Status Report 2

Group 5: Taylor Hopkin, Davin Lim, Oliver Tipton, Kiko Lancastre

April 7th 2024

1 Introduction

1.1 Highlights

- What was the plan for this iteration?
  - The plan for this iteration was to fully integrate the data with Dijkstra's algorithm.
  - Similarly, we wanted to have a functioning UI that featured drop down menus, search bars, had integrated google maps API, and had some version of backend algorithm integration.
- Highlight what the team accomplished.
  - We hit all of the points in our plan in the answer above. Similarly, we were able to fine tune the UI so that it was more aesthetic than we had originally planned to have at this point in the project. We found that as we were building the site, it was easy to update the looks of the website and it did not take extra time that would have prohibited us from working on our other goals.

1.2 Changes

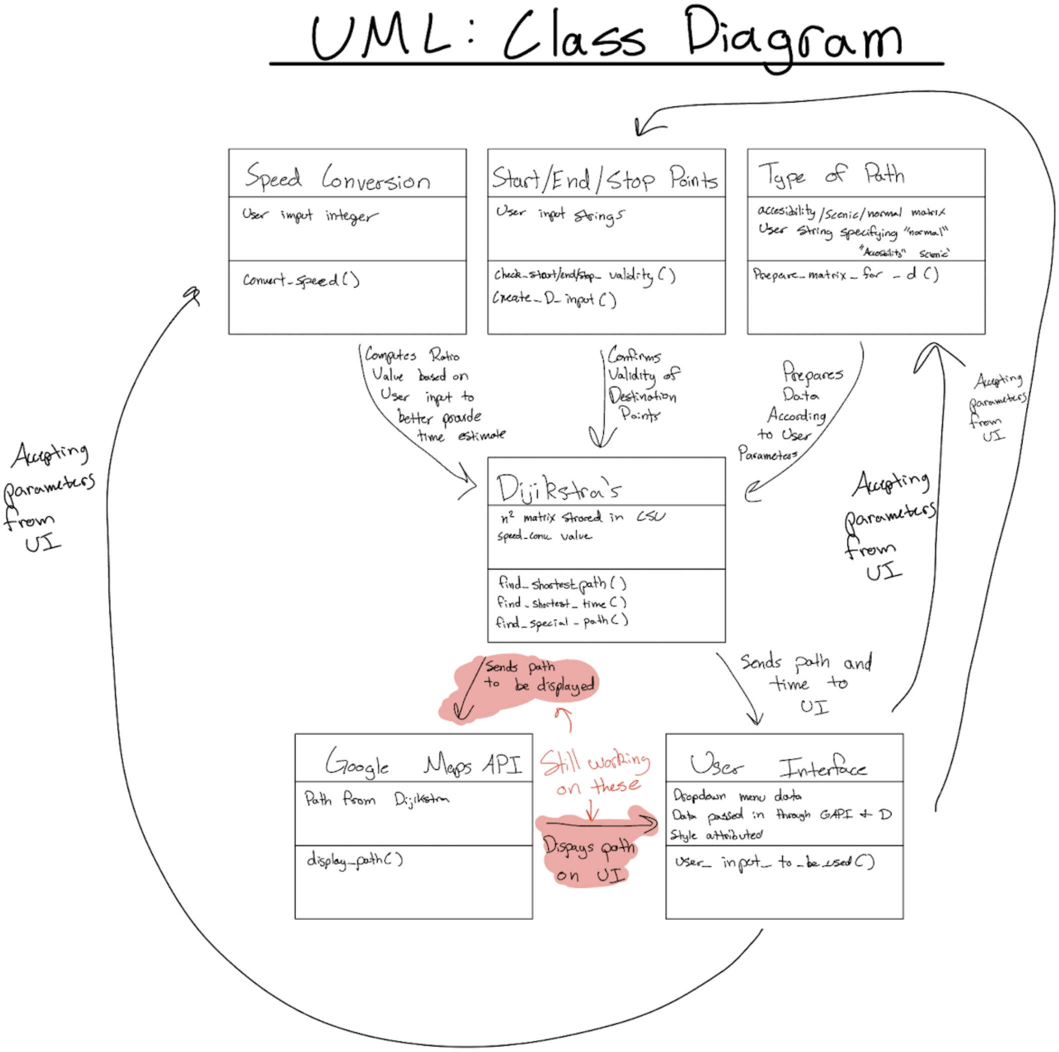
- Summarize any major changes since Status Report 1.
- Include each change’s date, motivation, description, and implications.
- If there were none, note that there were no changes.

There have been no major changes to the scope of the project since Status Report 1.

Progress changes have been logged on selection 4.

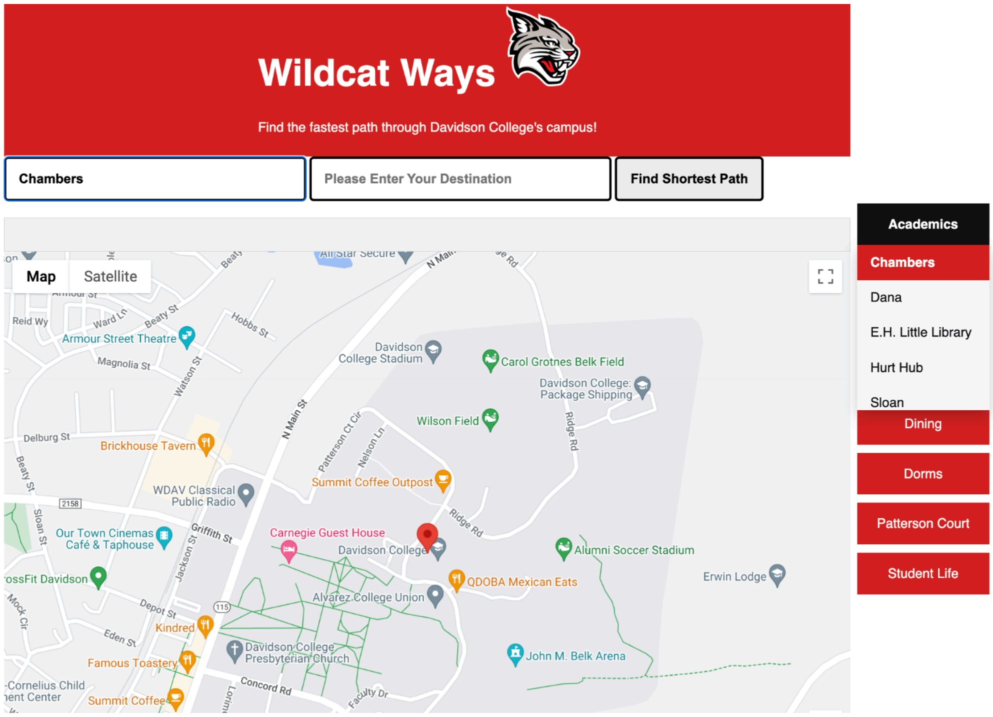
2 UML: Class Diagram

- Draw a class diagram that only includes the important classes your group mainly worked on during this iteration. Identify a single owner on the team for each class, even if multiple team members contribute.

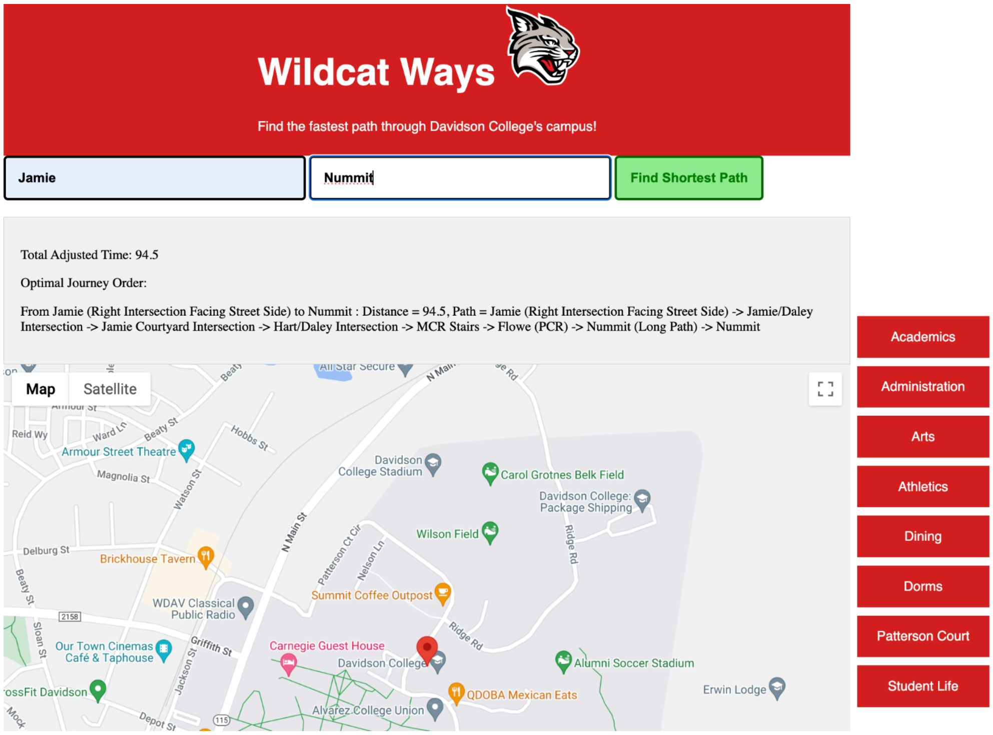

UML: Class Diagram
Speed Conversion
User input integer
Convert_speed()
Start/End/Stop Points
User input strings
Check_start/end/stop_validity()
Create_D_input()
Type of Path
accesibility/scenic/normal matrix
User string specifying "normal" "Accesibility" "scenic"
Prepare_matrix_for_d()
Computes Ratio Value based on User input to better provide time estimate
Confirms Validity of Destination Points
Prepares Data According to User Parameters
Accepting parameters from UI
Accepting parameters from UI
Accepting parameters from UI
Dijikstra's
n^2 matrix stored in CSV
speed_conv value
find_shortest_path()
find_shortest_time()
find_special_path()
Sends path to be displayed
Sends path and time to UI
Still working on these
Google Maps API
Path from Dijikstra
display_path()
Dispays path on UI
User Interface
Dropdown menu data
Data passed in through GAPI + D
Style attributed
User_input_to_be_used()

3 Current Status

- Add screenshots of the parts that are working



Chambers



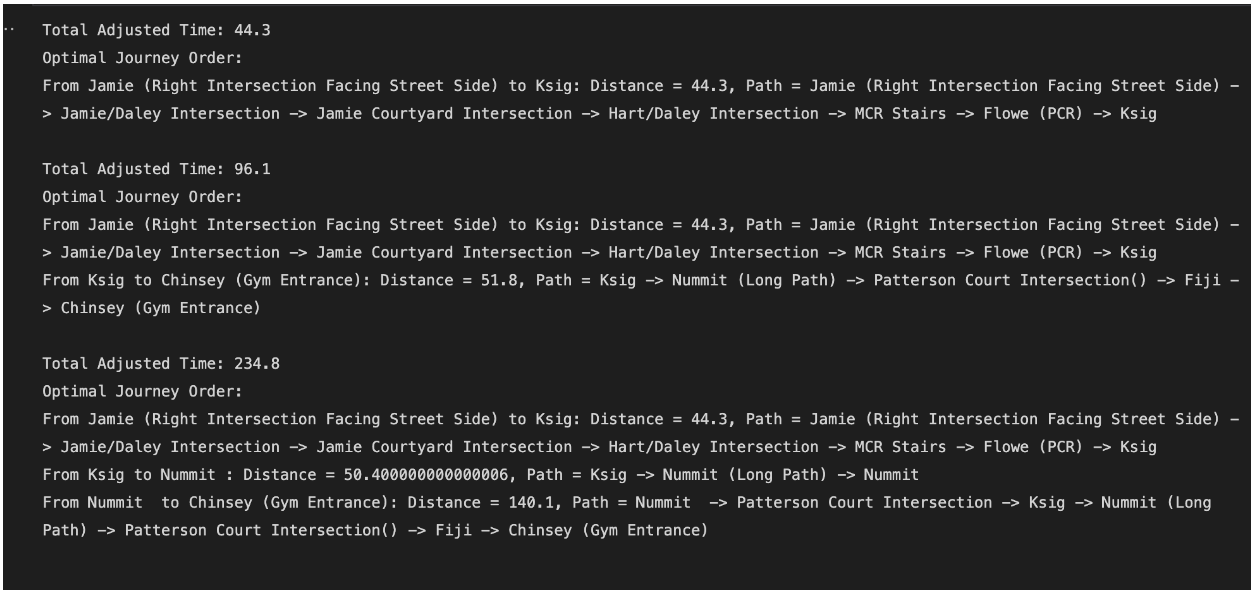
Jamie

Nummit

Total Adjusted Time: 94.5

Optimal Journey Order:

From Jamie (Right Intersection Facing Street Side) to Nummit : Distance = 94.5, Path = Jamie (Right Intersection Facing Street Side) -> Jamie/Daley Intersection -> Jamie Courtyard Intersection -> Hart/Daley Intersection -> MCR Stairs -> Flowe (PCR) -> Nummit (Long Path) -> Nummit



```
Total Adjusted Time: 44.3
Optimal Journey Order:
From Jamie (Right Intersection Facing Street Side) to Ksig: Distance = 44.3, Path = Jamie (Right Intersection Facing Street Side) –
> Jamie/Daley Intersection -> Jamie Courtyard Intersection -> Hart/Daley Intersection -> MCR Stairs -> Flowe (PCR) -> Ksig

Total Adjusted Time: 96.1
Optimal Journey Order:
From Jamie (Right Intersection Facing Street Side) to Ksig: Distance = 44.3, Path = Jamie (Right Intersection Facing Street Side) –
> Jamie/Daley Intersection -> Jamie Courtyard Intersection -> Hart/Daley Intersection -> MCR Stairs -> Flowe (PCR) -> Ksig
From Ksig to Chinsey (Gym Entrance): Distance = 51.8, Path = Ksig -> Nummit (Long Path) -> Patterson Court Intersection() -> Fiji –
> Chinsey (Gym Entrance)

Total Adjusted Time: 234.8
Optimal Journey Order:
From Jamie (Right Intersection Facing Street Side) to Ksig: Distance = 44.3, Path = Jamie (Right Intersection Facing Street Side) –
> Jamie/Daley Intersection -> Jamie Courtyard Intersection -> Hart/Daley Intersection -> MCR Stairs -> Flowe (PCR) -> Ksig
From Ksig to Nummit : Distance = 50.400000000000006, Path = Ksig -> Nummit (Long Path) -> Nummit
From Nummit  to Chinsey (Gym Entrance): Distance = 140.1, Path = Nummit  -> Patterson Court Intersection -> Ksig -> Nummit (Long
Path) -> Patterson Court Intersection() -> Fiji -> Chinsey (Gym Entrance)
```

- Map the screenshots to the class diagram. In other words, which class does the feature you show in the screenshot belong to? Multiple classes may belong to a single feature. Clearly describe what those are.
    - For the first and second screenshot of the UI, the features shown in the screenshot belong to the "UI" and "Google Maps" classes. The screenshot of the website shows that a user can select a starting point and destination from the drop down menu on the right-hand side, which is a feature that belongs to the "UI" class. Once the user finishes inputting the starting point and destination, the user can hit the "Find Shortest Path" button (which is part of both the "UI" class and "Dijkstra's"). Once the button is pressed, the "UI" class displays the shortest path to get from the starting point to the destination and the time it takes to take the path onto the website.
    - For the third screenshot that shows the output of Dijkstra's algorithm, the class Dijkstra's is being shown. Inputs from the three classes, speed conversion, start/end/stop points, and type of path were passed through the diagram along with the data matrix. The screenshot shows a successful output of the algorithm that shows the path that should be taken between 4 locations across campus. The output is ultimately shown on the UI and after our next sprint will be passed to the Google Maps API to be displayed on an actual map.

4 Project Management

- Continue to maintain the Change Log. Add any new changes to the project, tracking the date and description of each change. Use the table below:

Integrated data into the matrix for Djakostra's algorithm.	March 30th
Added feature that allows users to include more than 1 destination in the algorithm	March 30th

Created basic UI for Wildcat Ways website	April 1st
Added drop down menu for different categories of locations	April 1st
Integrated the Google Maps API into the website	April 1st
Began working on customized paths/ walkways/buildings to the Google Maps API for Davidson College’s campus	April 1st

5 Review and Retrospective

- What went well?
  - We were able to make far more progress on our UI than the previous iterations, ultimately finishing with a functional UI set for back-end integration.
  - We were able to integrate the search function to talk directly to the backend Dijkstra's class that ultimately shoots out the route to take.
  - We were able to meet frequently as a team.
- What didn’t go well?
  - We were overly optimistic about our coding skills and how long it would take to do some of the steps in this sprint. We spent far more hours than we predicted we would on creating our UI and data integration into Dijkstra's class.
  - However, both things were ultimately accomplished and it leaves us in a good spot in this next sprint to be able to really fine tune our product and presentation.
- For the goals that were not met, what were the issues?
  - Our goals were met.
- How do you plan to overcome the issues?
  - No issues to overcome.
- What do you plan to do differently in the next iteration?
  - We liked our setup from a meeting and team collaboration standpoint. There is no need for us to change anything there. The only change going into this next iteration may just be a slightly different mindset and understanding that the difficulty of the goals of this project may be longer and more strenuous than we thought.

6 Team Management

- What were the team roles for this iteration?
  - Taylor: Scrum Master / Developer
  - Davin: Product Owner / Developer
  - Oliver: Developer
  - Kiko: Developer
- What did each team member contribute?

    - Davin designed and coded the website UI for Wildcat Ways
    - Oliver worked on integrating the Google Maps API into the Wildcat Ways website. He also made progress in customizing the Google Maps API for Davidson's campus (etc. creating new paths that are not displayed on google maps, adding new buildings that have been built but not updated on google maps)
    - Taylor finished the data integration into Dijkstra's algorithm class so that optimal outputs are now being discovered and calculated. Also, kept customers updated on the progress of our software
    - Kiko completed the data collection and added the 'how long it takes to walk 10 meters' feature. Also, kept customers updated on the progress of our software.
- What were the challenges regarding team management, e.g., regular meeting, etc.?
    - Our initial plan was to increase the frequency of our weekly meetings. However, we struggled to find a time that works for everyone in the group to meet more than twice a week.
- What are the plans to overcome the challenges?
    - To overcome the challenges of increasing the frequency of our weekly meetings, we will utilize the period of time before and after class on Tuesdays and Thursdays to hold our weekly meetings.
- If you were the third party who knows very well about your team, what suggestions would you give to your team?
    - Do a better job of effectively communicating within the team for weekly meetings and progress checks
    - Figure out the most efficient way to check each other's progress before each meeting

7 Goals for the Next Iteration

- Write the next iteration's product log.
    - As a prospective student, I want to be able to see my optimized paths visualized on the WildcatWays Google Maps so that I can know where I'm going and not get lost.
    - As an elderly user, I want the ability of only having accessibility paths displayed to WildcatWays so I can take accessible paths.
    - As a busy student, I want to be able to input multiple destinations on the WildcatWays website, so that I can map out my day and not waste any time.
- Write the next iteration's sprint log.
    - Set up a back-end server to host our back-end functions.
    - Integrate Google Maps API with front-end UI and back-end algorithm
    - Meet with customers to show demo and get feedback
- Other than the issues discussed in Section 5, i.e Review and Retrospective, what potential challenges do you see in the next iteration?
    - The most difficult/technical aspect throughout this project has been learning how to use the Google Maps API and integrate it with our data. As we now have our data integrated with the algorithm and connecting back-end to front shouldn't be too difficult, we have placed this Google Maps API Challenge at the forefront of our next iteration.
- Briefly explain how your team would overcome each challenge.
    - This Google Maps API challenge requires diligence and research for it to be a successful functionality of our project. After learning how to add the map representation to our UI, we are now diving deep into understanding how to add custom paths and times, both of which are not a part of the given Google Maps API for developers. As a result, we are taking a methodical approach through watching videos, trial and error, and multiple different methods to ensure the WildcatWays functions at the level we want it to.