RoadBuddy

Developed by **RoadBuddy Team**

Sprint 4 Report

Actions to Stop:

- Using websocket io for messaging and notification.

Actions to Start:

- Using pulling to get real-time updates on notification.
- Using a real-time Firestore listener for ride chat messages.

Actions to Continue:

- Completing tasks for MVP
- Styling for pages without styling
- Frequent communication
- Updating scrum board when task progress changes
- Merging to GitHub
- Completing tasks for current and past sprints.

Completed Work:

- **#01: I, as a driver, should be able to delete my service so if emergencies happen I can choose not to drive.** [5 story points]
    - Provide an option to allow user to cancel/delete a ride [2 hours]
    - Create cancel and delete endpoints in the backend to allow the frontend send requests when a user wants to cancel or delete a ride.
      [3 hours]
- **#02: I, as a user, need to be able to message my driver/passenger(s) so I can organize the carpool.** [13 story points]
    - Frame of Messaging UI In React Native (no DB connectivity) [2 hours]
    - Messaging UI Styling for React Native [4 hours]
    - Messaging functionality in backend [6 hours]
    - Connect Messaging UI Frame to backend [< 2 hours]

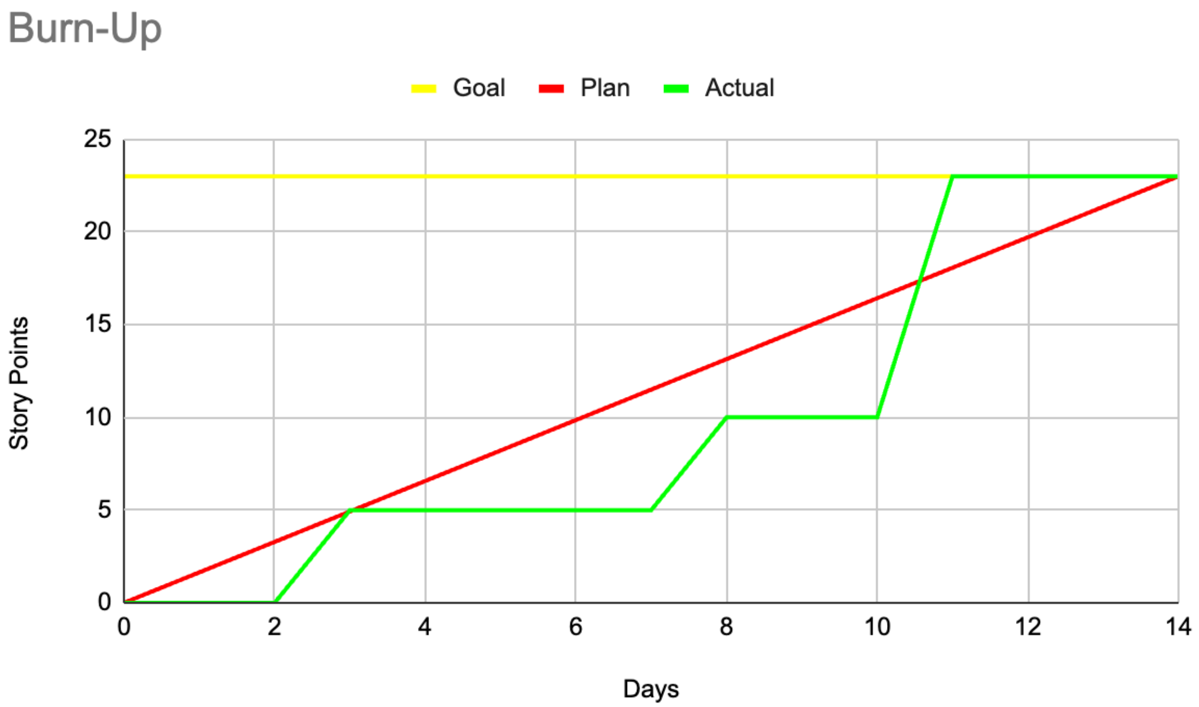
- **#03: I, as a driver and passenger, want to be notified when a passenger in my ride has cancelled or the owner of a ride has deleted a ride that I am a passenger in.** [5 story points]
    - Set Up an endpoint in the backend to allow the frontend to ask if the user has any new notification. [2 hours]
    - Send a notification to the owner of the ride if a user cancels. [< 1 hours]
    - Send a notification to all the passengers if the owner has deleted the ride. [< 1 hours]
    - All notifications are stored in FireStore. [1 hours]
    - Create notification page in the frontend [1.5 hours]

Uncompleted Work:

Infrastructure:

- Frontend
- Backend
- Database
- Github
- Continuous integration

<u>Burnup Chart (below)</u>


Burn-Up
Goal
Plan
Actual
Story Points
Days