 2) There are now two environments for developing in Django. By default, all commands and changes are made to the DEV environment. The server is the only entity that changes the	8	59

		<p>PROD environment when deploying changes to the main branch.</p> <p>3) Updated the bash files for CI/CD commands. The scripts will now detect failures in deployment and propagate the failure up to the GitLab pipeline.</p>		
Braden Buckalew	<p>1) Display mock location on frontend</p> <p>2) Start Frontend Unit Tests</p> <p>3) Start Backend Unit Tests</p>	<p>1) Work on displaying WebSocket information to simulate a small route and changing the icon color if the connection value is false.</p> <p>2) Intialized Frontend Unitesting that will run on the pipeline</p> <p>3)Initialized Back Unit testing that will run on the pipeline</p>	10	57
Endi Odobasic	<p>1) Bus Icon</p> <p>2) Bus Route Color</p> <p>3) Starting Point (Reset) Button</p>	<p>1) Working on adding a bus onto the Google Maps route based on random routes along the path.</p> <p>2) Try to make the path turn another color in case there is no connectivity from the bus.</p>	9	59
Andrew McMahon	<p>1) Added stops table</p> <p>2) API & its implementation</p> <p>3) ARIMA implementation research & documentation</p>	<p>1) Added all stops in CyRide system (via CyRide contact) to backend so they can be displayed on Google Maps UI. Also, fixed errors pertaining to one vehicle having many locations on the backend.</p> <p>2) Created a temporary basic API documentation. This adds some important documentation and should also increase comprehension of the application by future users, as well as help the frontend during the dev stages.</p> <p>3) Researched and gathered findings on why the ARIMA ML model fits our project the best, and how it can be transitioned to a SARIMA model.</p>	11	57

Plans For the Upcoming Week

- Create and practice the final presentation - Team
-

Weekly Client/Advisor Meeting Summary

In our weekly meeting, all of the team members listed out our past week's accomplishments. We displayed everything we got done for the week and we were able to show our respectable advancements in the overall development of the application. We got some feedback from both the client and advisor, including using timestamps as a primary key, suggested models for ML, and suggestions on the Google Maps UI that we're developing. We also laid out our plans for the upcoming week – both individually and as a group. We explained how we wanted to add a few more UI displays with the help of Google Maps API and come to a great stopping point for the next week.

Building on the stopping point, we discussed our plan for the next few weeks to work on wrapping up the technical tasks for the semester. Since the end of the semester is near, we then planned to transition off of technical tasks, and focus solely on the faculty panel presentation, documentation, and all other “clean-up” tasks for the semester. Our team will prepare and put our slides in our shared box folder with our Client and Advisor, so the development of them is a group experience, and we are given as much feedback as possible leading up to the presentation.