

## **Sprint 2 Plan**

**Goal:** Modify the source code to implement the same rogueness factors that were previously implemented in the python script.

**Total Points: 57**

**User Stories:**

- 1. As a SUMO user, I want to be able to simulate the same rogue cars within a traffic simulation, and observe the data collected from the simulation with a more intuitive python script or vehicle class declaration. [Ryan] (17)**
  - a. Backtrace the traCI source code. (8)
  - b. Implement the same vehicle flag changes that the python script changes to be associated with our created vehicle class. (8)
  - c. Additionally, add sigma and impatience variations to the rogue vehicle class. (1)
  
- 2. As a developer, I want a more complex traffic simulation with pedestrians, emergency vehicles, and other abnormalities in speed, safety, and traffic in our simulation for demonstration as well as testing purposes. [Harshaan] (24)**
  - a. Add two lane roads that do not throw visual errors. (4)
  - b. Add pedestrian class walking opposite direction of traffic on designated sidewalk edge class. (2)
  - c. Add emergency vehicles to sim with alternative speed settings (break the speed limit) (2)
  - d. Use the same rerouters and stop lights that were implemented in the example sim into the eichstatt OSM. (8)
  - e. Add to the eichstatt OSM or create Santa Cruz OSM and use netedit to add variation. (8)
  
- 3. As a SUMO developer, I want to be able to take Basic Safety Messages (BSM) and implement them into a new vehicle device so that it can be used to produce DSRC messages [Alexis] (16)**
  - a. Create a new device from the example device skeleton provided within the source code [4]
  - b. Find already-existing BSM parameters provided inside pre-defined devices within the SUMO source code. [2]

- c. Transfer written functions found inside the BaseVehicle files and add them to the new Device. [5]
- d. Create the missing BSM parameters not found locally in SUMO [5]