**Team Management and Process**

The group number and the list of team members:

Group number: 22

Team members: Harry Tennent, Stuart Davis, Animesh, Shubh Garg, Osman Poonawala, Shitin.

Team members’ roles and their contributions to the project:

Harry Tennent:

Stuart Davis:

Animesh:

Shubh Garg:

Osman Poonawala:

Shitin:

List of meetings and who attended:

31/1/19:

Discussed basic requirements.

Draft use case diagrams.

All members of the team were present.

5/2/19:

Reviewed and added to our functional and non-functional requirements, and the use case diagram.

Thought of the classes needed for the class diagram.

12/2/19:

Harry, Stuart, Shubh, and Shitin present.

Drew a first draft of the class diagram.

19/2/19:

Completed class diagram.

All members of the team were present.

26/2/19:

All members of the team were present.

Created a state machine diagram.

5/3/19:

Began work on the sequence diagram.

Added annotations to the state machine diagram.

All members of the team were present.

12/3/19:

Updated the sequence diagram.

Created a test plan.

All members of the team were present.

19/3/19:

Refactored the class diagram to use a design pattern.

Decided on the tasks each team member should do for the report.

Harry, Stuart, Shubh, Shitin, and Animesh present.

26/3/19:

List of software tools used and comments about the group experience with them:

One of the main software tools that we used as a group was draw.io. This is an online diagram drawing tool that we used extensively to make all of the diagrams for this project. It had a number of UML-specific shapes, such as boxes for class diagrams and different types of arrows for class diagrams, sequence diagrams, and state machine diagrams. This was useful and made it easier to create and edit a diagram. However, the interface made the site hard to use because it was hard to connect arrows to boxes so that parts of the diagram could be moved and the arrows would remain connected. One advantage to draw.io, however, was that we could save our diagrams directly to the group’s GitHub repository.

GitHub was very useful for this project. We used a repository to keep all of the project files together and to track what we had done when. All of the team members were added as contributors and then we added and updated files as we needed to. The GitHub site makes it very easy to see who added files and what the current state of the project is. Not only did we keep the diagrams we created on the repository, but we also created a meeting record to track who attended the meetings each week, and files to keep a record of what tasks need to be done by who before the next meeting.

**Class Diagram**

For this class diagram, we used the State design pattern. This is because the shopping basket is in a number of states during the use of the application. Either the basket is empty, has some number of items in it, or it is “completed” and the user is proceeding to checkout and is about to turn the basket into an order. As a result, we created an abstract basket class which is extended by the 3 different states. When a user uses the QR code scanner to add an item to their basket, the basket moves from the empty state to the normal state. Once the user is ready to checkout, an order can be created which contains data such as the order status, the date created, and date shipped.