Sage Drewke, Brana Kim, Joshua Hernandez

*PitCrew*

*Project GasUP* Requirements Document Draft

# Introduction (BK & SD)

This document contains a detailed summary of the GasUp app, its functions along with multiple descriptive diagrams displaying the utilities. The description model describes the requirements of the system and details the functions. The class diagram displays all of the system objects, the attributes, and known methods. The use case diagram shows all the possible uses of the system, and the use case scenarios breaks down into more detail. Lastly the system sequence diagrams showcase each use case scenario in detail.

# Description Model (BK & SD)

Using text, describe the requirements for your system. Expand on the function section from your project plan. Include requirements for the following categories: Output, Input, Processes, Performance and Security.

1. Input and Output (SD)

The app will require inputs such as a start destination, zip code, or gas mileage information to properly output directions and accurate miles per gallon calculations. Other inputs from the user can include a numerical rating, a review, and user information. Outputs include gas stations within a certain radius, gas station convenience stores, and notifications.

1. Performance (BK)

The app should be able to quickly load each new page in a matter of seconds. Ideally when the user first opens the app it should load within 10s and open up to the home page. If a user is logging in the login time should also be within a few seconds. When opening the maps it should only take up to 10s to display. Displaying location of the gas stions should only take a few seconds. However, loading directions may take up to 30s depending on the route. If the user pulls up the information on a certain station this should only take up to 5s. Ideally the app itself be able to support any amount of users, but the server hosting the accounts may cause longer loading times depending on how many people are using the app at once. The two main areas where the app may slow down is logging into the user account and loading directions. This is mainly due to the fact that the app will have to communicate with an outside source to load the information. In the future there could be problems that arise from updating the app to support newer software version and maybe slow the app down for some users.

1. Security (BK)

GasUP will have user accounts which will be verified by username, password and an email. These accounts will be encrypted on our end. Other login methods will be verified through third party providers, such as Google, Facebook, Twitter, etc. All saved addresses, searched locations, and MPG information are stored on the users devices and therefore don’t require any additional security.

1. Processes (SD)
   1. *Login* 
      1. The user will either create an account or use GasUp as a guest. To create an account the user will need to supply a username, email, and password. The password must have at least one number and capitalized letter to enhance account security. Additionally, a confirmation email will be sent where they will have to verify their account.
   2. *Locate a gas station*
      1. This tab will allow the user to input a zip code or use their device’s location and a desired radius. The app will then output the nearest gas stations according to the user’s data. They can view the outputted gas stations in either a list form or on a map. Additionally, they can filter the list/map on price and reviews.
   3. *Add a MPG entry*
      1. This tab will allow the user to input the number of miles driven and the number of gallons purchased. The app will then output the calculated miles per gallon.
   4. *View Miles Per Gallon history*
      1. On this tab the user will be able to view their past mpg entries in either a list view or graph view. In the list view they will be able to filter the entries by date, mpg, and miles driven. On the graph view they can choose to view their data as a histogram, bar graph, or a dot plot.

# Class Diagram (SD)

Create a class diagram. The Class Diagram should contain all of the system objects, their attributes, and any known methods. This diagram may be included as a separate file – it does not need to be inserted into this Word document.

# Use Case Diagram (BK)

Create a Use Case Diagram for all of the "uses" of your system. This diagram may be included as a separate file – it does not need to be inserted into this Word document.

# Use Case Scenarios (BK)

Create a full description Use Case Scenario (detailed descriptions) for each use case of the system. This intermediate scenario should include an enumerated list of steps involved in the activity as well as any exception conditions.

# System Sequence Charts (SD)

For each Use Case Scenario, provide a sequence diagram. Use your class diagram, use case diagram and scenarios to create the corresponding Sequence Diagram.