**Functional Specification:**

The essential functional element of our app will be of course a phone. To be more exact any smartphone with a camera capable of scanning QR codes. You will be required to have a valid internet connection when trying to scan an item to access the database for all the information. When trying to add a new item to a company’s product list or contribute to the transportation paths for a certain item, the phone will need to have your geolocation which will also require an internet connection.

**Functional Requirements Tables**

*Table 1* refers to the functional requirements for the app on the company owner side. It shows all the functions that we would either want or need in our project. Majority of these features will be a necessity but there are some such as ‘*FR112*’ which would be optional. This means if during the app development phase, we start running out of time this would be one of the first things to get removed to realistically finish the project on time.

*Table 2* is the same functional table except it is for functions that the customers will use. Here we have more things that we could potentially remove as these functions are more for the sake of clarity.

*Table 3* shows the functional requirements for the website that we will create. Most of the functionality of the website will only be used by admins of the website and the company owners to sort out their company related issues.

**Use Case Diagrams**

The use case diagrams show us how the users will interact with the system and the different cases that can happen. *Figure 1* and *figure 2* shows the use case for the mobile app that we have split up into consumer and company owner respectively.

As for *figure 3*, that is the use case diagram for the website split into threeseparate sections. We don’t have many if not any plans for adding the customers to the website since it would just overcomplicate the system, however we have still added it just in case we want to change our minds along the way.

**Class Diagram**

*Figure 4* displays the full class diagram that we plan to implement into our program, showing all the links between the classes. It includes all the variables that a certain class will use along with the functions related. This will most likely be updated as we go along once we run into issues during the actual development phase, however it does provide us with a decent starting point. It also points out the relations between the classes which could be beneficial once building the databases.

**Sequence Diagrams**

The sequence diagrams show how the different objects interact with each other. In *figure 5* and *figure 6* you can see the different interactions between them in certain scenarios. It also gives us a good idea of what functions are being used together and what calls might be made between them.

**State Diagrams**

The state diagrams show the different states that the system can be in depending on the transition occurring. In *figure 7* and *figure 8* it becomes clear how the system reacts to different inputs/button presses and what kind of functions get called when certain interactions are made.

**Collaboration Diagrams**

Collaboration diagrams are a combination of information from class, sequence and use case diagrams. They describe the relationship or collaboration between classes/objects and the related functions with them. In *figure 9* and *figure 10* you can see the collaboration between objects when a customer (*figure 9)* or company owner (*figure 10)* interacts with the system.