5. Documentation

Neil M. Patel's contribution:

1. Program Code: I assisted Brice and Vineet with the user interface/backend of the web application, mainly through implementing the rotation feature of the application (using the "A" and "S" arrow keys for counterclockwise and clockwise rotation, respectively). I also debugged and corrected errors in implementing partial states, that is, allowing the robot to move in a combination of the cardinal directions (e.g. northwest, southeast, etc.).
2. Unit Testing: After implementing the above mentioned features of the web application, I performed unit tests on them by running the application, pressing the various keys, and checking the MongoDB database to ensure that the appropriate collection was populated with corresponding data values.
3. Webpage Design: Contributions have been listed in item 1 ("Program Code") above.
4. Integration Testing: Since our first demo featured a web application designed to be run on a desktop/laptop only, no cross-platform issues were encountered. Moving forward, however, we will need to design an application to be run on a tablet.
6. Debugging: Contributions have been listed in the previous items.
7. Documentation: In designing the above mentioned features of the web application, I heavily commented my code for ease of understanding. I also contributed an equal portion of the weekly documentation, explaining concepts such as the proposed web application user interface and system architecture plans.
8. Database: As mentioned above, I worked with other members of my team to determine the database schema and to perform database manipulations, such as pushing rotational and translational movement actions to it.
10. Brochure: As a member of the web application sub-team, I created the initial draft of our section of the brochure.
12. Project Management:
    - Assisted during meetings by providing updates from my sub-team and occasionally delegating tasks to other team members.

Vineet Sepaha's contribution:

1. Program code: I worked with Brice and Neil on the backend of the web application. We used Flask and HTML to implement our web application. I mainly worked on the multi-directional feature such as going northwest, southwest and etc.
2. Unit Testing: Our web application sub group tested our implementation to see if it was able to push and retrieve items from the Mongo database.
3. Webpage Design: I designed the aesthetics of the webpage which was implemented in HTML and CSS.
4. Integration Testing: There was not a large amount of integration needed for the web application. As mentioned above, we had to make sure it was integrated with the database.
5. Debugging: I assisted Brice in working through some of the problems which occurred while using the rotation and speed buttons on the web application. Some of the errors

we encountered were due to the improper placements of functions and the compatibility of the database with the robot.

6. Documentation: I have contributed weekly to the documentation of report 1 and 2, working on things such as use cases, sequence diagrams and the design and functionality of the web application.
7. Database: As previously mentioned, I worked on pushing the cardinal directions into the database and other aspects along with my web application team members.
8. Powerpoint: I worked with Jon in creating the powerpoint to give a general overview of the software project. The main focus here was to show who it was for and what it does.
9. Brochure: I worked with Neil in creating the our subgroup's portion of the brochure.
10. Project Management: I made sure there was good communication within our subteam along with out subteams in our software group. I also worked on making sure that our subteam was up to date with the rest of the team in terms of getting things done.

Cedric Blake's Contribution:

1. Robot communication: Developed the code that allows the robot to send and receive data from the database.
2. Web App communication: Tested and Debugged the code used for the web app to communication with the database
3. Debugged cross-platform issues with ROSE cpu which included changing code so the implementation can be ran on the Nvidia Jetson TK1
4. Designed an Implicit schema for each collection such that the robot can communicate with the web app without ambiguity in type conversions of the variables sent.
5. Robot: assisted in soldering wires, robot frame assembly and testing arduino motors.
6. Documentation work:
    a. Enumerated Functional Requirements
    b. Functional Requirement Specifications (parts "a" to "d")
    c. Systems Operations Contract
    d. Class Diagrams and Interface Specification (parts "a" to "c")
    e. Server side Technical documentation
    f. Server side README documentation
7. Brochure: Worked on the database segment of the Brochure
8. Powerpoint slides: contributed to data collection for verifying that our project is more cost effective than hiring a waiter
9. Project Manager: Worked as project manager for the team. The position of project manager has consisted of
    a. Setting up both weekly meeting time and additional meeting times for the entire team to discuss and plan how the work of each week would get done.
    b. Planning topics of discussion for each team meeting
    c. Handling communication between each subteam so that each team member is up to date with the team's progress and plan of work
    d. Keeping track of the progress that each subteam has made throughout the course of each week

Jonathan Cheng's Contribution

1. Robot communication: assisted Cedric in working on database communication between the robot and the database and with the coding.
2. Integration/Data Collection: wrote code and ran test simulations with a test database to retrieve information from the database that the robot would push information to. Note that these are test runs which were utilized before actual integration between webapp, database, and robot.
3. Powerpoint: worked with Vineet in writing up a brief and concise overview of our project, what our product does, and who it is meant for. Collaborated with other team members such as Cedric to run through and verify that the information posted demonstrates a concise and general overview of our project.
4. Brochure: contributed to the write-up of the "database" section, providing readable information to anyone who is unfamiliar with the technicalities of our usage of the database.
5. Project Management - Collaboration: worked with webapp subteam to ensure that our implementations between the webapp and its back-end were communicating properly with the database, and at the same time allowed the server to also send and retrieve information from the database. I volunteered to be record keeper in what we discussed during our weekly meetings, and have been providing "meeting minutes" for all members to see on our Google Drive so that everyone knows what they're doing. This provided a central and organized space for all members to refer back to should they need a reminder for their responsibilities for upcoming deadlines regarding documentation and demos.
6. Documentation: contributed to some comments in the webapp side of the project, which was involved in various testing with our test database on mLabs. Some of the features will be more visible in a future demo.

Srihari Chekuri's Contribution:

1. Computer Vision: Tried various libraries (matlab and opencv) and different algorithms to detect objects and people (brute force, SVM, haar cascades). In matlab, I used SVM, by finding the surf features of an object(coke can) and comparing those with each frame from the camera feed. This worked very well for comparing between images but there were problems while updating frames in the webcam feed. Next, I tried brute force algorithms in python by getting critical points of a soda can and comparing with an image with multiple soda cans. This didn't work really well because a lot of critical points matched up with other soda cans as well. Next, I tried making haar cascades to determine objects. I downloaded 2,000 negative images, superimposed a positive image on all those negative images, and then ran the opencv cascade training command. The resulting cascade file wasn't as good as expected, but the method with haar cascades was very good for detecting facial features. Next, I found an open source robotics computer vision library. I downloaded the necessary dependencies and added various objects to the testing application to detect all the objects I needed. The application uses

SVM. The application finds the SURF features of the object you choose and compares them with the SURF features of each frame to detect the object. This is very similar to the code I tested in matlab, but it was a lot faster. Since, there was a 1 second delay, I am trying to make it faster by writing my own c++ application that can detect a limited number of objects in a more efficient way. Currently, the application can detect the surf features on an object and compare them with other features on other images to detect objects. I am also testing neural networks (CAFFE), higher quality haarcascades (more testing phases in training) and more efficient SVM algorithms to come up with the most efficient and scalable way to detect objects.

2. Localization: In order for the robot to pick up objects and interact with the environment, it must be able to detect objects and feature spaces. We have a couple of different objects to detect this time around. Initially, we decided to try doing object detection in a way in which it can be used to detect anything. The first object we tried to detect was chilitags. These tags are similar to enlarged barcodes and can be detected very quickly and efficiently. We started with these because they are very useful for a particle filter that we will be implementing for our final demo for localization. If the object detection using the above computer vision methods are not efficient enough on the jetson-tk1, then we will use chilitags to detect all the objects that we need to detect.
3. Text to Speech: To give the robot a voice, for example, when asking customers to move or telling them to enjoy their food, I was testing some text to speech libraries. I used festival to make a c++ application to make the robot announce certain messages. I also tested the pyttsx library for python and made an application to convert text to speech.
4. Robot: I worked with Ajay to design and build the robot and attaching various electronics to the robot. I helped parse the encoder values for the robot. I helped with making custom cut metal and with assembling the robot. Made sure the circuitry on the robot was correctly done and made sure the arduino's, USB hub and the jetson tk-1 got the appropriate voltage.
5. Unit Testing:I wrote unit tests for each type of object detection that was tried. There is a unit test for matlab object detection using SVM. There is unit test for SVM in c++. There is a unit test using the robot object detection application using SVM in c++. There is also a unit test for facial feature detection in python.
6. Debugging: I worked on debugging various problems I ran into while checking the latency behind object detection using various methods.
7. Brochure: I designed the brochure and did the computer vision segment.
8. Documentation Work: I worked on the Problem Statement, consumer experience, use cases, interface layout and design, mathematical models, network protocols, global control flow, algorithms, data structures, hardware requirements and design of tests on reports one and two.
9. Project Management: I helped set goals for presentation and allocated work to be shown before the deadline. Assisted Ajay on some parts of building the robot.

Ajay Srivastava's Contribution:

1. Robot: The ROSE system was designed to be a mobile system of integrated components designed to replace a waiter. In order to accomplish this task, I decided to design a system that would be able to accomplish these tasks, as well as have potential for future innovations. The design Srihari and I chose for the robot was an omnidirectional chassis, along with a 6 Degree of Freedom (DOF) arm, that would allow us to move around in any direction, as well as pick up various different objects. After mechanical design and construction was complete, we had to consider how to approach the problem of complete control of our system. After delving deeper and realizing how much would go into software for control of our system, Srihari and I approached the problem piece by piece and determined how we would go about solving our task. The very first thing we had to deal with was inter-communication. Not to get too much into hardware, but we have multiple arduino microcontrollers along with adafruit motorshields, and multiple sensors sending back information to our central computer, which is the Nvidia TK1. We had to communicate between these devices, so we set up our robot as a distributed system, which communicates with each device using the serial TTY protocol. I developped this serial system to communicate with the systems. After this system was set up, we were able to successfully move the motors and read all the sensor values. The next item we dealt with was localization, and we decided on using a particle filter, a relatively newer algorithm that utilizes multiple simulated robot objects in order to determine the location of ours in reality. I developed this particle filter in order to help us localize. Our arm also needed to be controlled in some way. The inverse kinematics algorithm allows us to autonomously control our arm and tell it to go to a certain location. This is still under development, however this is what I have been working on recently. The integration of the webapp component also had to be dealt with, so all of the integration between the webapp, database, and robot was done by me.
2. Computer Vision: I worked with Srihari in implementing object detection using chilitags. The chilitags will help identify the object designated with the tag.
3. Integration: I integrated the robot with the database and sent the robot encoder values and robot status information to the database so it can be shown on the manager interface of the web application. I helped the web application subgroup and databases subgroup integrate their parts with the robot. I helped the web app subgroup with sending key inputs to the database and the robot.
4. Unit Tests: I made unit tests to show robot status such as battery, connection status' and etc. I made tests to move the base of the robot forward, backwards, left and right. I made unit tests to move the robot to rotate clockwise and counterclockwise. I made unit tests to move the robot arm in various angles. I built a sliding bar user interface for the user to move the arm in real time by sliding bars. I helped everyone make test cases to integrate the web application, database and the robot.
5. Debugging: I did the debugging for the web app, database and the robot. I fixed the problem of the webapp causing the the robot to move even after closing the application. I fixed the problem with the robot not receiving the proper values from the web application. The integration of all items required multiple items to communicate with each

other, and I made sure that all items had the same protocols and standards to work with each other.

6. Documentation Work: I formatted, edited and assisted everyone with their documentation each week. I also worked on the documentation with Srihari each week. I was responsible for submitting the final drafts after reading the reports over.
7. Brochure: I worked on the robot sub section of the brochure.
8. Project Management: I set the final goals and milestones each week and assigned work to the subgroups, in terms of actual work needed to be done to accomplish our final product.

Brice Howard's Contribution:

1. Program Code: I (with Neil and Vineet) developed the code responsible for hosting the web application and the web application itself, written in Python and HTML respectively. I implemented buttons for cardinal movement and functions for affecting the speed of the robot, later allowing both to be controlled by keyboard buttons and making the process asynchronous (that is , without the need for a page refresh). I created a set of Developer Tools, the visibility of which could be toggled, that allows a user to see the button they press, delete the existing items in the database, and populate the database with default values (ex. 0 speed). I wrote the code and HTML responsible for displaying the items within the database on the web application, and the code responsible for pushing data to the database.
2. Unit Testing: Using a database hosted on MLab rather than on another computer, I tested the items listed above for complete functionality.
3. Web Page Design: As mentioned above, the web application was written in HTML. No functional aspects of the web application have been given any aesthetic attention, as the current functional web application was meant only for driving and testing purposes, as well as to serve later as a "manual override" for the robot. However, simple aspects such as where the information is displayed, where the buttons are, the ability to hide the developer tools, and the visibility of certain other tags (like the number-type input for speed) were decided by me.
4. Integration and Integration Testing: The only integration thus far has been on a computer, and making sure that such occurs properly was part of the unit testing of the web application noted above.
5. Debugging: I fixed an error in the web application code that resulted in a directional value staying in the database, as opposed to returning to the default value of "STOP", and a similar error that occurred with the rotation value of the arm. I modified the web application code so that the necessary items would be inserted into the database if they were not already, and would be modified if they were. I fixed an error that would cause the web application to fail to load if there were no documents in the database. I fixed a bug that caused the web application not to read or display the values in the database. I fixed an error that caused the web application to display the overall change in speed that

the user has requested over the time spent on the page, rather than displaying the speed within the database.

6. Documentation:
    a. Code Commenting: As the above mentioned code was to be a simple piece for testing, I was irresponsible in my code commenting, and will make absolute sure to comment heavily in the future.
    b. Report #1: I was responsible for writing a portion of the technical terms and a portion of the user stories. I worked with others to create the domain model writeup, and created Figure 15 of Report 1 (“The Domain Model”).
    c. Report #2: I assembled Figure 3.1 (“UML Package Diagram”), in addition to the writeup of the Subsystems to Hardware mapping.
7. Project Management: I created a group account on MLab for testing purposes. I helped to define a structure for the presentation.