Project Plan - SafeRide

Team: #0000FF Thunder
An Ta
Andy Lee
Leon Chen
Orion Tang (Team Leader)

Date Issued:
October 27th, 2021

Goals:

The goals for this project for the next semester, where the team will attempt to code the features, and project specified in the documents created in the first semester. Goals for next semester will be split up into different sprint goals, where if we don't complete the goals for that sprint, we will put it as the goal for the next sprint, and will prioritize the previous sprint uncompleted goals.

For features that cannot be completed, they will be left out, but will prioritize core features over unique features. For example, in sprint 2, core features include User Management, User Access Control and Backend network, while the unique features include Map Overlay, Hazard Labelling, and User data collection. If there is spillover from sprint 1, for example incompleted features from core, User registration and login/logout, and map waypoints from unique features, we will prioritize core feature completion first, user registration and login/logout first, then after completing everything in sprint 2 first, then go back and finish unique feature spillovers, which will be the waypoint feature. This way, in theory, if there are features that are uncompleted at the end of the semester, they will only be unique features, unless all unique features are uncompleted, which could then also include unfinished core features.

Cost:

Costs should be kept at a minimum. Software should be free or at least free with an account for a limited time greater than the service is going to run for. For example, aws load balancer is 750 hours free per month, with network and application load balancer with an aws free tier account, which means that it is available to use. The same applies to aws EC2 instances which is limited to 750 hours per month free limited to one t2.micro instance. T2.micro includes 1 vCPU, and 1GB of ram. We will try to limit our application to this one VM, however depending on our speed needs we may need a few more instances for our microservices. We will spend up to $5 per month for instances on aws. We believe that to provide adequate service to one customer, the client, testing not included, 1 t2 micro instance should be enough. The application will also only be online or run when required. For example, the client should specify which periods during the month or week the application should be online, such as during demos. This could cut costs by a huge amount and increase our amount of allocated instances, keeping downtimes up.

Software costs

- Windows 10

Hardware costs

- Aws instances
    - $5/month (max)

Risk Estimation:

Possible risks include features not being done in time, where there are feature spillovers from sprint to sprint. The mitigation for this is to acknowledge and choose certain features that might not get done, and prioritize core features over unique features. Other risks include not providing enough compute in our backend such that our speed requirements are not met, for this we have a budget of $5 per month for aws instances to mitigate this speed problem. Other risks in our team include meeting schedules not being synchronized, meaning insufficient planning at the beginning or sudden events leading someone to not be able to attend meetings. For this risk, we will try our best to come up with a plan for future sprints, as well as increased communication to let others know that they could not go to the meeting. There is a scope creep risk where problems could lead our team to include more features as time goes on. For this, we should stick to our core feature priority and include extra features as needed after core features have been done.

Methodology:

Sprint Planning

- Visiting Project Backlog to find items and task that need to be completed this milestone
  - Pull in work that group can deal with
- Scrum Master will assign tasks/features to the team.
- Task Breakdown will occur after the assigned feature
  - "Task Breakdown"
  - Breaking down features into multiple task to complete features
  - These single items to complete features, are called Task (daily items)
- When the previous stages are done
  - Team/Individual Capacity VS Velocity
    - Measure the team capacity of how much story points the team is able to complete during this sprint duration
    - Measure the velocity of the sprint duration is
      - This is the rate of completion
      - Tracking story points throughout the sprint duration
    - ** STAY UNDER CAPACITY LOAD **

Project delegations for this project will be split up according to several things.

First

- Unique feature wise
    - Route analysis
        - Leon, Orion
    - Map overlay
        - Andy, Frankie
    - Hazard labelling
        - An, Andy
    - General map features
        - Andy, Orion
    - Data Collection
        - An, Leon
    - Route waypoints
        - Frankie
    - History tracker
        - An, Orion
    - ML-based algo
        - Frankie, Leon

Second

- High level design wise, as best as possible will not contradict too much with feature wise delegations.
    - Presentation layer
        - Andy, Frankie, Orion
    - Logic layer
        - Frankie, Leon, Orion
    - Data layer
        - An, Leon

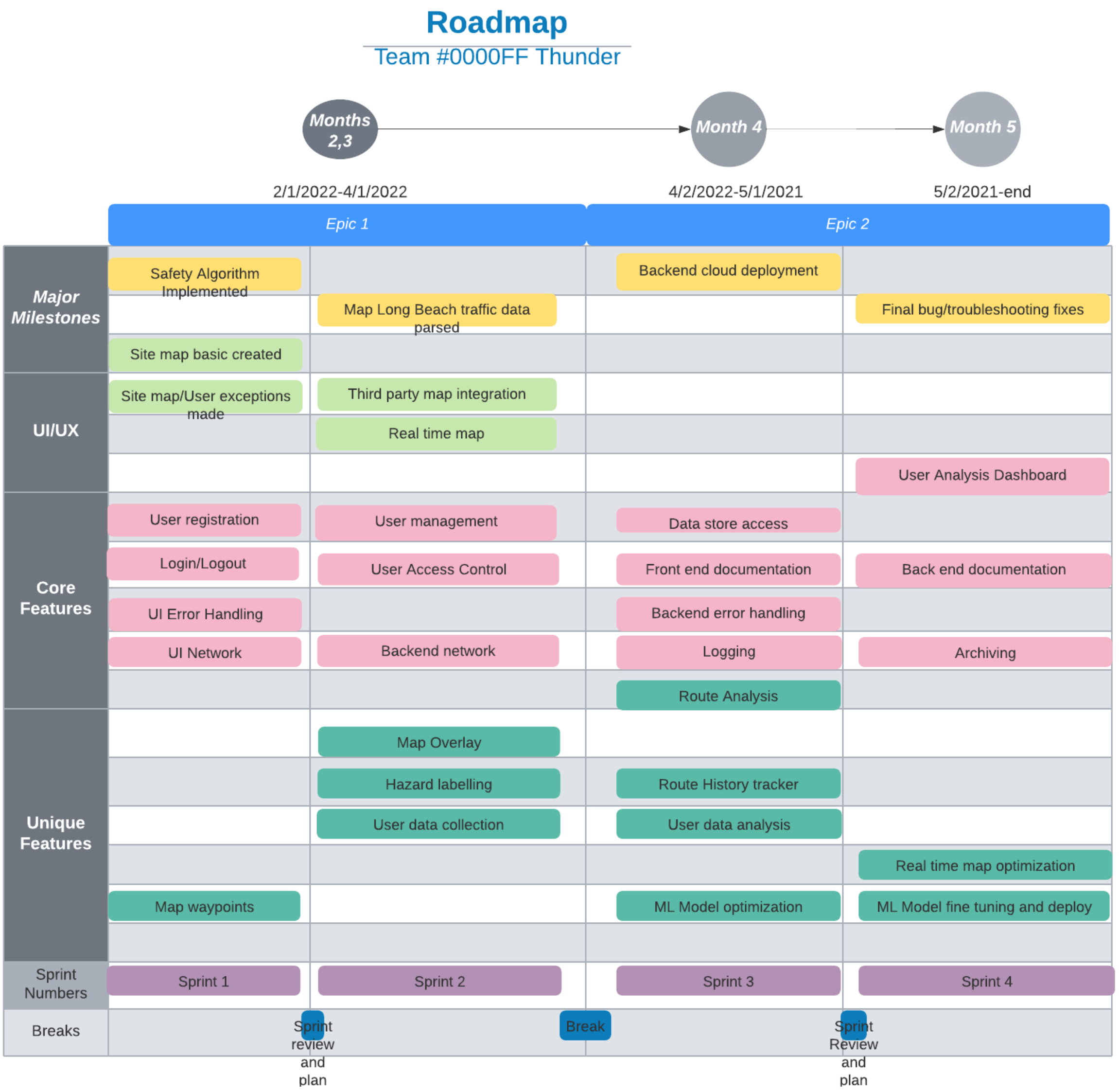
Third

- Grouped core features, will group similar core features and map to groups of developers
    - (Registration, Login/Logout)
        - Andy, Orion
    - (User Management, User Access Control, Usage Analysis Dashboard)
        - An, Andy, Frankie, Leon
    - (Logging/Archiving, Error Handling)
        - An, Leon, Orion
    - (Data Store Access, Network Communication, Documentation)
        - An, Andy, Frankie

Fourth

- Testing, testing will be split up according to different techs
    - User interface (backend to frontend interface)
        - Orion, Andy, Frankie
    - Controllers (frontend to backend interface)
        - Leon, An, Orion
    - Backend (backend/database interface)
        - An, Andy, Frankie, Leon, Orion
    - Database (data testing)
        - Frankie, An, Andy

Project Roadmap:



Roadmap

Team #0000FF Thunder

Months 2,3 → Month 4 → Month 5

2/1/2022-4/1/2022

4/2/2022-5/1/2021

5/2/2021-end

	Epic 1		Epic 2	
Major Milestones	Safety Algorithm Implemented		Backend cloud deployment	
		Map Long Beach traffic data parsed		Final bug/troubleshooting fixes
	Site map basic created			
UI/UX	Site map/User exceptions made	Third party map integration		
		Real time map		
				User Analysis Dashboard
Core Features	User registration	User management	Data store access	
	Login/Logout	User Access Control	Front end documentation	Back end documentation
	UI Error Handling		Backend error handling	
	UI Network	Backend network	Logging	Archiving
			Route Analysis	
Unique Features		Map Overlay		
		Hazard labelling	Route History tracker	
		User data collection	User data analysis	
				Real time map optimization
	Map waypoints		ML Model optimization	ML Model fine tuning and deploy
Sprint Numbers	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Breaks	Sprint review and plan	Break	Sprint Review and plan	