PROJECT SUMMARY OF ROBOCON SYSTEM CONCEPT

-Pavneet Kaur Mukar
9614873572

RoboCon consists of a variable number of robots which form a convoy and follow a leader robot to a specified destination. RoboCon will provide a demonstration and experimentation platform for CS568 System image processing and artificial intelligence algorithms.
There are many different components of this system and different approaches were followed to handle them. They are described as follows:

1. **Firstly**, we create a module of the project, so that all the working could be recorded in one area space. It makes things easy to work and can be shared between group members. Next comes the SRS creation.
2. **INTRODUCTION**: It briefly explains what the system is supposed to do. There are different attributes in it like **Heading** was used to make a heading in the SRS, and one can alter the system. I defined these terms separately in Features so that a different view can be seen easily and managing the things is done effectively. Introduction had various sub-sections which were like Purpose, Scope, Definition and Acronyms, References and so on. All these were artifact type "feature" and text mode. All these were collaborated under "Features and Vision" section of the project. In the module, these artifacts were imported under headings. One could alter them by double clicking in the module section or by going in feature folder and then altering it. Definitions and acronyms terms were defined in the "glossary folder" of the project. Linking them was easy as all the terms related of same type are inside one single folder. I made this separation for easier interlinking.

3. **OVERALL DESCRIPTION:**
   - **Product Perspective:** It was a feature attribute which simply explained the perspective of the RoboCon.
   - **Usage Scenarios**: They are of further two types: experimentation and demonstration. It was defined under features and vision section, as they were simply the features of the system.
   - **Next came the USE CASEs**: I made the use cases of both experimentation and demonstration using GLIFFY SOFTWARE. Constructing them was easy as the tools which this software provides are accessible and defined properly. The use cases formed involved 3 different actors and showed how the actions take place and which actors are responsible for which actions. Main actors I used were: Developer, Tester, Manager.
   - **Stakeholders Goals**: The main task is to identify who the stakeholders are and prioritize them accordingly. Then next step is to record the main goals of the system. It is recorded using Stakeholder Goals property and under Stakeholders request system. The main reason behind doing this is because they are of high priority. Thus, separating them from overall feature and assigning priority is possible. Moreover, traceability becomes easy to create. I also added specific tags so that different views could be seen.

- **Stakeholders Requirements:** They were defined under the Software Requirements folder as “Requirement attribute”. It basically explained in detail the requirements which the system should have according to the stakeholders. The traceability was made between these requirements and the main goals using “satisfied by” and “satisfies” link.

4. **SPECIFIC REQUIREMENTS:** It explained the main requirements of the system. It had various parts.
   a. **System Description:** It explained the overall system requirements of hardware, software and administrative control unit(ACU). These were all defined in Software Requirements and a sub folder was made individually for all these requirements. These are mandatory for the system to have, without them it is not clearly possible to construct the system. They can be considered as constraints. These attributes were then imported in the module. It also explained the 3 different operation modes which the system has. They, being convoy formation, movement and recharge mode. They were described under software requirement folder and were defined as requirement attributes only.
   b. **Activity Diagrams:** I created the activity diagrams using the same GLIFFY SOFTWARE. Activity diagram basically depicts all the activity flow in different areas of the system. I made 3 different activity diagrams, one for each convoy formation, movement and recharge operation modes. It visually depicted step by step procedure of the occurrences of activities in the different modes of system and under which section of the system they take place. All the diagrams were placed under Activity diagram folder in the system. Also, they were of “diagram” attribute type.
   c. **Class Diagrams:** They are the diagrams which show the overall description of how different classes are linked. I made 2 class diagrams, one for each architecture and another for all the routes in the system. It showed the cardinality, dependencies of classes on one another as well as inheritance properties [if existed]. It is easy to understand the flow of the system if class diagrams are made properly. The architecture class diagram showed interlinking between classes like, ACU, RCU, actuators, iRobot, sensors and many more. On the other hand, the route class diagram shows different routes taken by robots, obstacle occurrences and solutions. All the diagrams were placed under Class diagram folder in the system. Also, they were of “diagram” attribute type.
   d. **Sequence Diagram:** They show the messages exchanged between different components during execution of a scenario. I tweaked the system as bit and showed the flow from *Right to Left* as opposed to *L to R.* The flow can be understood as they are occurring in steps and can be easily differentiated as to which step occurs first. I made 3 different diagrams, one for each convoy formation, movement and recharge modes of the system. I feel making this change made better understanding due to sequential flow in the system. Moreover, I could show how different conditions affected different areas of the system. It was visually possible to make that difference out. All the diagrams were placed under Sequence diagram folder in the system. Also, they were of “diagram” attribute type.

5. **NON-FUNTIONAL REQUIREMENTS:** Lastly, I specified additional quality characteristics for the product that is important to either the customers or the developers. Some of which are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. I also clarified the relative preferences for various attributes.
6. I made various tags and corresponding views of the system. Traceability is also shown. And different kinds of attribute types like text, diagram, collaborations etc were done. Thus completing my SRS documentation efficiently.