Optimum Path Generator

By

GC Abdul Wadood

GC Hamza Nawaz

GC Mureed Ullah

GC Waqas Zahid Abbasi

Submitted to the Faculty of Department of Computer Software Engineering National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering, July 2020

In the name of Allah, the Most Beneficent, the Most Merciful

ABSTRACT

To study/observe a given terrain, it is imminent to have a map of that environment. There may arise a need to study of a place having no previous geographical data or such a place is unreachable for a human being. Such cases can be seen in survey of an environment, tunnel system and military operation. We have made a compact size robot namely OPG Bot that can solve this issue and provide a 2-D map of the environment to be used for different purposes. The main technique we used is Simultaneous Localization and Mapping (SLAM) to plot a virtual map of the environment. We used ultrasonic readings for the map generation at backend system and adruino based hardware at the robot side. Different terrains were used to train the system. On the basis of the results, we refine the rules and finalize them. Then we calculated accuracy and efficiency of the designed system. In this thesis we explained all the steps and technicalities of our system with detailed explanation.

CERTIFICATE OF CORRECTNESS AND ORIGINALITY

It is certified that the work done in this thesis document – Optimum Path Generator carried out by GC Abdul Wadood,GC Hamza Nawaz,GC Mureed Ullah and GC Waqas Zahid Abbasi under supervision of Dr. Ayesha Maqbool for fractional completion of Degree of Bachelor of Computer Software Engineering is correct and approved.

Approval

Dr. Ayesha Maqbool

Department of CSE, MCS

Dated:

_____

DECLARATION

The work for this project is not done for any group or individual award or for any qualification in the institute or outside the institute.

DEDICATION

To our parents, without their support and cooperation it would not be possible for us to work in that manner. To worthy teachers who always stood with us and helped us with their valuable suggestions and great motivation.

ACKNOWLEDGEMENTS

All glory goes to Almighty ALLAH, the most merciful, Who led us to this extent. May all glory, honor and Adoration be unto Thy Name.

Our special thanks go to our supervisor **Dr.Ayesha Maqbool** for her guide line, in completing this project successfully. We would also like to pay our thankfulness to the people from MCS, especially Dr Muqeem Ahmed Sherri, A/P Zaki Murtaza and LE Marium Hida.

A deep gratitude towards **Dr. Adnan Ahmed Khan** (Head of Computer Software Department) for his guidance and facilitation for the Project.

At last, we are most obliged to our Parents, their support contributed immensely to the success of this project.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

1. Introduction

Study of a given terrain requires having a map of the concerned environment. Most of the times no previous record or data is available which make it difficult to perform the specific operation. Moreover discovering passages can be impractical for human reach by reasons of security and small space. Such cases can be seen in survey of an environment, tunnel system and military operation.
For this a mobile device which is capable of creating a virtual map of a location can be used in place of survey teams saving both time and money.

1.1 Intended Audience and Reading Suggestions

The Software Requirements Specification (SRS) document is meant for all the stake holders.

- **Project Supervisor:** It will help to supervise the project and guide the team in a better way.
- **Development Team:** It will help the developer to develop the product and to trace back the functional requirements.
- **Testing Team:** It will help the testers to understand the constraints.
- **UG Project Evaluation Team:** Evaluation committee which will evaluate the progress of UG Projects.
- **Staff :** The intended audience is both higher management and local officials of the different departments.

1.2 Motivation

The first step to survey a place is to create a map of it so one can understand the terrain and make arrangements accordingly. Civil engineers give a great importance to create a paper map of the given area so that they can design the architecture.

In Natural Disasters like earthquake and hurricane, buildings do collapse hence killing people and trapping them. A compact size machine is essential to enter into such collapsed buildings and help in the rescue mission.

Pakistan has been affected the most by terrorism. Before Operation Zarb-E-Azb there was a routine of attacks on markets, roads, buildings, government offices, military convoys and paramilitary forces. APS Peshawar Attack 16 Dec 2014 claimed the souls of 150+ teachers and students. Military had to conduct Intelligence Based Operations (IBOs) for a successful swipe of terrorists. The biggest challenge was the lack of

information regarding the composition and structure of terrorists' compounds. So a map generation robot was of great importance.

1.3 Project Objective

The objectives of the project includes creating virtual 2-D map of the environment and then finding the optimum path available. The mobile robot will have the tendency to communicate with the backend system in order to give the desired results. Robot could also be controlled from the system manually to initiate manual override. The software code will include both Matlab and Adruino files intercommunicating in achieving the end result.

1.4 Scope

As with the increase in technology, more compact size robots are being produced for different purposes and are becoming an essential part of our daily life. From easy jobs such as remote control to complex operations in hospitals, robots are everywhere. Moreover keeping priority of the human life, robots are employed in risky tasks like bomb disposal, stealth military operations etc. With the current trend, our project and its applications will increase in the upcoming future. It is to noticed that it has a 2-D map generation capability but can be enhanced with additional components to generate a 3-D map.

1.5 Deliverables

1. Project Hardware(OPG Bot)
2. Code
3. Documentation
4. Video Tutorials

Chapter 2

2 Literature Review

2.1 Introduction

Conventional procedure of creating a map was that engineers used to visit the site, take snapshots and aerial pictures and then making a map using topographical conventions. It usually takes a lot of time and can be inconsistent given if some of the survey is not done. The net result is not dynamic and one cannot give a judgment instantaneously. With the OPG Bot in hand dynamic mapping can be done with an efficient result.

2.2 Problem Domain

As per the existing scenario, there is extreme need by civil engineers to have a survey robot which can map the environment saving time and money. Rescue teams need to excavate through devastated building to dug up dead bodies or alive people having no compact size robot to navigate the area and find an alternate route if any. Military needs a stealth based robot to aid in IBOs to minimize human casualty.

This process incurs many problems, like:

1. Inability of the physical means to meet up the desired accuracy and speed.
2. Inability of the military person to understand the exact nature of the compound to be raided, unless he visits the site.
3. Takes too much time and effort.
4. Inability of authorities to get the problem solved until it is triumphed.

It urges the use of modern technology to minimize the effects of this existing disparity by making a compact size robot addressing these issues.

2.3 Shortcomings/issues

1. Communication is a critical aspect.

2. Mechanical movements have their own limitations

3. Batteries have a limited time.

4. For earlier versions, the contents of the application will be in English language only.

2.4 Proposed Project

A robotic mobile device which is capable of creating a virtual map of a location. The OPG Bot will be put in a location which is partially known at that moment and it will move freely in the area and create a map of the location. Moreover, it has different sensors which will be able to perform different tasks, such as object detection.. Once the map is known to the bot it will be able to find the best available path in that map and reach the destination through the optimum path available.

2.5 Deliverables of the Project

2.5.1 Software Requirement Specification (SRS)

This document is presented to give a description of the project. It explains the purpose and features of the system, the interfaces of the system, what the system will do, its processes and workings, also the constraints under which it must function and how the system will respond to external stimuli. This document is planned for both the developers and the stakeholders of the system. It shall explain how the system will principally aid concerned authorities to team up and cooperate with each other.

2.5.2 Software Design document

The Software design document takes into account all our functional requirements and shows how they interact with each other conceptually. The low level design also shows as to how actually we've been implementing how we are going to implement all of these requirements.

2.5.3 Final Project Report

This is the thesis report which compiles all the previous and current working for the project. Thesis report provides the whole summary for the project and also give details about each and every aspect of the project starting from introduction of the project, literature review, requirements leading to design discussions then testing and lastly future work and conclusion.

2.6 Technological Requirements

OPG Bot has the following requirements in terms of hardware and software.

2.6.1 Project Software Interfaces

1. Matlab/Adruino IDE should be able to run on Windows 7 and onwards.

2. Primary Operating System will be Windows 8.

3. Matlab and Adruino will communicate through serial and object calling.

2.6.2 Hardware Interfaces

2.6.2.1 Computer System

- System shall have
  - Mouse Input
  - Keyboard Input
  - Monitor
  - Usb Port

2.6.2.2 Mobile Device

- OPG Bot containing
  - Ultrasonic Sensors
  - Adruino Uno
  - Motor Controller
  - Batteries
  - Compass Module

2.6.3 Communications Interfaces

1. A connection shall be established between the backend System and the Bot

2. To access the readings, adruino IDE will be used

3. Communication shall be continuous.

2.6.4 Programming Interface

Programming interfaces for project are:

1. Matlab
2. Adruino IDE

Chapter 3

System Requirements Specification (SDS)

3. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to present detailed description of the Urdu finger reader ring for visually impaired and elderly people by defining the problem statement in detail. The detailed requirement of the Bot are provided in this document.

3.1 Purpose

It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react. This document is intended for both the stakeholders and the developers of the system.

This document is meant to outline the features and requirements of OPG Bot, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

3.2 Document Conventions

Basic and standard conventions of writing a specification document are followed without any advanced or complex structure.

3.2.1 Headings Conventions

Headings are focused to make the document in a well defined pattern. They are arranged in a numbered pattern with the top priority heading having a single digit and subsequent headings having numbers according to their level.

About main headings, all of the them are titled as follows: single digit number followed by a dot and the name of the section (All bold Times New Roman, size 18, Centered).

In the same way, all second level sub headings have the same number as their respective main heading, followed by one dot and subsequent sub heading number followed by name of the sub section (All bold Times New Roman, size 16).

Similarly, further sub headings, i.e. level three and below, follows the same rules as above for numbering and naming, but different for font (All bold Times New Roman, size 14).

3.2.2 Figures:

All figures discussed in the document have captions and are numbered accordingly. Context and flow diagrams are based on Unified Modeling Language (UML) standards.

3.2.3 Reference:

All necessary references in the document are provided where needed and where not provided, the meaning explains itself. All ambiguous or unequivocal terms have been clarified in the glossary section at the end of this document.

3.2.4 Links to web pages:

All links have been provided with underlined font, the title of the web page or e-book is written at the top of the link and the title may be searched on google to pinpoint to the exact address.

3.2.5 Basic Text:

All other basic text appears in regular, size 12 Times New Roman. Every paragraph explains one type of idea.

3.3 Intended Audience and0 Reading Suggestions

The intended audiences for the Bot include the project supervisor, the BESE 22 FYP group (developers), UG project evaluation team, and other persons at MCS CSE Department.

3.3.1 Project Supervisor:

It will help the supervisor to supervise the project and guide the team in a better way. This document will be used by her to check whether all the requirements have been understood and in the end whether the requirements have been properly implemented or not.

3.3.2 BESE 22 FYP group (developers, testers, and documentation writers):

For FYP group members, this document will provide the guideline for developing and testing the project.

3.3.3 UG Project Evaluation Team:

It will help the evaluation team to evaluate the progress of FYP project. The document will provide the evaluators with the scope, requirements and details of the project to be built. It will also be used as basis for the evaluation of the implementation and final project.

3.3.4 Reading suggestions:

The SRS begins with the title and table of contents. All level 1 and level 2 headings are given in the table of contents, but the lower sub headings are not included. Each main heading is succeeded by a number of sub headings, which are all in bold format. The product overview is given at the start, succeeded by the complete detailed features, including both functional and non-functional requirements. The entire interfaces are also described. The SRS ends with appendices, including a glossary.

3.4 Project Scope

This document only covers the requirement specifications for OPG Bot. The Bot will help the military/rescue persons, given that the area is previously partially observed, by generating a virtual map of that environment using Simultaneous Localization and Mapping (SLAM) technique. It must be noted that being compact in size it will be able to discover passages that can be impractical for the operating team to do and keeping the users far away from danger. Major implementation of the first version will be able to generate the map, find the optimum path and various other variables by using different algorithms. Moreover it will also be capable of being used by the military to be used in a terrorist compound for Intelligence Based Operations(IBOs) and for the rescue workers in case of devastated area after any natural disaster e.g. earthquakes etc or criminal activity e.g. bomb blast.

3.5 References

3.5.1 IEEE Computer Society Conventions:

- **Use Case Modeling Guidelines**, which documents the guidelines used to develop the use case model specifying the functional requirements in this specification. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=787548
- **System Requirements Specification Content and Format Standard**, which specifies the content and format of this specification.

http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=15571&arnumber=720574&punumber=5841

- **System Requirements Specification Template**, which provides the skeleton for this specification. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=16016&arnumber=741940&punumber=5982
- **Centre for Linguistic Engineering** http://www.cle.org.pk/software/ling_resources.htm
- **MIT NEWS** http://newsoffice.mit.edu/2015/finger-mounted-reading-device-blind-0310

3.6 Overall Description

3.6.1 Product Perspective

OPG Bot is basically built to help in the creation of a virtual map which in terms can be helpful in various activities as described in latter sections.

3.6.2 Product Features

The main features of OPG Bot are highlighted below:

1. Wireless Connection of Bot with Remote Device i.e Laptop
2. SLAM
3. Data Encryption at both Bot and Laptop Side
4. Use of Ultra Sonic Sensors
5. Use of Adruino
6. Self Destruct capability
7. Desktop Application for Virtual Map Generation
8. Use of camera and Lights for Live Feed

3.7 User Classes and Characteristics

3.7.1 Summary of User Classes

The following section describes the types of users of the OPG Bot. There are explanations of the user followed by the interactions the user(s) shall be able to make with the software.

3.7.1.1 Rescuer

The user in this is basically the rescue worker that has been called after a natural disaster and has job to find a passage under the devastated building so that the operation to find and rescue the people strangled in the debris.

3.7.1.2 Military IBO Officer

The user in this is basically a military officer given the command of an IBO to find and neutralize/arrest terrorist from a compound whose structural design is not completely known.This Bot will be used from a remote location and owing to its compact size it will provide with the necessary stealth required.

3.7.1.3 Surveyor

The user in this is basically a surveyor that wants to get a very specific and accurate map of an environment.

Use Case Diagram

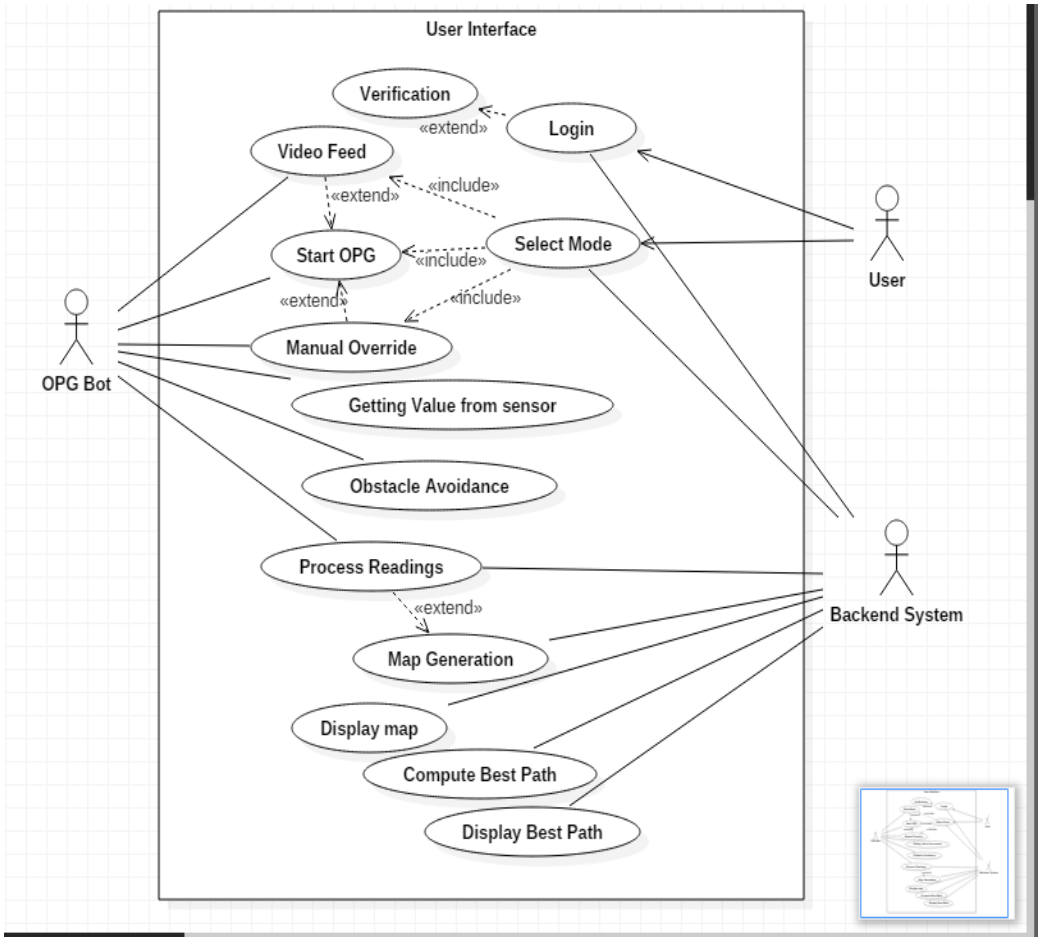


Fig 1 Use case Diagram

3.8 Operating Environment

3.8.1 Hardware

The OPG Bot operates, both directly or indirectly given the condition, with the mentioned external hardware:

- **Camera Mounted on Bot**: The camera is used to make video in real time to provide with the live feed and for Image Processing. i.e OV5647 Webcam for Adruino

- **Wireless Modem**: The data will be transferred from the Bot to the computer. Rosewill USB WiFi Nano Adapter Dongle
- **Adriouno**: Directly in communication with the Bot and Remote System.i.e Adruino Uno
- **Sensors**: UltraSonic sensors will be used particularly. i.e HC-SR204
- **RC Car**: To be used as the carrier of the Sensors,modem and processing unit i.e adriuono.i.e Mx Mixse Smart Robot car Kit for The Adruino Uno

3.8.2 Software

- Windows: 7, 8, 8.1, 10.
- Python, Matlab.

3.9 Design and Implementation Constraints

- OPG Bot will only process Ultra Sonic signals for map generation.
- It will only be used for ground survey.
- It will only be used for 2D analysis
- Input may contain noise along with the data.
- There can be time delay and less fluency.
- It has to be remotely controlled to avoid big gaps in the ground
- There is a limited distance the OPG Bot can travel from the backend system

3.10 User Documentation

Following are the guides for the OPG Bot:

- Usage manuals with pictures and text for using the device

3.11 External Interfaces Requirements

3.11.1 User Interface

- Current frame will be displayed on left top side of screen.
- Map under construction will be displayed on right side of screen.

3.11.2Hardware Interfaces

- Video input will be taken through camera.
- Communication between software and camera will be done wirelessly.
- Ultra Sonic sensors will give input direct to Adriuono.

3.11.3 Software Interfaces

- To convert readings into virtual map.
- To generate the best path.

3.11.4 Communications Interfaces

- Wireless modem will be used as medium of communication between camera and software. Wireless Modem will transfer video from camera to computer.
- Signals from sensors will be used in the map generation process.
- Camera feed will help the user to better understand the environment and remotely control the OPG Bot
- Map will be in 2-D.

3.12 System Features

This section describes in detail the system features of the OPG Bot. System features are as follow:

1. Video/Ultrasonic Input
2. Computation on Ultrasonic readings
3. Convert readings into map simultaneously
4. Keep track of the map
5. Generate Optimum Path

3.12.1 Video Input

Use Case Requirement: The camera mounted on the ring will capture the video in real time. This video will be used for live feed
Use Case Paths • Normal: – Video sent for feed. • Exceptional: – Beep Sound is produced
Normal Path: Video sent feed
Externals • Wireless Communication
Preconditions • The camera captures the video in real time.
Interactions • The captured video is sent to the system for live feed.
Post conditions • Captured video is displayed
Categorization • **Frequency**: Normal • **Criticality**: Normal

• **Probability of Defects**: Low • **Risk**: Normal
Exceptional Path: Beep Sound is produced
Externals • Wireless Modem
Preconditions • The camera is not working
Interactions An error signal is sent to the system
Post conditions • The beep sound is produced through the speakers of the computer.
Categorization • **Frequency**: High • **Criticality**: High • **Probability of Defects**: Low • **Risk**: High

3.12.2 Input Ultrasonic Readings

Use Case Requirement: The readings captured by the sensors are sent to the remote device wirelessly.

Use Case Paths
 	
• Usual:
  – Readings received successfully
• Unusual
  – No readings received

Usual Path: ***Readings received successfully***
Externals • Ultrasonic sensors

Preconditions • The readings are sent to the system for processing.
Post conditions • The system start generating the virtual map.
Categorization • **Frequency**: High • **Criticality**: High • **Probability of Defects**: Medium • **Risk**: High

3.12.3 Path Generation

Use Case Requirement: Once the map is generated,different algorithms will start computing to find the best available path from the path(s) availale

Use Case Paths
 	
• Normal:
  – Path Generated
• Exceptional:
  – Path Generation Unsuccessful

Normal Path: Path Generated
Externals • Complete Map
Preconditions • Whole map of environment is generated
Interactions • Whole map is examined
Post conditions • Best path is found
Categorization • **Frequency**: Normal

• **Criticality**:Normal • **Probability of Defects**: Low • **Risk**: High
Exceptional Way: Unsuccessful Path Generation
Externals • Complete map
Preconditions • Whole map of environment is generated
Interactions • Whole map is examined
Post conditions • Best path cannot be generated
Categorization • **Frequency**: Low • **Criticality**: Normal • **Probability of Defects**: High • **Risk**: Normal

3.13 Other Nonfunctional Requirementsc

3.13.1 Security Requirements

Application running on the computer should not need any additional information other than the collected data from the user or already present data.

3.13.2 Software Quality Attributes

3.13.2.1 Usability

The graphical user interface of app is to be designed with usability as the first priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate.

3.13.2.2 Accuracy

Ensuring reliability and correctness, there will be zero tolerance for errors in the algorithm that computes results.

3.13.2.3 Portability

Portability is the key factor in it as the remote system can be placed anywhere and so does the bot to ensure maximum portability.

3.13.2.4 Availability

The application will be available from boot to shutdown, provided system is in working state and the application is installed and configured properly.

3.13.2.5 Flexibility

The design and architecture of the application will be flexible enough for catering any new requirements, if any at some later stage or for the application enhancement.

3.13.2.6 Data Integrity

If the system crashes during addition, deletion or editing there will be no changes.

3.13.2.7 Scalability

The application is expected to handle one user at a time. One instance of the application could be opened on a single mobile at a time.

Chapter 4

System Design Specification (SDS)

4. Introduction

This design document covers all our functional requirements and demonstrates how they interrelate with each other abstractly. The low level design also illustrates as to how we have been implementing and how we are going to implement all of these requirements. This low level design for the time being does not address any non-functional requirements that our system has and that have been mentioned in the SRS Document.

4.1 Purpose

The purpose behind this Software Design Document is to convey a depiction of the structure of our framework reasonable enough to take into consideration programming advancement to continue with a comprehension of what is to be constructed and how it is anticipated to be created. This Software Design Document gives data basic to get a portrayal of the subtleties for the product and the framework to be manufactured. The motivation behind this record is to introduce a structure see and point by point portrayal of OPG Bot. It will clarify the reason, highlights, interfaces, wha the framework will do, its whole procedures in detail, the requirements under which it must work and how the framework will respond to inputs and what will be its yields.

4.2 Scope

This document only covers the requirement specifications for OPG Bot. The Bot will help the military/rescue persons, given that the area is previously partially observed, by generating a virtual map of that environment using Simultaneous Localization and Mapping (SLAM) technique. It must be noted that being compact in size it will be able to discover passages that can be impractical for the operating team to do and keeping the users far away from danger. Major implementation of the first version will be able to generate the map, find the optimum path and various other variables by using different algorithms. Moreover it will also be capable of being used by the military to be used in a terrorist compound for Intelligence Based Operations(IBOs) and for the rescue workers in case of devastated area after any natural disaster e.g. earthquakes etc or criminal activity e.g. bomb blast.

4.3 Definitions, Acronyms & Abbreviations

The conventions which are used to prepare this document are as follow

- Font: Times New Roman, (size 12)
- Main headings, Font – Times New Roman, Bold, size 18
- Sub headings, Font – Times New Roman, Bold, size 16

- Sub-sub headings, Font – Times New Roman, Bold, size 14

Acronyms	Complete
OPG	Optimum Path Generator
IBO	Intelligence Based Operations
MCS CSE Dept	Military College of Signals' Computer Software Engineering Department
NUST	National University of Sciences & Technology
SDS	System Design Specification
RC Car	Remote Controlled Car

Table 1 Acronyms

4.4 References

- NUST Regulations (Revised 2016): Part-II Academic Programs
- 2017, Software Requirements Specification for Academic Analysis System
- https://cms.nust.edu.pk
- 2014, Software Engineering Standards Committee of the IEEE Computer Society, "IEEE Recommended Practice for Software Design Descriptions", IEEE Std 1016-1998.
- Applying UML and Design Patterns - An Introduction to Object-Oriented Analysis and Design (Craig Larman) available from: http://www.ebookdirectory.com/
- UML basics: An introduction to the Unified Modeling Language (Donald Bell), available from:http://www.ebookdirectory.com/
- UML Distilled: A Brief Guide to the Standard Object Modeling Language by Martin Fowler, available from: http://www.ebookdirectory.com/
- Learning UML 2.0 by Russ Miles, available from: https://www.scribd.com/
- The Unified Modeling Language User Guide by Grady Booch, available from: https://www.scribd.com/

4.5 Overview of the Document

The document is divided into sections and is already listed in the table of contents and figures list. However, a brief description of all the sections is mentioned below.

4.6 System Architecture Description

In this section, the overall architecture of the system is discussed, including the introduction of various components and subsystems. It is mainly supported by System Architecture diagram which shows an insider's perspective of the system by describing the high level software components that perform the major functions to make the system operational.

4.7 Structure & Relationships

This section ponders upon the interrelationships and dependencies among various components. It is mainly described by a diagram which is further augmented by explanatory text. UML Class diagram also helps us understanding the system structure.

4.8 UML Class Diagram

UML Class diagram further manifests the description of low level components of the software that include data storage and state details, thus making the system adequately comprehensible.

4.9 User Interface Issues

This section presents the main principles of the product's user interface. Not touching about the technical details, the section is described by an overall diagram which is also augmented by explanatory text. Moreover, Activity diagrams, Sequence diagrams, and UI Design diagrams also elaborate the User Interface issues in a more intelligible manner.

4.9.1 Activity Diagram

Activity Diagrams follow a methodology dependent on work process structure to characterize the general working and stream of activities of the framework. They are gainful to perceive how various advances are associated with significant assignments inside a framework utilizing an example dependent on stream outline without diving into the further specialized subtleties.

4.9.2 Sequence Diagram

Sequence diagrams show how different objects are involved in the completion of a functionality of the system. They have a single format that gives permission the reader to see how many objects are used vis-à-vis their duration; for the completion of a system requirement.

4.9.3 Detailed Description of Components

This section contains detailed description of all the major components of the system in a structured pattern (table), comprising of 10 x rows. The pattern (table) maintains symmetry in the document structure; and therefore it is followed for each of the components. Each part/row of the table is identified by a label, explaining the purpose of each point. The description of each point vis-à-vis the component being discussed, ponders upon the detailed account of it in the system.

4.10 Overview of the Modules

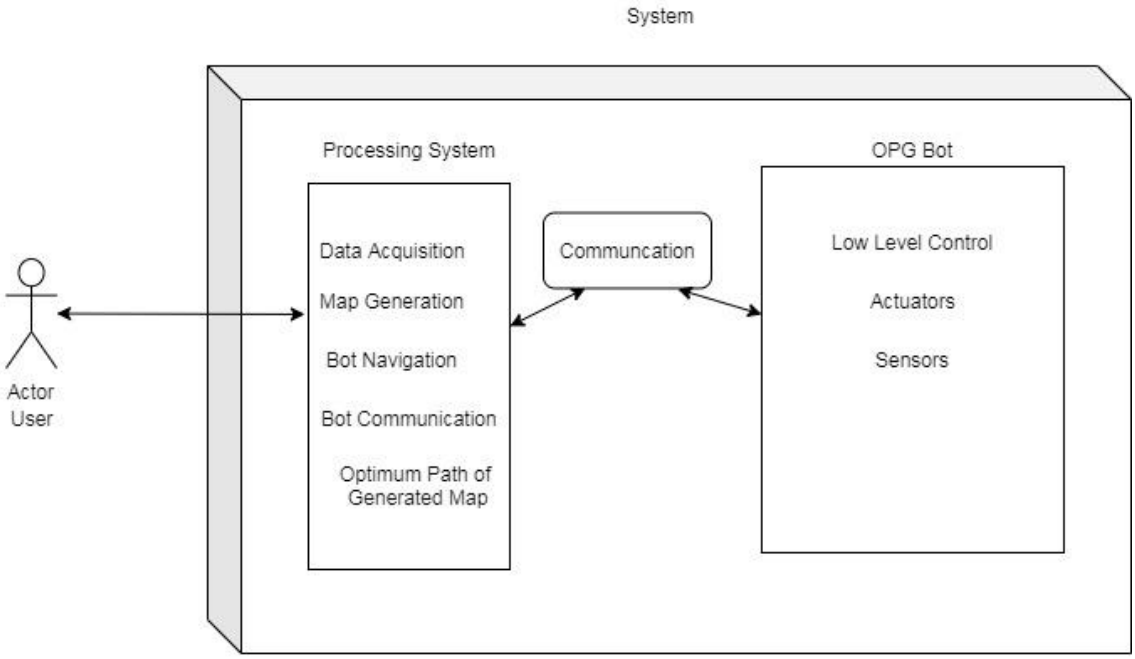


Figure 2 - Abstract Diagram

4.11 Explanation of Abstract Diagram

The system will be architected mainly in three fundamental modules “Users”, “Processing System”, “OPG Bot”.It will further be having sub modules as shown in the abstract diagram above. Abstract diagram provides an overview of the system, from users accessing the system till the processing in databases. The sub modules of the Abstract diagram are further elaborated below.

4.11.1 Users

Users of the OPG Bot will access the backend System to login and then choose functionality according to the requirement. Users consist of Army Officials/Civil Engineers/Rescue Team etc..User interacts with the Bot through Remote Backend System that further accesses the services provided by the Bot.

4.11.2 Processing

Backend system will provide a platform to the user to interact and consume the services of the Bot in an affective and more manageable manner.

4.11.3 OPG Bot

The Bot itself is the deployable machine capable of moving in an environment to make the virtual map. It takes measurement from the sensors and generate the map according to the defined algorithm.

4.11.4 Block Diagram

The principle parts of the OPG Bot are shown in the figure and their relation is defined by the connection between them. The details of all the modules is also given below

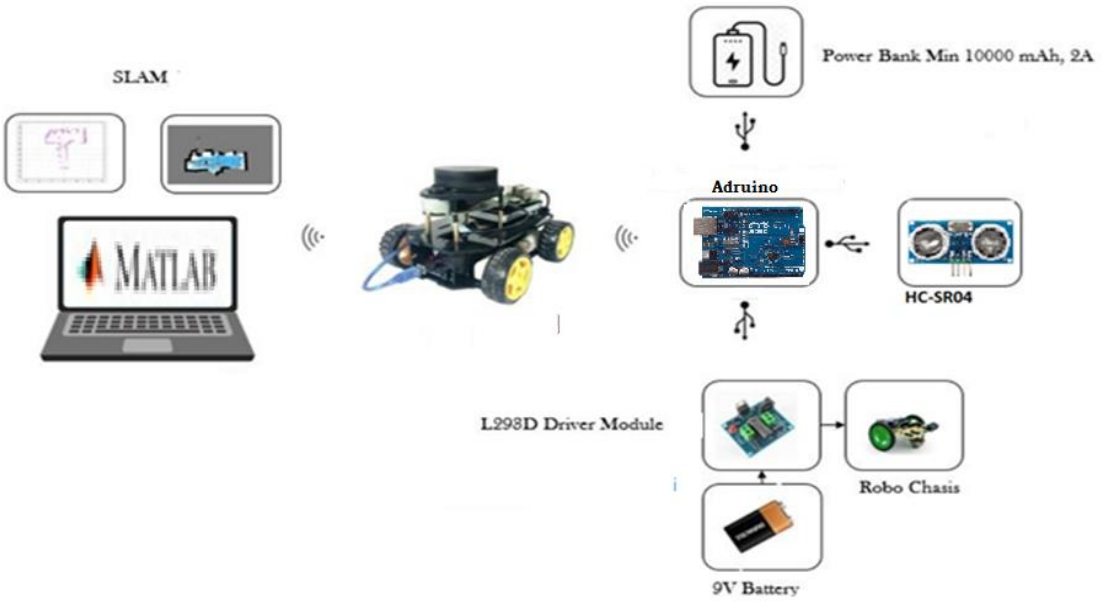


Figure 3 - Block Diagram

4.11.5 Block Diagram Description

Adruino loaded with code through IDE Software Package

9V battery is connected to the driver circuit

Adruino get supply by connecting with the Power Bank

Matlab/Python installed in Laptop receive HC-SR04 scan Values by using ROS Network configuration with ROS software package in Matlab

Based on the scan control commands for a car can be sent from laptop to the Bot

4.12 System Architecture Description

The main purpose of the project is to develop a Bot for the military capable of generating a map of environment to aid for the operations in congested and unknown environment. The user can use the system to see the terrain in a virtual environment . It will offer the users to let the Bot run in auto mode or they can choose to switch into manual mode. The system is being developed to mechanize the entire process, which is currently being performed on manual basis.

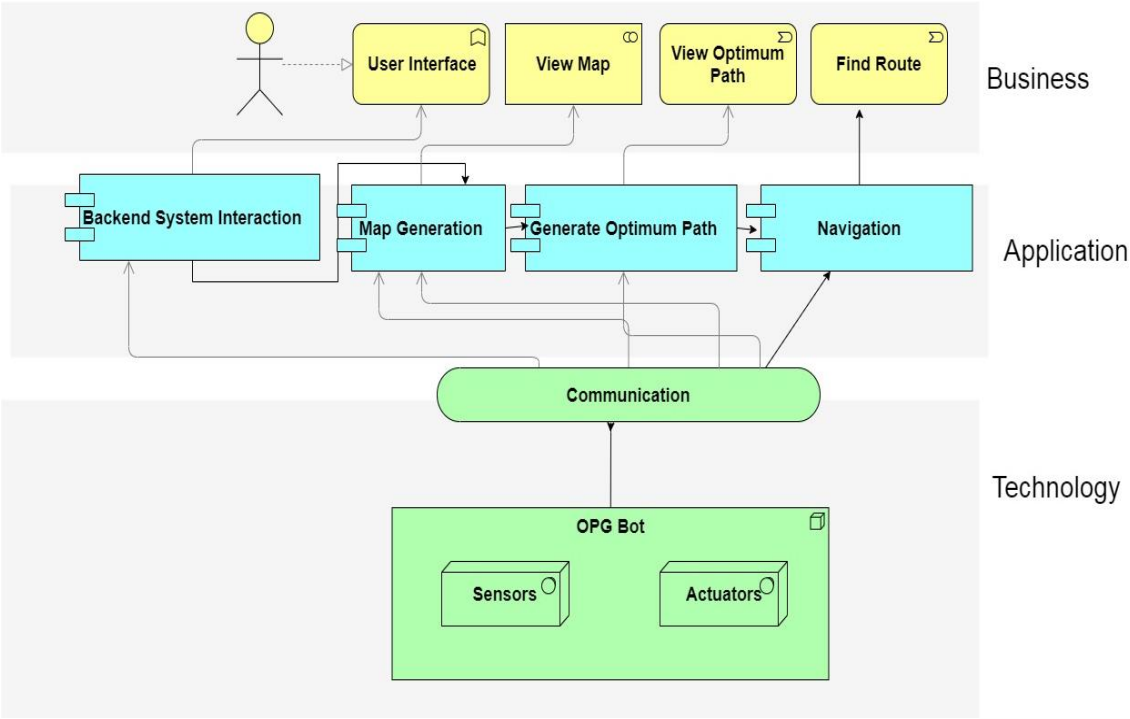


Figure 4 – System Architecture Diagram

4.13 Use Case

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.

The various user classes identified the following use cases and primary actors for the OPG Bot:

4.13.1 Bot Design

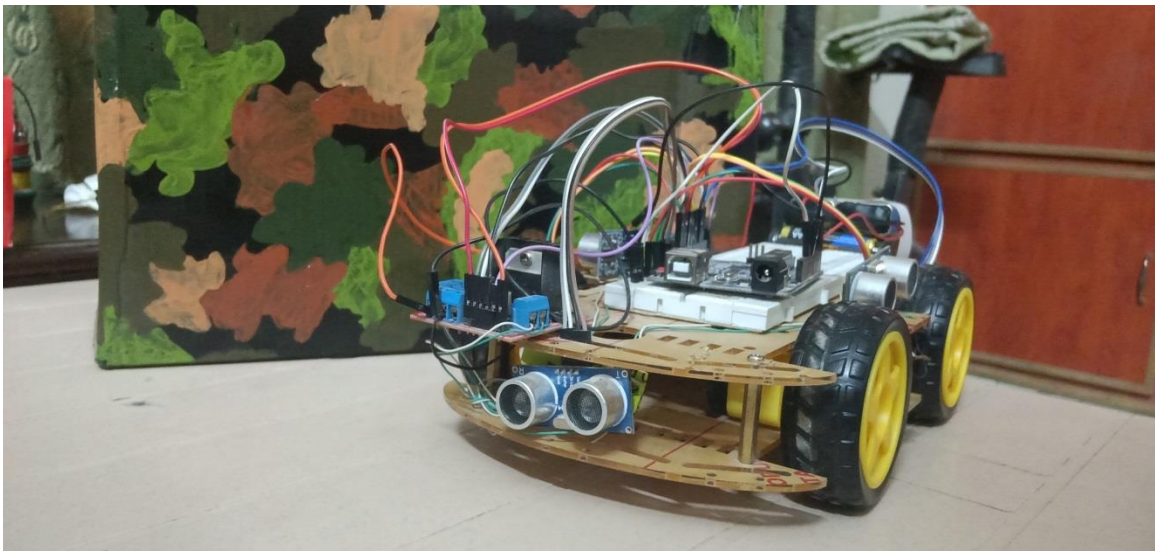


Figure 5 -Bot Design

4.13.1.1 Video Feed

Use Case Requirement: The camera mounted on the ring will capture the video in real time. This video will be used for live feed
Use Case Paths Normal: Video sent for feed. Exceptional: Beep Sound is produced

Normal Path: Video sent feed
Externals Wireless Communication
Preconditions The camera captures the video in real time.
Interactions The captured video is sent to the system for live feed.
Post conditions Captured video is displayed
Categorization **Frequency**: Normal **Criticality**: Normal **Probability of Defects**: Normal **Risk**: Normal
Exceptional Path: Beep Sound produced
Externals Wireless Modem
Preconditions The camera is not working
Interactions An error signal is sent to the system
Post conditions The beep sound is produced through the speakers of the computer.
Categorization **Frequency**: High **Criticality**: High **Probability of Defects**: Low **Risk**: High

4.13.1.2 Getting Value from sensors

Use Case Requirement: The readings captured by the sensors are used to generate the map using defined algorithms.
Use Case Paths Usual: Readings received successfully Non Usual Readings not received
Usual Path: ***Readings received successfully***
Externals Ultrasonic sensors
Preconditions The sensors are correctly connected with the Rpi
Post conditions Generate/Update the virtual map.
Categorization **Frequency**: High **Criticality**: High **Probability of Defects**: Medium **Risk**: High

4.13.1.3 Compute Best Path

Use Case Requirement: Once the map is generated,different algorithms will start computing to find the best available path from the path(s) available

Use Case Paths Normal: Path Generated Exceptional: Path Generation Unsuccessful
Normal Path: Path Generated
Externals Complete Map
Preconditions Whole map of environment is generated
Interactions Whole map is examined
Post conditions Best path is found
Categorization **Frequency**: Low **Criticality**: Normal **Probability of Defects**: Normal **Risk**: Normal
Exceptional Path: Path Generation Unsuccessful
Externals Complete map
Preconditions Whole map of environment is generated
Interactions Whole map is examined
Post conditions Best path cannot be generated
Categorization

Frequency: Low **Criticality**: Medium **Probability of Defects**: High **Risk**: High

4.14 Class Diagram

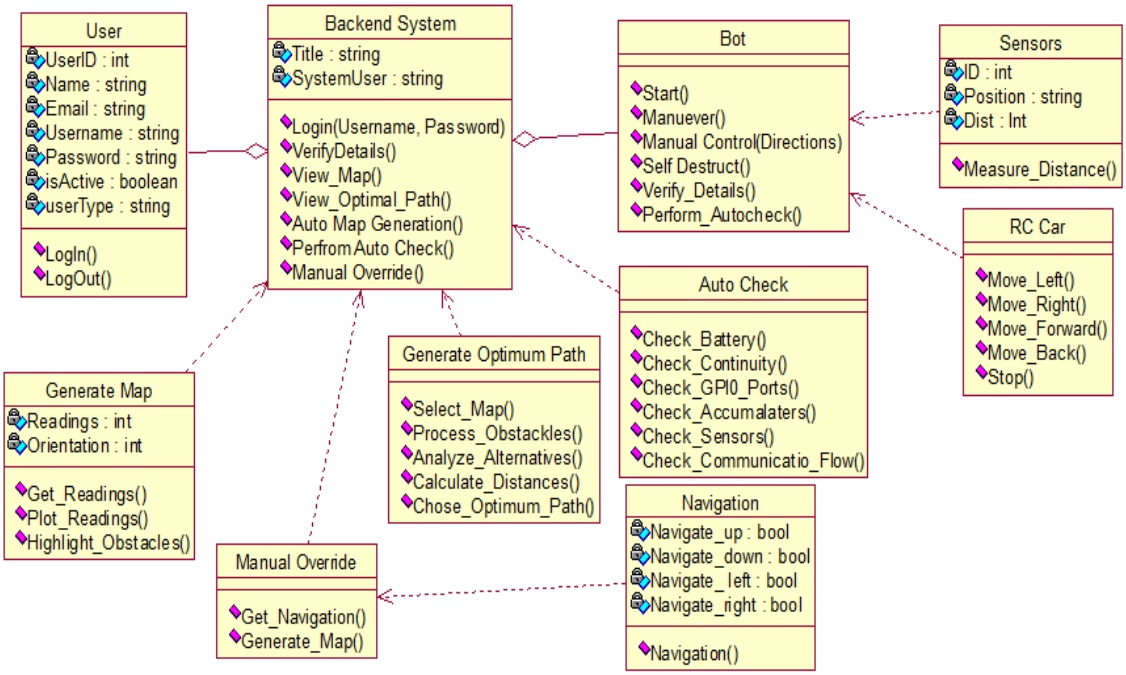


Figure 6 - Class Diagram

4.14.1 Class Diagram Description

Class Name	Description
Backend System	Backend System class contains the origin for the function OPG Bot has to perform. It is the main class which will be acting as a gateway to all the other classes
User	User class contains all the information related to user management. It has aggregation with other classes of user categorization and the functions that performs all the user management functions.
Bot	Bot Class contains the main attributes and functions performed by the maneuverable RC-Car. It takes measurements from these sensors. Generates the

	Map and also can self destruct.
Sensors	This class contains the information of individual sensors. Distance is calculated by these sensors.
RC-Car	RC Car class contains the attributes including all the movements of the RC car in every direction.

4.15 Activity Diagrams

4.15.1 User Management

The diagram below (Figure 6) displays how the users are logged into the system and displayed specific interfaces according to their category.

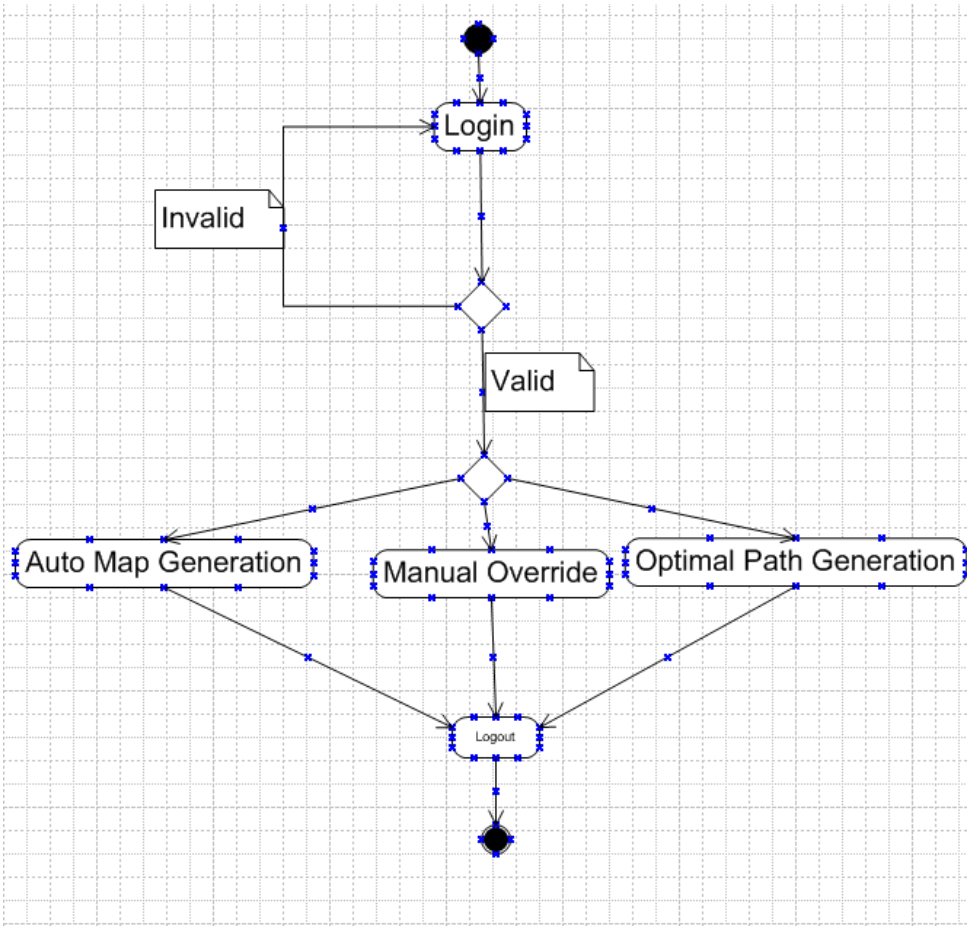


Figure 7 - User Management – Activity Diagram

4.15.2 Auto Map Generation

Figure 7 displays the activites performed when the user selects to generate the map automatically.

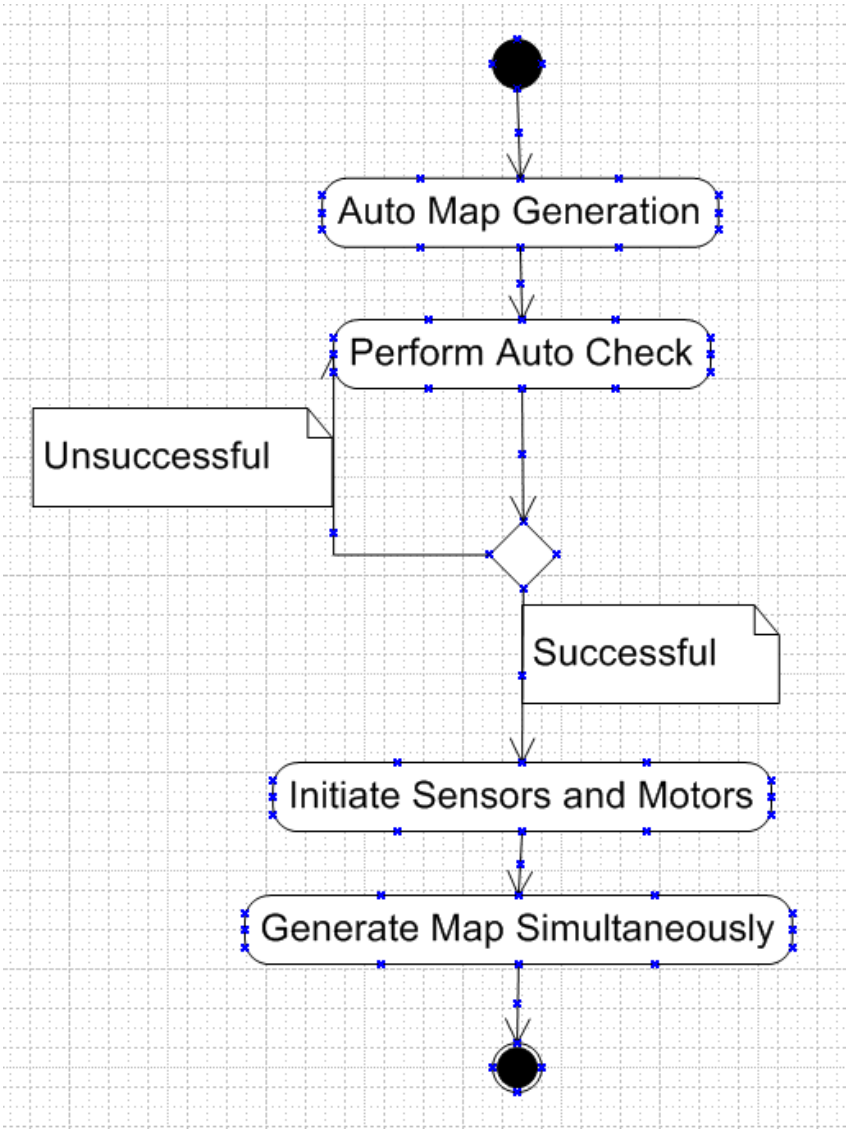


Figure 8 – Auto Map Generation – Activity Diagram

4.15.3 Optimum Path Generation

Figure 8 shows the tasks performed by the optimum path generation module

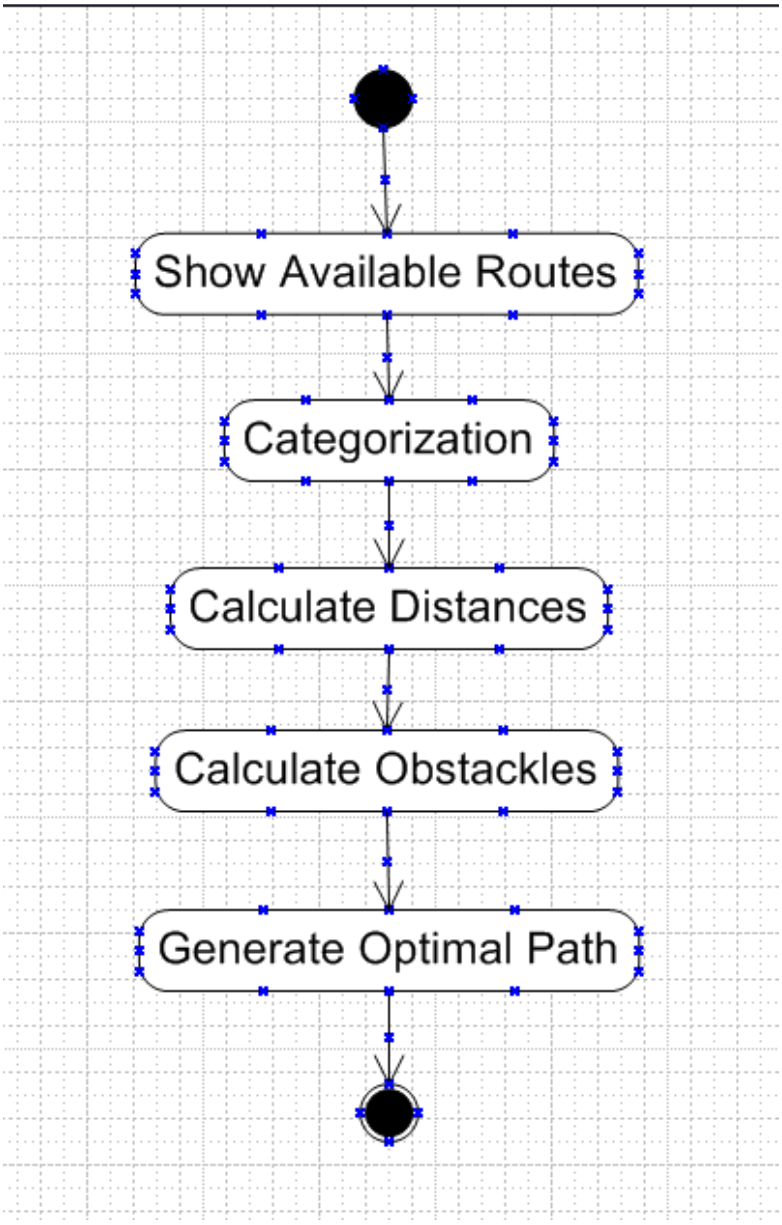


Figure 9 –Optimum Path Generation – Activity Diagram

4.15.4 Manual Override

Diagram below shows the activites being performed when the OPG Bot is in Manual control mode

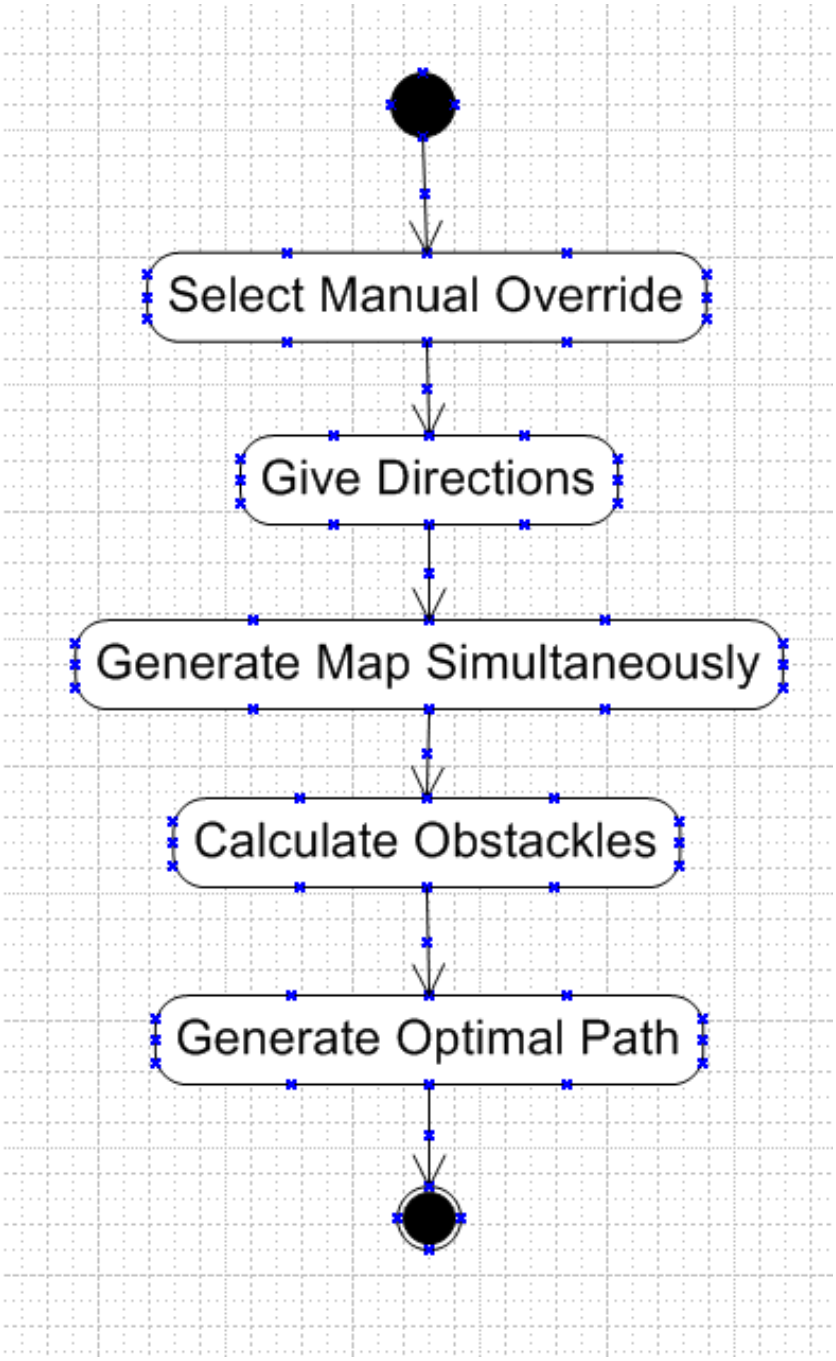


Figure 10 – Manual Override – Activity Diagram

4.16 Sequence Diagrams

4.16.1 Login

The following diagram shows the sequence of events as a user log in into the system.

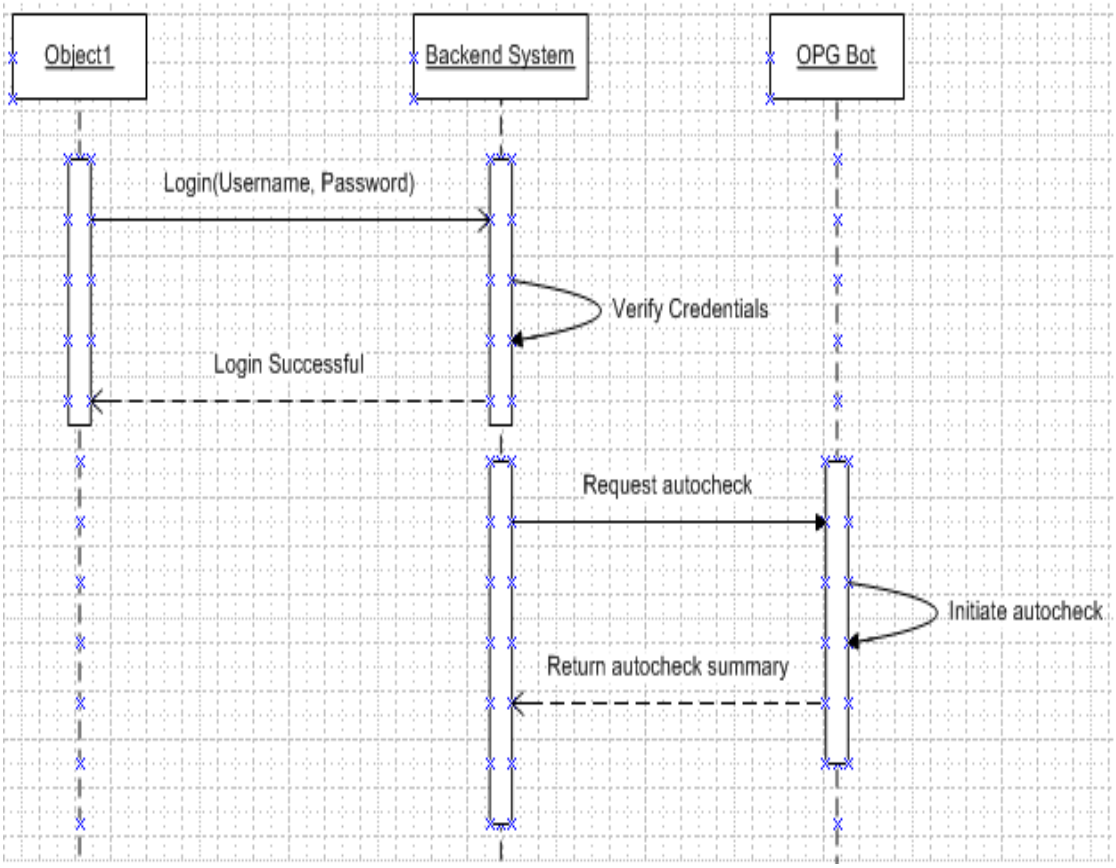


Figure 11 - Login- Sequence Diagram

4.16.1 Auto Map Generation

This sequence diagram displays the procedure of the Bot generating the map.

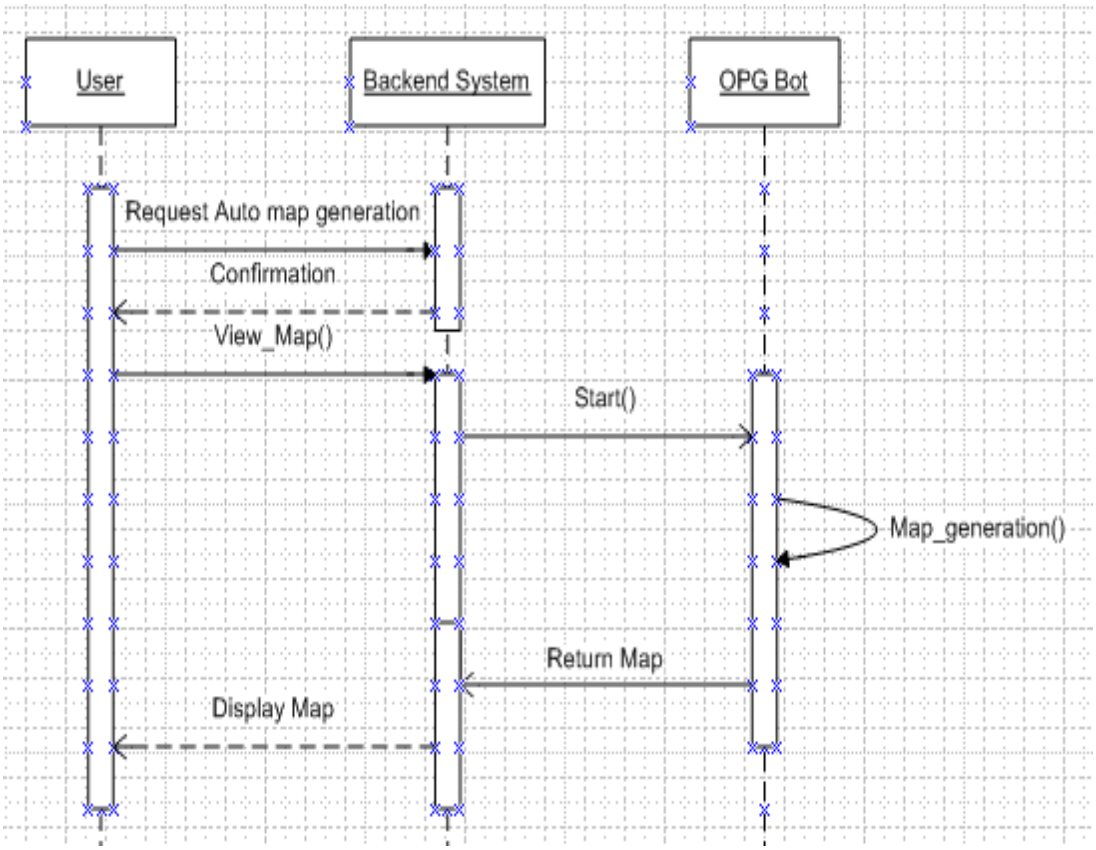


Figure 12–Auto Map Generation - Sequence Diagram

4.16.2 Display Optimum Path

All the effort is focused to find the best available path.Following Sequence diagram depicts the procedure to show the optimum path in the map.

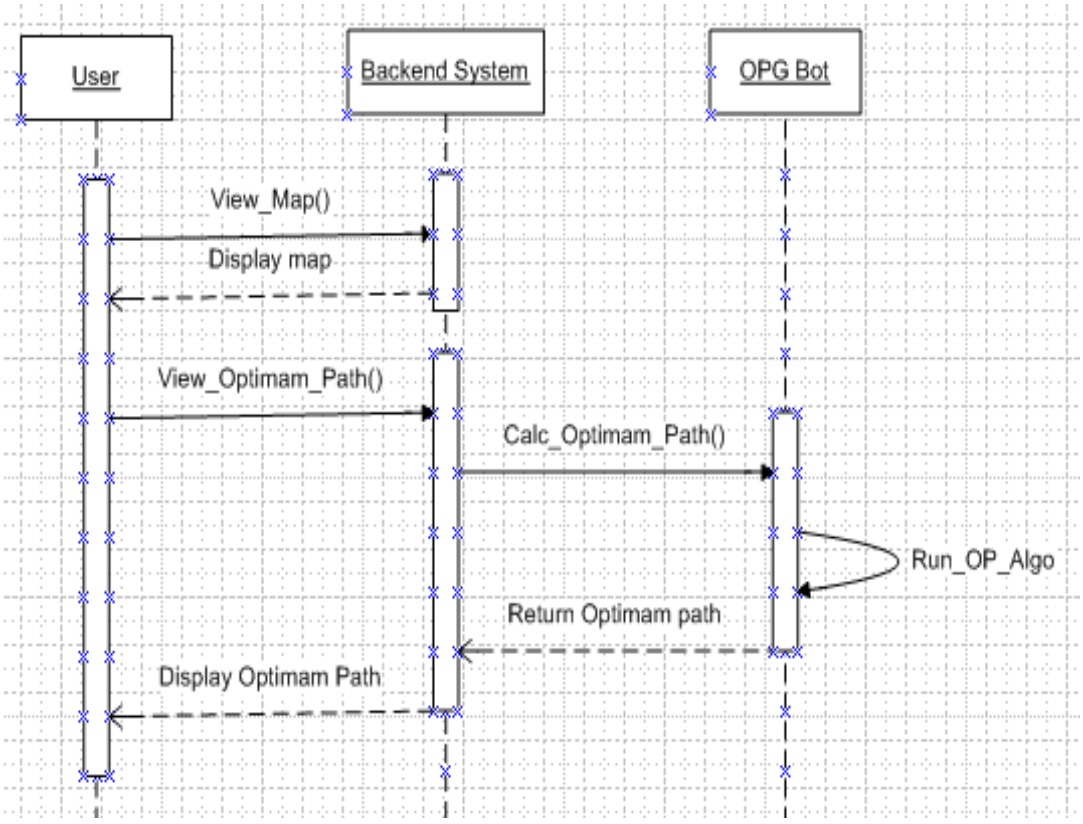


Figure 13–Display Optimum Path - Sequence Diagram

4.16.3 Manual Override

It is used to manually control the OPG Bot by giving directions from the backend system

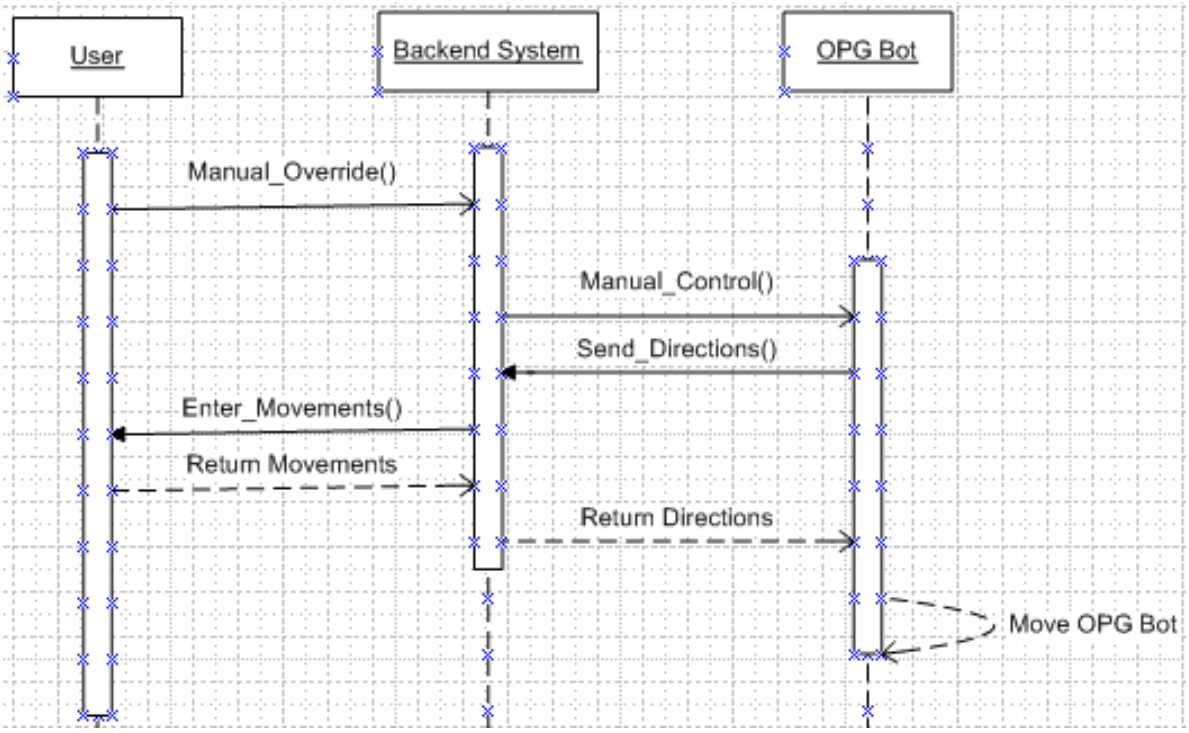


Figure 14 – Manual Override - Sequence Diagram

4.17 UI Design

The system under development shall support an intuitive and easy to use UI that will have an extremely shallow learning curve and require minimum training to be operated at maximum efficiency. In this way, users of the web application won't feel reluctant to use the application to keep an eye on the results, while the faculty and staff won't feel reluctant to adopt the new system as means of academic analysis either.

4.17.1 Login Screen

Following are the sketches of a possible UI implementation for OPG Bot. This will be the first screen that the user sees upon opening IT. The user shall be presented with a form to enter username and password for login. As visible, the interface is simple and self-explanatory.

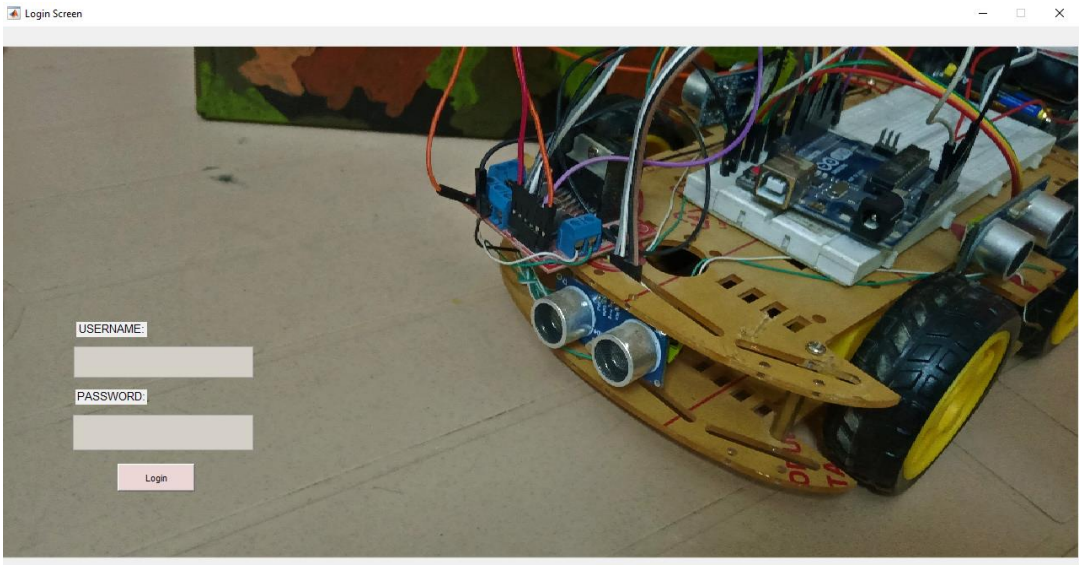


Figure 15 - Login Screen UI

4.17.2 Main Menu

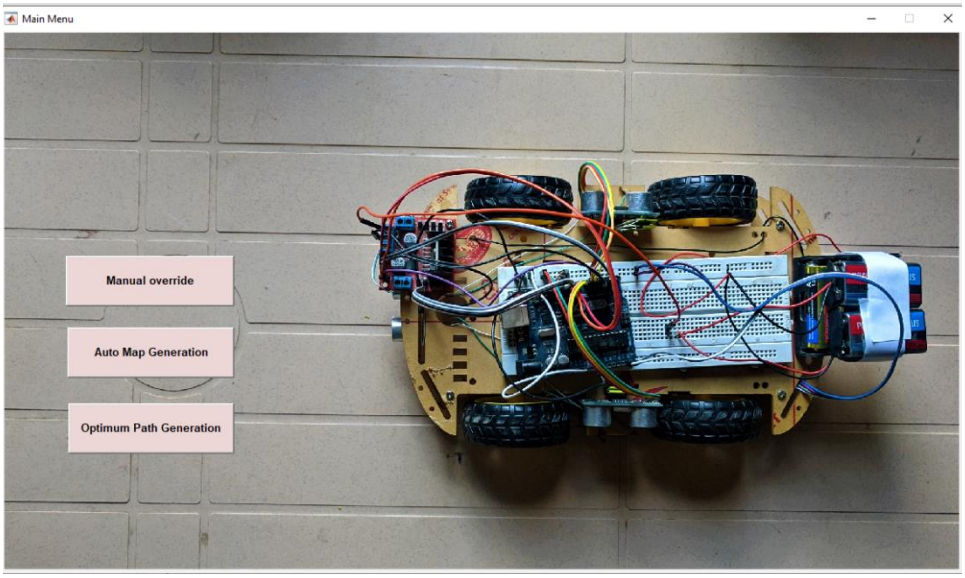
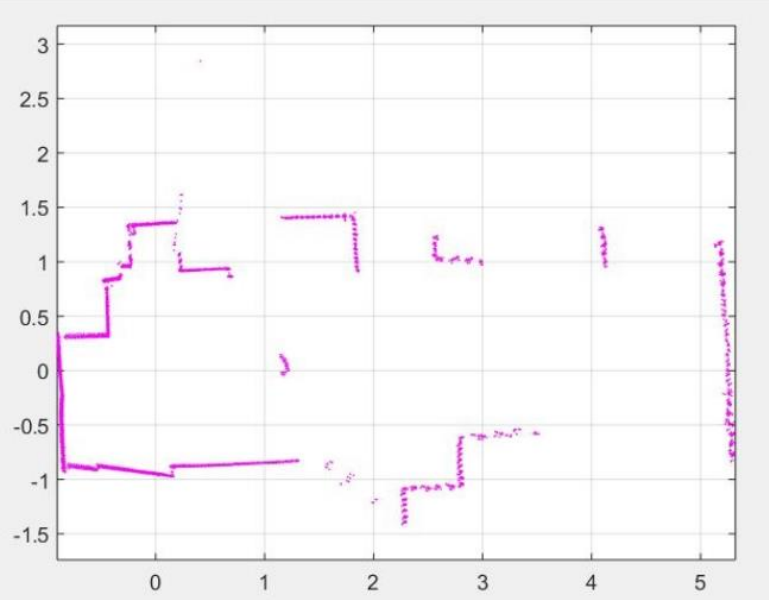
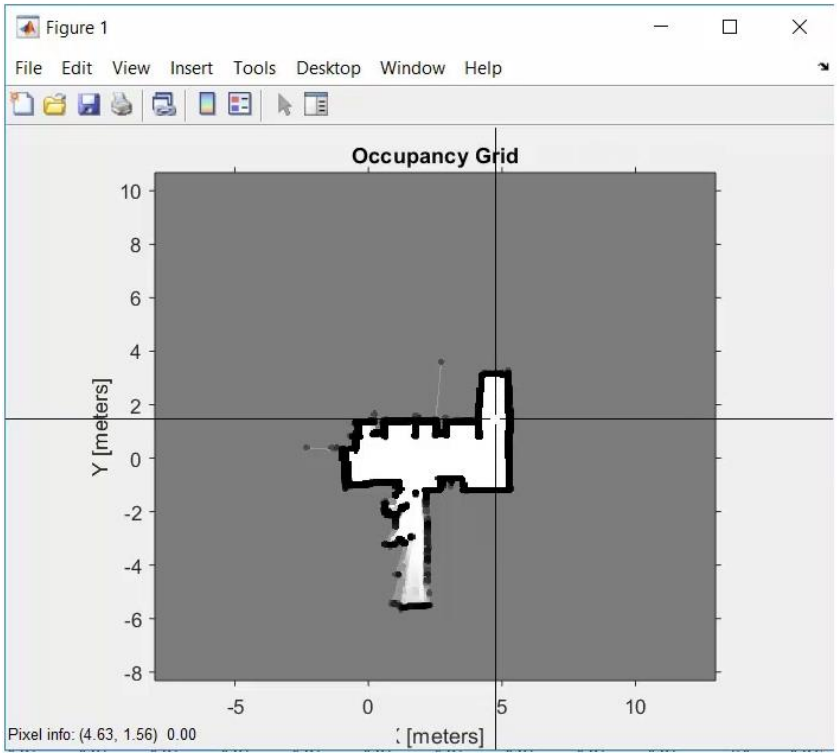


Figure 16 – Main Menu UI

4.17.3 Map Generation Process



4.17.4 Destination Marking in binary occupancy grid



4.17.5 Path Planning

4.17.6 Destination Reached

4.18 Appendices

Activity diagrams - are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).

Class diagram - In the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

SDS – In the context of software, Design Specification is usually a design document that describes all data, architectural, interface and component-level design for the software. A design specification provides explicit information about the requirements for a product and how the product is to be put together.

Sequence diagram – It is an association chart that shows how procedures bury work with each another and in what request. It is a development of a Message Sequence Chart. A succession outline shows object associations orchestrated in time arrangement. It portrays the items and classes engaged with the situation and the arrangement of messages traded between the articles expected to do the usefulness of the situation.

SRS–A software requirements specification is a description of the software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Use case diagram - At its most straightforward is a portrayal of a client's connection with the framework and delineates the particulars of an utilization case. An utilization case graph can depict the various kinds of clients of a framework and the case and will frequently be joined by different sorts of charts too.

4.19 Requirement Matrix

Requirement ID	Requirement description	Component
3.1	User Interface	User Interface Design
3.2	Login	Login Screen Figure 13
3.3	Main Menu	Main Menu Screen Figure 14
3.4	Map Generation	Map Generation Process 10.3

3.5	Communication	Backend System
3.6	Readings	Use Case Description 6.2.2
3.7	Path Generation	Use Case Description 6.2.3
3.8	Components with Interaction	Block Diagram Figure 2

Table 2 – Requirement Matrix

Chapter 5

5. System Implementation

5.1 Technology Used

5.1.1 Development Tools

Matlab is being used as the primary map generation tool communicating with adruino IDE to work with robots mechanical movements.

5.1.2 Database

The systems Database was established and achieved using MS Excel

5.1.3 Operating System

The System Interface shall be running on Microsoft Windows and can run on Windows 7 and all future iterations of it. The Adruino code shall be executing on Adruino IDE.

5.2 Complete System Implementation

The system comprises of two main components. Backend System and Robot. Backend system is responsible for all the computations and control. OPG Bot is the mobile device moving and getting readings from sensors and transferring it to backend system

Login Module

Module is linked with the database and forms the data access layer of the application. Model Layer consists of all the functions that are then accessed from the view. Usernames and Passwords stored in the data base are compared with Username and Password entered by the user.

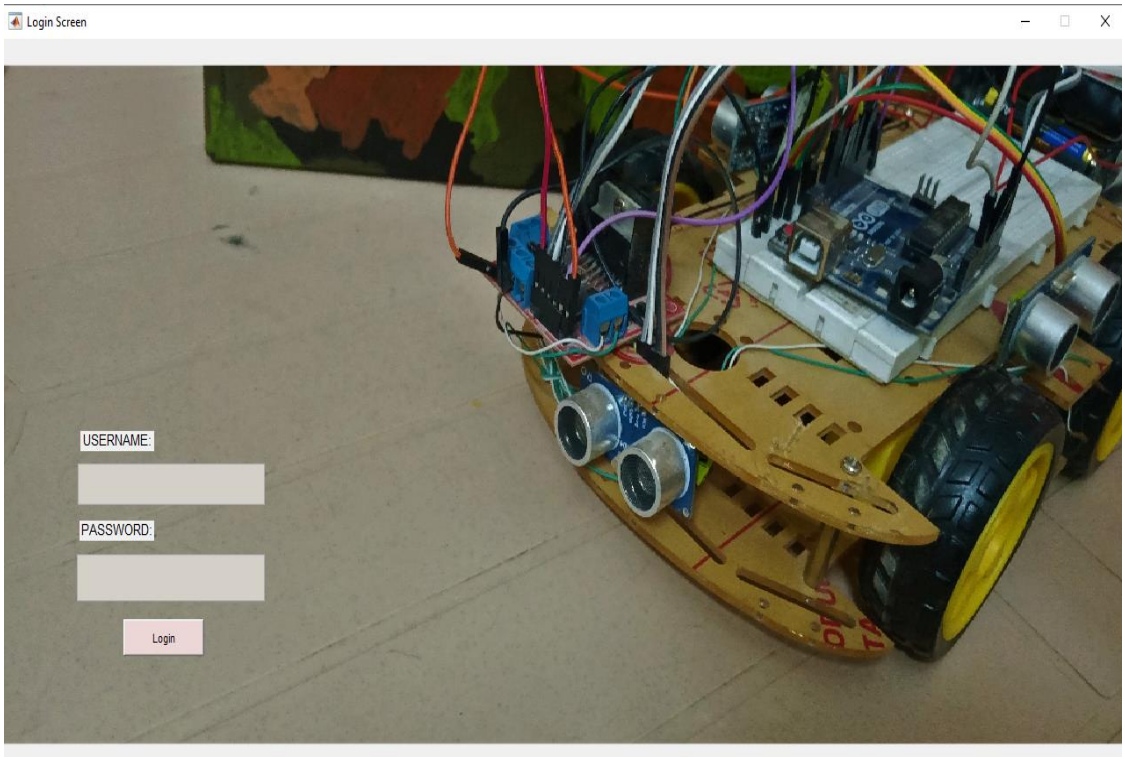


Figure 17 - Login Screen

Main Menu Module

This is the main module, which greets the user following successful login. This is used to access all the different functionality of the application and can be considered as a main menu. It can be used to select between map generation.

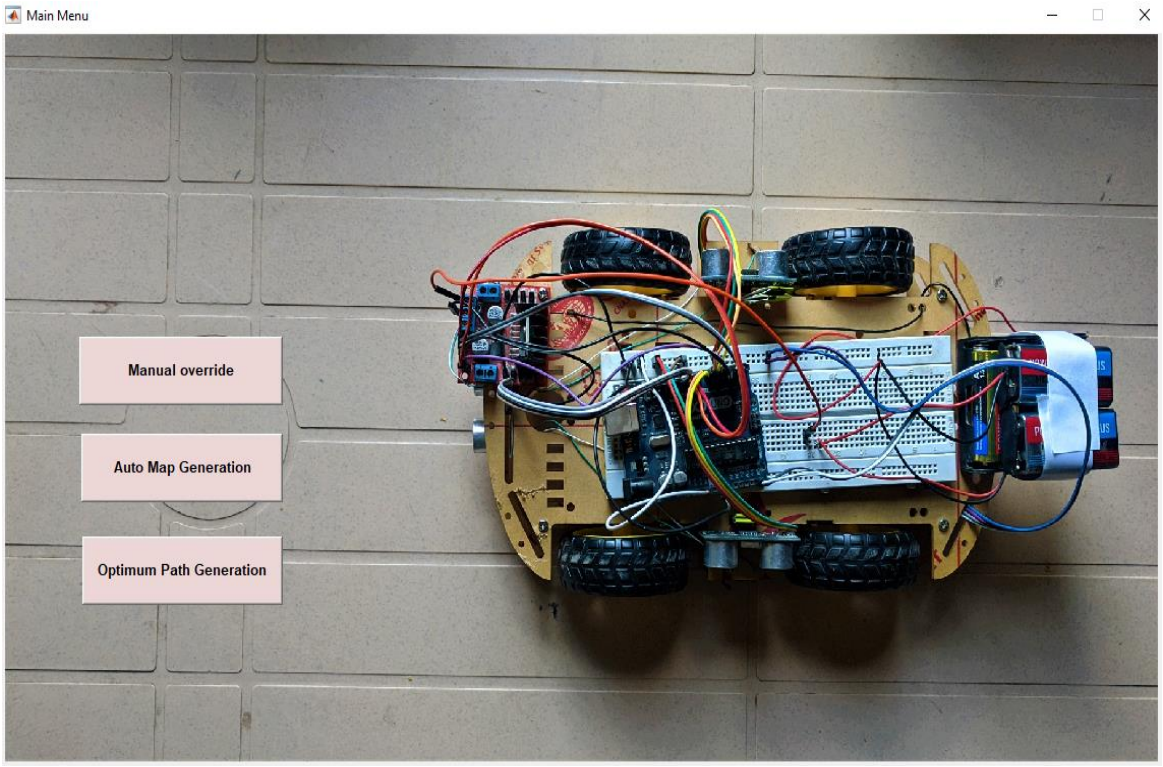


Figure 18 - Main Menu Screen

Manual Override Module

This is the module which gives user the power to control the movements of the robot through keys. Every action is taken by user and once user is satisfied with the map progress he/she can stop the path generation process

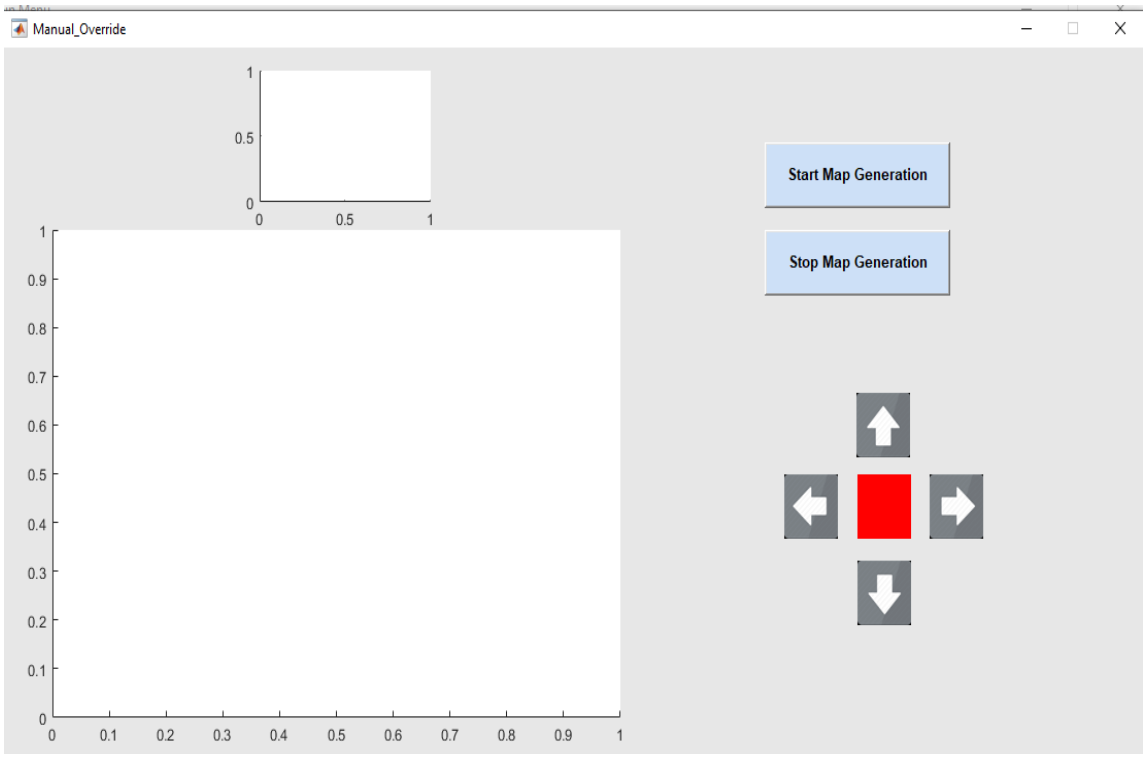


Figure 19 - Manual Override Screen

Auto Map Generation Module

This module allows the robot to make a map of the environment autonomously and user can stop the map generation when he/she progress is satisfactory.

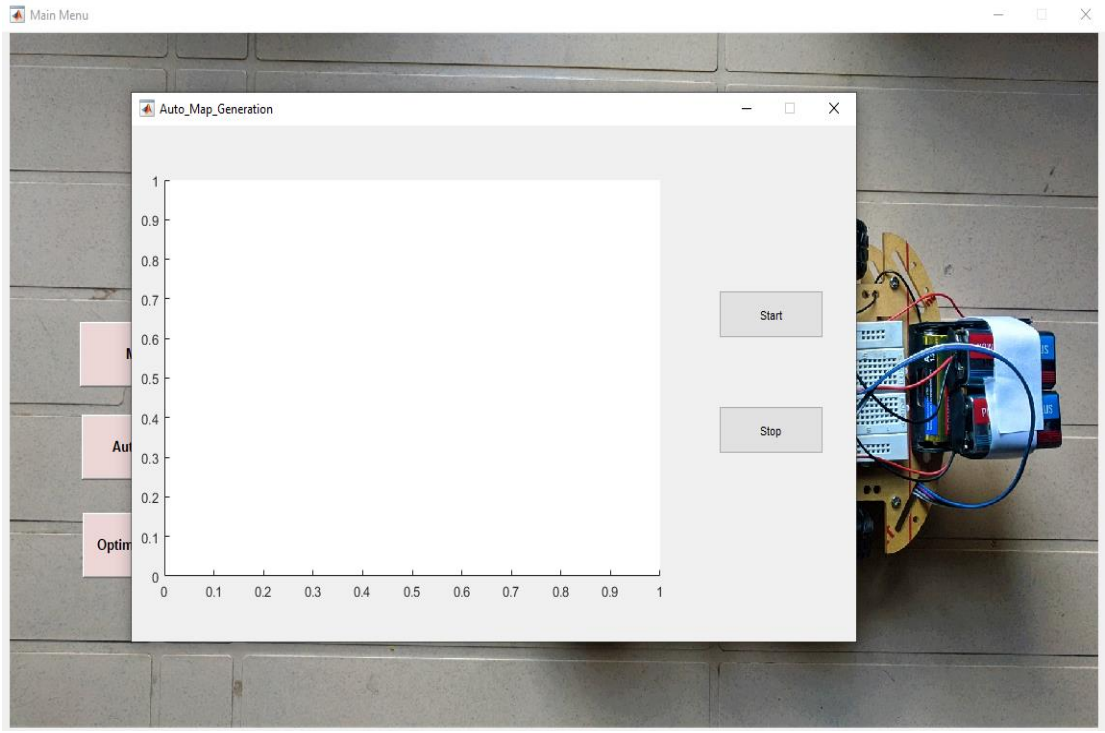


Figure 20 - Auto Map Generation Screen

Optimum Path Generation Module

This module allows the user to find out the best available path of a generated map.

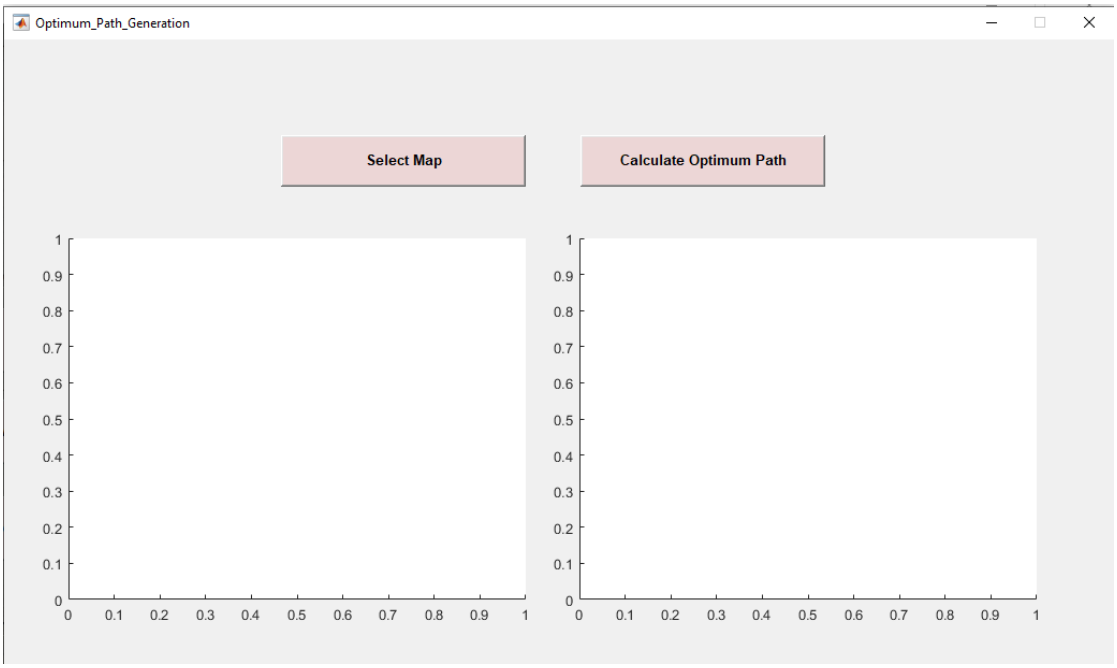


Figure 21 - Optimum Path Generation Screen

Chapter 6

6 System Testing

6.1 Overview

Testing of software projects include different levels of testing to ensure that the software which is being developed is error and fault free. The different levels at which testing was performed is argued here.

6.2 Unit Testing

It includes the testing of each module at completion.

6.2.1 Login Feature Testing

Test Case Name:	Application Login Feature testing
Test Case ID:	1
Description:	This feature requests the user to enter his/her credentials for login. If not already signed up, user can sign up. This test case aimed to checked that feature works conferring to user requirement
Testing technique used:	Black Box Testing
Preconditions:	System is running and linked to database.
Input values:	Username String Password string
Valid Inputs:	Valid or legal username Valid or legal password
Steps	Enter Email Enter password Tap SIGN IN button

Expected Output	The user identifications will be passed to the server for verification. The valid users will be directed to Main menu after login.
Actual Output	Successful login. User is directed to home screen.
Status	PASS

Table 3 - Login Feature Testing

6.2.2 Manual Override

Test Case Name:	Manual Override Testing
Test Case ID:	2
Description:	This feature allows the user to manually control the
Testing technique used:	Black Box Testing
Preconditions:	System is executing and linked to database.
Input values:	UP key Down Key Right Key Left Key
Valid Inputs:	Single Key Pressed
Steps	Tap Button from Valid Inputs

Expected Output	Robot Moves accordingly
Actual Output	Robot Moved accordingly
Status	PASS

Table 4 - Manual Override Testing

6.2.3 Auto Map Generation

Test Case Name:	Auto Map Generation Testing
Test Case ID:	3
Description:	This feature makes the robot to generate the map autonomously
Testing technique used:	Black Box Testing
Preconditions:	System is running, user is logged in and linked to database.
Input values:	Tap the Start Button
Valid Inputs:	Start button is clicked

Steps	Tap Start Button Wait for Map Generation Tap Stop
Expected Output	2D Map will be generated
Actual Output	2D Map Generated
Status	PASS

Table 5 - Auto Map Generation

6.2.4 Optimum Path Generation

Test Case Name:	Optimum Path Generation feature Testing
Test Case ID:	6
Description:	This feature finds the optimum path available in the map
Testing technique used:	Black Box Testing
Preconditions:	System is running and linked to database. Map is pre generated
Input values:	Tap Find Optimum Path
Valid Inputs:	Find Optimum Path Button is clicked

Steps	Click on Find Optimum Path Click Start
Expected Output	Optimum will be shown
Actual Output	Optimum Path Generated.
Status	PASS

Table 6 - Optimum Path Generation Testing

Chapter 7

7. Conclusion and Future Work

7.1 Conclusion

We acquired the objectives of the project and successfully developed a Robot that lets us make a virtual map of the environment to help civil engineers/rescue teams and military.

Due to the limitations of less time and small team size, the scope of the project was kept small. However, in the future flag markings can also be done on the maps as to have an analysis of which areas are hit by what kinds of different problems.

We firmly believe that our project can genuinely bring about a significant change in the lives of people.

7.2 Future work

Due to time and team size constraints, some of the things have to omitted from the scope of this project for its timely completion. However, this leaves room for a lot of future enhancements and improvements.

First of all, at the moment the Project only caters for a small area and distance which can be enhanced by adding advanced hardware and updating the software.

At the moment, the system only caters for one robot with one system. It can be further upgraded to multiple robots working in collaboration hence minimizing time and increasing efficiency.

Currently, the robot can only create 2D Map of the environment but it can further be enhanced to make a 3D Map. Moreover obstacles can be identified by category and others factors like atmospheric condition i.e. Temperature, humidity, light intensity and noise level all could also be integrated.

Among minor changes, the application’s User Interface could be modified to be even more user friendly and the application could be improved to run faster and enhance performance on lower end devices.

7.3 Glossary

API	Application Programming Interface
App	Application
AS	Assumption
Black box Testing	Testing emphasizes on the external behaviour of the software entity
CO	Constraints
App	Application
CEO	Chief Executive Officer
DBMS	Database Management System
DEP	Dependency
FRs	Functional Requirements
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
MCS	Military College of Signals
NFRs	Non Functional Requirements
NUST	National University of Science and Technology
OE	Operating Environment
OS	Operating System
REQ	Requirement
SR	Safety Requirements
SRS	Software Requirements Specification
UD	User Documentation
UML	Unified Modelling Language
White Box Testing	Testing emphasizes on the internal behaviour of the software entity

Table 7 – Glossary

7.4 Bibliography

https://www.coursera.org/

https://www.quora.org/

https://bytehost.com

https://www.w3schools.com

https://www.mathworks.com/

https://www.arduino.cc/

https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping

https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm

Durrant-Whyte, H.; Bailey, T. (2006). "Simultaneous localization and mapping: part I". IEEE Robotics & Automation Magazine. ***13*** *(2): 99–110. CiteSeerX 10.1.1.135.9810. doi:10.1109/mra.2006.1638022. ISSN 1070-9932.*

Cadena, Cesar; Carlone, Luca; Carrillo, Henry; Latif, Yasir; Scaramuzza, Davide; Neira, Jose; Reid, Ian; Leonard, John J. (2016). "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". IEEE Transactions on Robotics. ***32*** *(6): 1309–1332. arXiv:1606.05830. Bibcode:2016arXiv160605830C. doi:10.1109/tro.2016.2624754. hdl:2440/107554. ISSN 1552-3098.*

Perera, Samunda; Barnes, Dr.Nick; Zelinsky, Dr.Alexander (2014), Ikeuchi, Katsushi (ed.), "Exploration: Simultaneous Localization and Mapping (SLAM)", Computer Vision: A Reference Guide, Springer US, pp. 268–275, doi:10.1007/978-0-387-31439-6_280, ISBN 9780387314396

Karlsson, N.; et al. (Di Bernardo, E.; Ostrowski, J; Goncalves, L.; Pirjanian, P.; Munich, M.) (2005). The vSLAM Algorithm for Robust Localization and Mapping. Int. Conf. on Robotics and Automation (ICRA). doi:10.1109/ROBOT.2005.1570091.

Appendix A. Glossary

.ino:	Adruino file extension
.m:	Matlab file extension
Matlab:	Numerical computing and proprietary programming language
Adruino:	Used to write and upload programs to Arduino compatible boards
Static Analysis:	Analyzing without executing the code
Dynamic Analysis:	Analyzing behavior while executing the code

Appendix B: Issues/Limitations

All possible issues have already been mentioned where required in the SRS. Any remaining ones are listed below:

1. The group shall try to match the features and NFRs as best as possible, however, like all software projects, any discrepancies are apologized for at this stage.
2. Feedback on requirements is expected from the users to help the group in improving the design and implementation of the project.

Plagiarism Report