**Abstract**

Over the last decade, the technology of robotics has been rapidly improved to serve safety and reliability in our digital world. We came a long way after developing our first robot in 1954. This paper aims to describe the design of our rescue robot system that we built, we did include software specification , a complete design and codes .Our main was to help our rescue team in extreme situations where human involvement is too risky. We considered water scenarios and land scenarios as our prime base. We  gave our robot sudden goals to achieve in these scenarios like finding the target in both of the scenarios. We also gave challenges in the given territory. We ended up implementing a reliable and compact base for our rescue robot.

**1.** **Motivation (Abdul-Azeez Olanlokun)**:

Natural disasters like that of the Tsunami (1) are uncontrollable disasters that can occur at any time without prior notice. It is then important to have emergency rescue to help save lives and properties. Such necessity motivated us to bring about a technological innovation that can serve as an emergency rescue in this kind of situation. As mentioned earlier, the Tsunami (1) “is a series of enormous ocean wave caused by earthquakes underwater landslides, volcanic eruptions, or asteroids.”

We then thought about a rescue robot that can be used in such emergency, not only that it could move on land, but also drive on/in water and in dangerous areas, where humans cant access easily. We then designed ResQ robot that drives autonomously on land and on water, and that can detect and overcome obstacles on land, and also detect humans in water. Our robot serves as a first aid rescue support before human intervention.

During the stages of building our robot, we faced many challenges, in terms of obstacle obstructions, but with much diligent research by our team, we were able to overcome these challenges and difficulties in making our robot work perfectly as we want. Also, we learnt a lot that will surely aid us in future projects and how to successfully achieve a working project.

**2.** **SysML DIAGRAMS (Abdul-Azeez Olanlokun):**

To design our robot to satisfy all engineering principles, we made use of SysML diagrams learned from our course of study in Systems Engineering, such as Requirement diagram, Constraint diagram, Block diagram, Use Case, Sequence/Activity diagrams and lastly system architecture, that helped us model our robot, as this kind of project is a complex system that requires a model that supports analysis, specification, design, verification and validation.

**2.1.** **REQUIREMENT DIAGRAM (Abdul-Azeez Olanlokun):**

The Requirement Diagram (fig 1) shows a graphical representation of all our project requirements. With this diagram, we were able to show a textual description of our requirements with priorities in functionalities. As we know requirements takes a huge part in any system development, so it was important we analysed it for the success of our project.

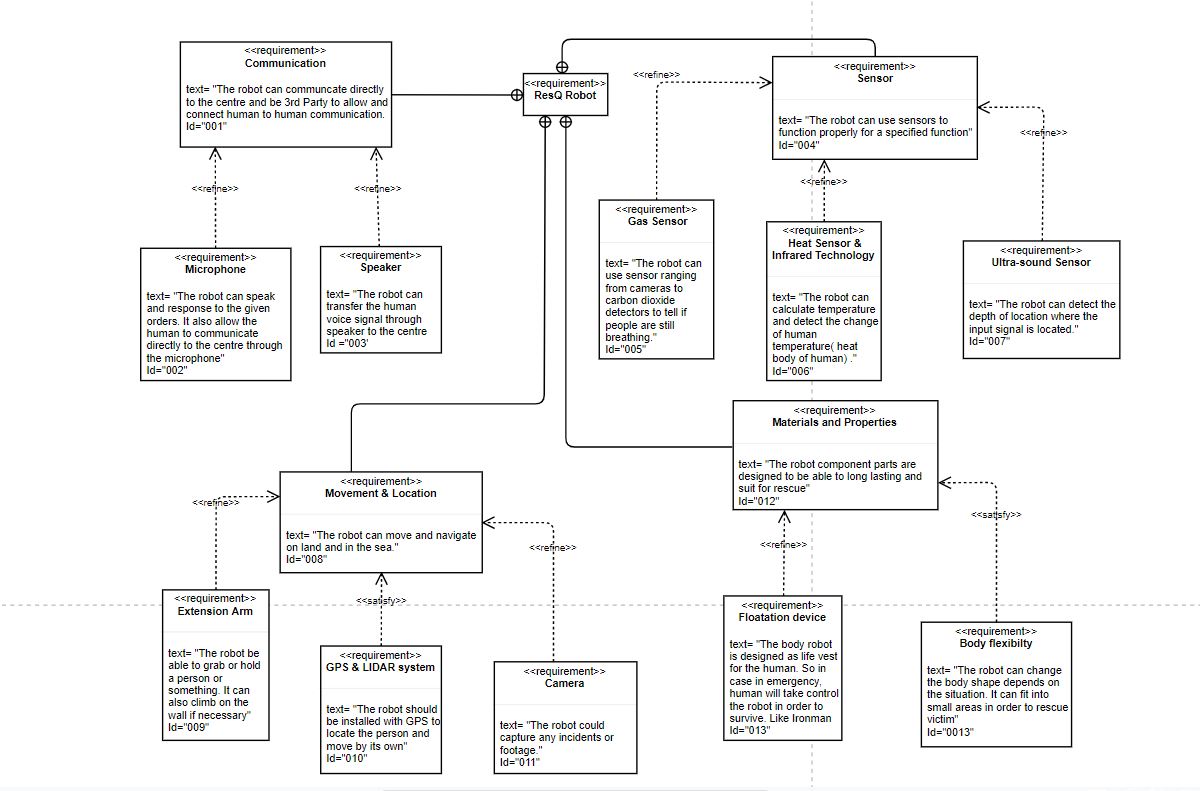
In this requirement diagram, as shown in fig 1, our ResQ Robot has range of requirements connected to one another with high, medium and low priorities, which are noted with the word <<refine>>. The must have of our robot are as follow:

**The communication:** which is refine with two communication channels (Microphone and Speaker), shows how the robot is able to communicate directly to the admin centre through the two-communication channels.

**Movement and Location:** This is the means to which our robot can navigate around, and it is also refine with camera and GPS system.

**Materials and Properties:** Our Robot parts are designed to withstand wear and tear, and also be aerodynamic. It is also refined with floatation device and body flexibility.

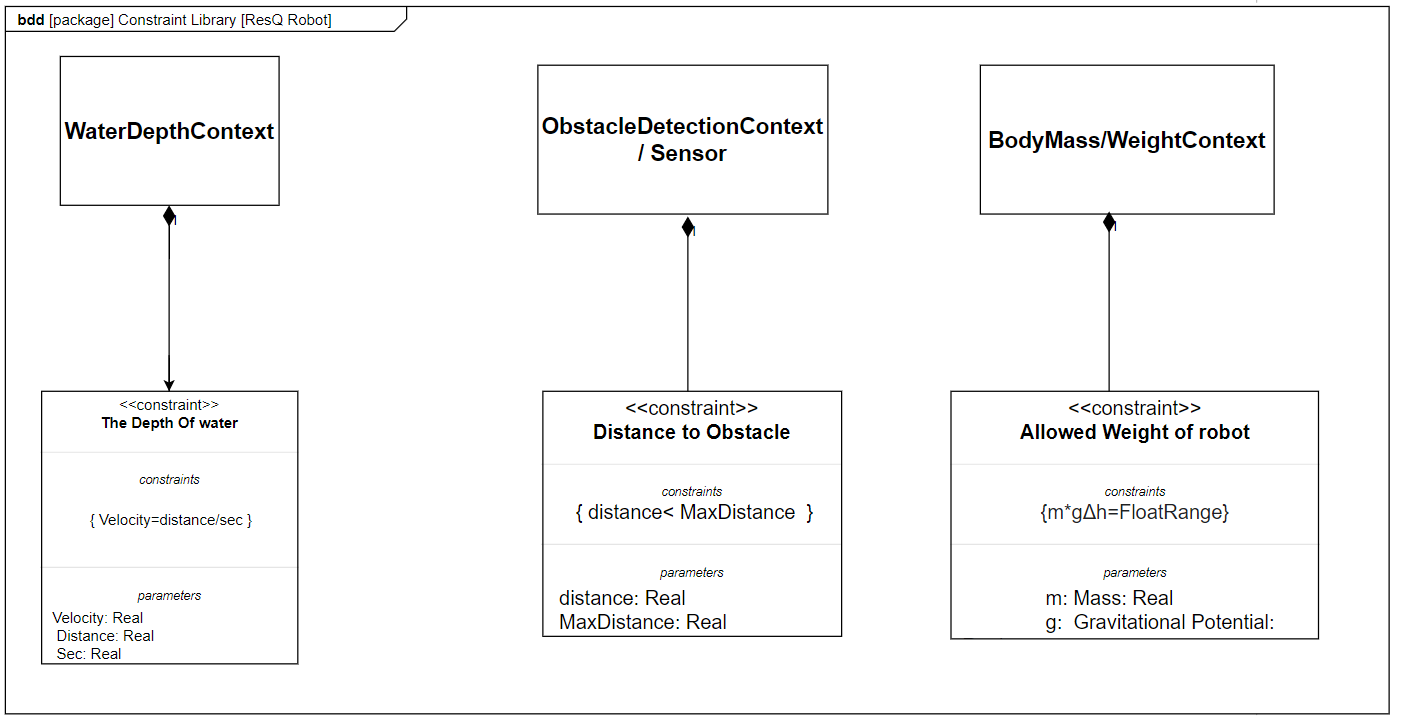
**Sensor:** Our robot needs sensors to perform its design purposes. It is refined with Gas, Heat & infrared and Ultra-Sonic sensors.

Fig. 1 Requirement Diagram

**2.2.** **CONSTRAINT/CONTEXT DIAGRAM (Abdul-Azeez Olanlokun):**

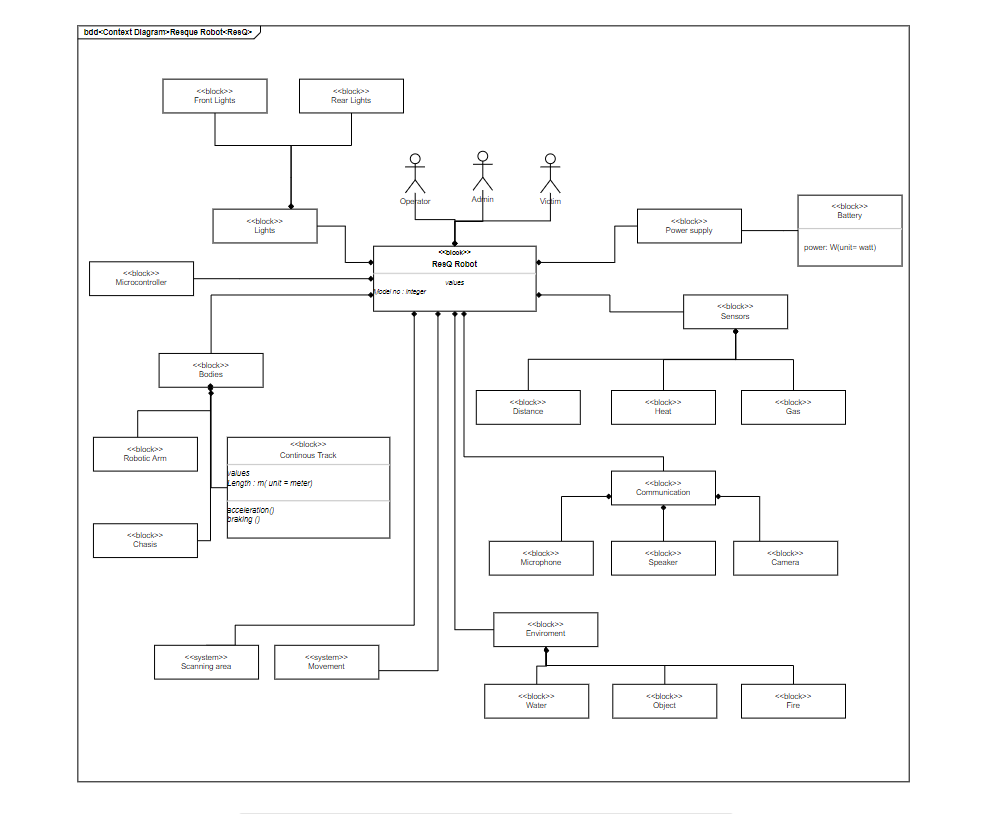
The constraint diagram (Fig. 2) basically specifies a network of constraints that shows mathematical expressions, these constraints are mostly from physical properties of a system. Our constraint diagram shows a graphical representation of mathematical expressions of all the constraints in our robot in a constraintBlock.

The properties of our robot constraints are: The depth of water constraint, Distance to obstacle constraint and Allowed weight of robot constraint.

Fig 2. Constarint Diagram

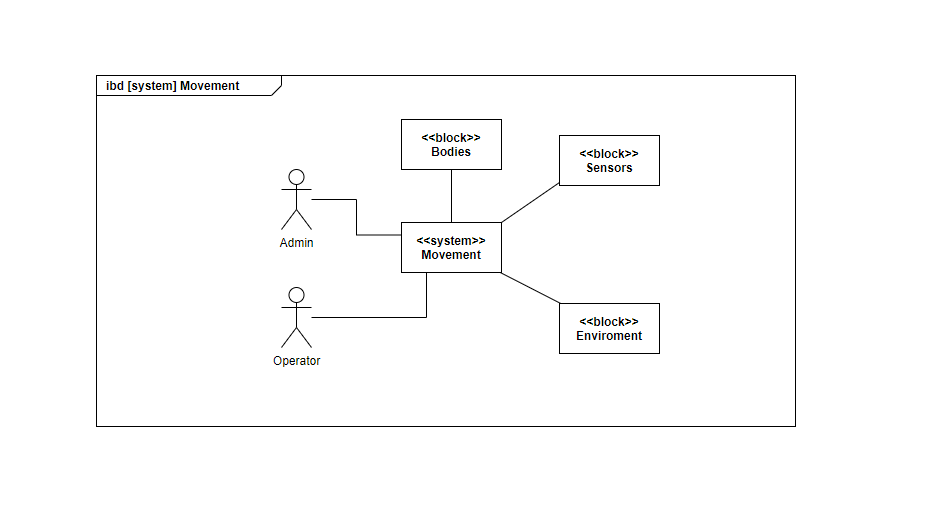
**2.3.** **BLOCK DIAGRAM (Abdul-Azeez Olanlokun):**

The Block Diagram (Fig 3), shows the modular unit of our system, which abridges the contents such as the operations, attributes and constraints of our system. The block diagram shows the structure of our robotic system, from the physical to the logical attributes.

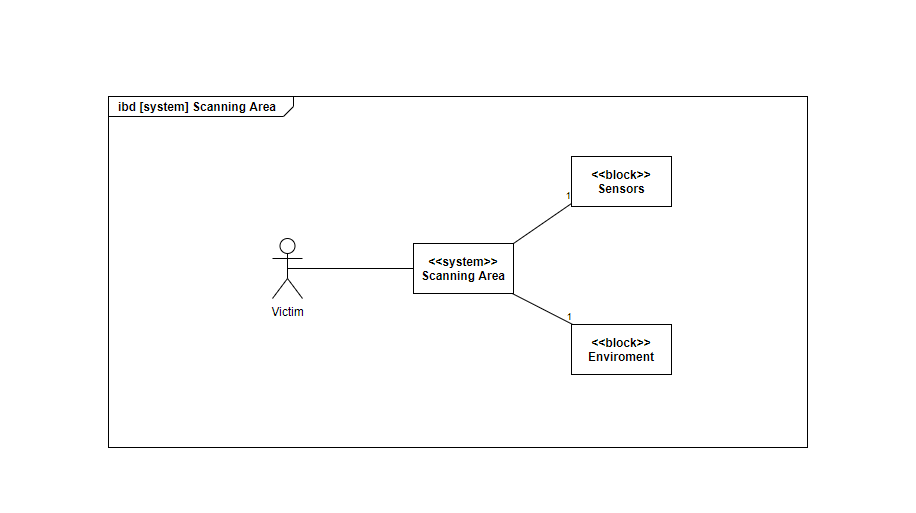
Fig. 3. Block Diagram

**2.4.** **INTERNAL BLOCK DIAGRAM (Abdul-Azeez Olanlokun):**

The Internal Block Diagram (Fig. 4), shows a graphical view of the internal structure of a block in our system. This shows all the main decomposition of our robot in one single internal block diagram. This decomposition shows the <<system>> Movement which other external features act on it. The external features are: The Admin/Operator, bodies, Sensors and Environment.

Fig. 4. Internal Block Diagram

**The Scanning Area (IBD):**

The Scanning area (Fig.5) is a segment of the internal block diagram, showing how our robot navigates around the environment and searches for the victim.  Fig. 5. Ibd [system] Sca

**2.5.** **ACTIVITY DIAGRAM (Amit Chakma):**

In the activity diagram [Fig 6] we can see the activities of the robot in water . Here the robot is activated by the rescue members and the robot dives in to search for the victim . When the robot detects the dummy it sends the exact location to the safety guard . After the robot finishes the mission it returns to its charging station for the next rescue .

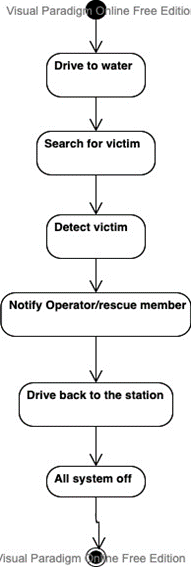


Fig. 6. Activity Scanning

**2.6.** **USE CASE DIAGRAM (Amit Chakma):**

In the use case diagram [Fig 7] can see the potential scenarios of our rescue robot. We have an operator who is responsible for controlling the robot.

The robot can move autonomously and can also be controlled by the operator. We will also get continuous GPS location from the robot in any scenarios. We included a camera which is capable of taking videos and photos in extreme conditions .

We also have an environment that is responsible for feeding datas to our sensors .Using our sensors we can check the body temperature of the victim . We will be also able to detect that our victim is breathing using a CO2 detector and at last we also have

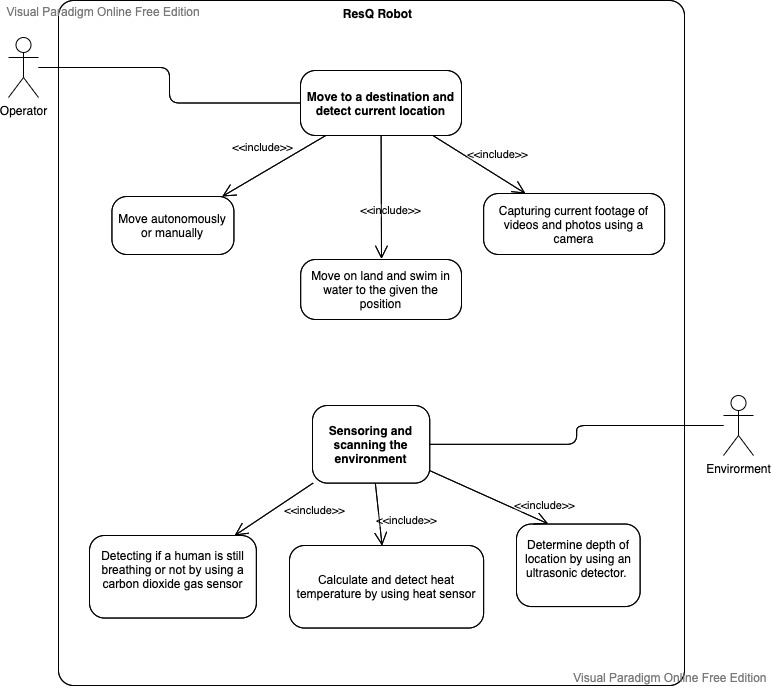
an ultrasonic sensor to measure the distance or depth .

Fig. 7. Use Case Diagram

**2.7.** **SYSTEM ARCHITECHTURE DIAGRAM (Amit Chakma):**

For the system architecture [Fig 8] we have used *Five layer architecture* which is very commonin real time systems. In a layered pattern we can work on each of the layers independently and update as required.

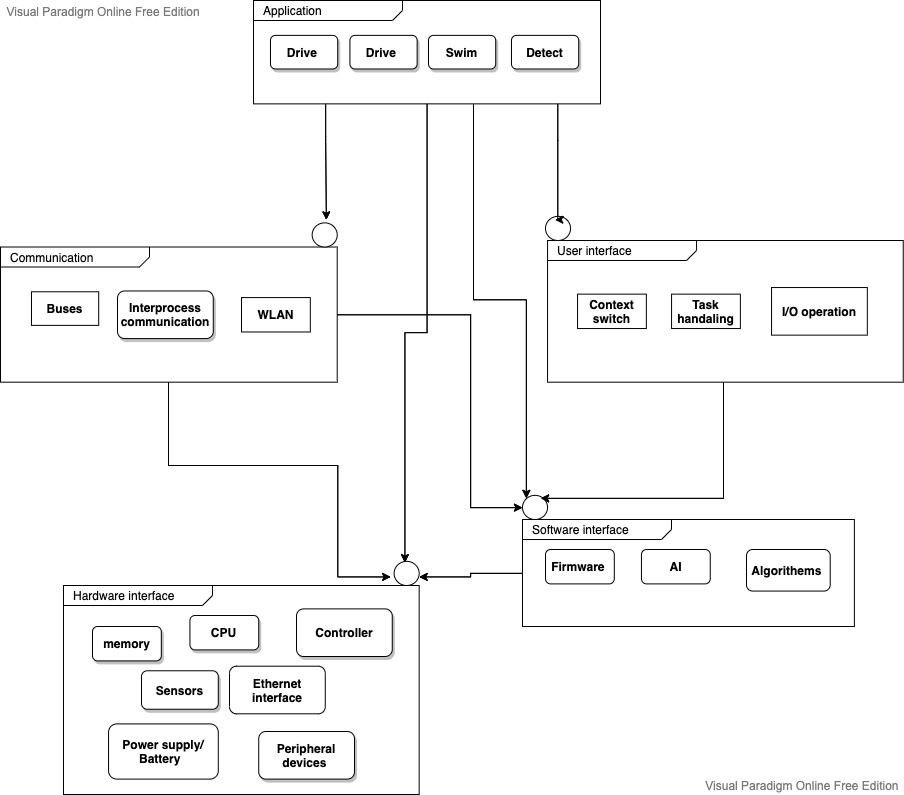
Application layer contains the application-level classes . In the application layer we can see the common tasks handled by the robot from things like diving on water , driving on land and water , swimming , object detection.

User interface layer contains classes specific to the user interface.I/O operations help transfer information between computer main memory and the outside world. Context switching used to switch between processes to handle multiple tasks .

Communication domain contains classes necessary to transport data ,commands and events among the object. Here we have Buses which help to transfer digital signals to transfer data rapidly .Interprocess communication used for exchanging data between multiple threads in one or more programs. The Processes may be running on multiple computers connected by a WLAN.

Hardware interface layer provides classes that represent devices and interfaces. We have our memory , CPU , peripheral devices , sensors , controller and ethernet interface.

Software interface layer provides the classes to manage threads and memory and other system services. Here we have firmware , AI  and algorithms which will provide instructions for our hardware devices.

Fig. 8. System Architecture