Divvy Van Navigator Information Scenario
Group 1: Andy Hansana, Kevin Elliott, Vincent Weaver, Ryan Crowley

The Divvy Van Navigator will pull from various Divvy APIs, which provide values including: station name, docks in service, status of bikes/scooters, and location. Other variables should be implemented as needed in order to produce more efficient routing results.

The purpose of this application is to constantly optimize the best convenient route for a Divvy services driver making maintenance stops at various Divvy locations. They should know how many bikes are broken, and how many need to be repaired. The same goes for the Divvy scooters as well. Google Maps, or a similarly reliable location service should be utilized as it will be an effective tool for guiding the driver along the best route possible.

There are numerous factors during the day that need to be taken into consideration in order to produce a convenient route. Primarily, the app should recalculate and optimize the route during high-traffic hours (especially in cities like Chicago). Secondly, the software should pull relevant construction data and alter the route in order to minimize idle time in certain high-concentrated areas. Finally, the same should be done for hazardous weather conditions, which may also cause some of the previous two obstacles mentioned.

Scenario: “Divvy Stats”

On startup the user is met with a notification that displays daily information relevant to the job (i.e. significant traffic, or a larger than usual workload). Next, the application loads the map and Divvy data to the user and applies it to the interface. Now, the driver can be prompted to input what their inventory is. After that, the driver will be prompted to pick up a certain amount of each inventory item. The app is now ready to navigate. After weighing each option on the map with weights such as size, traffic, percent damaged, and daily users, the driver should now have a route to the first station.

After unloading any equipment and repairing as needed, the driver can now report the change in inventory and receive new instructions. If the van is empty or low the app can route them back to a pick up location, if not then the driver is guided to the next station with a new set of tasks. This process loop is what the driver will be using while they are out in the field, with the ability to take a break, end their shift, and change pathing based on unpredictable scenarios like a car crash blocking the street. Once the driver decides to end their work day, the app will navigate them back to the starting location where the app was opened, otherwise the location can be changed to an alternate.

The user-friendly GUI would allow the driver to see all nearby stations and the number of bikes and scooters that are inoperable in each station. The first screen would prompt the driver with an issue/objective of the number of bikes that need to be picked up and moved to another station. The driver will be asked to virtually select the objective they wish to complete which will include the type of vehicle (scooter or bike), and the time it will take to complete the route transport.