Universität Rostock

Traditio et Innovatio

Faculty of Computer Science and Electrical Engineering
Institute of Automation

Master Thesis

A Project Report On

"Database Programming for Mobile Robot"

By

B. Chaitanya Krishna Reddy

Matrikel Nr: 209208515

Mentors: Prof. Dr. Norbert Stoll

Dr. –Ing. Steffen Junginger

Dipl. –Ing. Hui Liu

Rostock 08.10.2012

Universität Rostock Traditio et Innovatio

Universität Rostock | Institut für Automatisierungstechnik
18119 Rostock, R.-Wagner-Str. 31 / Haus 8

Master Thesis
For Mr. Chaitanya Krishna Reddy Bommareddy

"Database Programming for Mobile Robots"

For a research project in laboratory automation an automated mobile robot control system is to be designed. In the system an independent database will be developed to store and process all the key data for multi mobile robots. Master Thesis work includes database design, operation functions programming and performance test.

Details:
- Introductory training on mobile robot and MySQL database
- Concept development for robot database architecture
- Developing of database interface functions for storing, deleting, updating and querying of sensor data
- Developing of database interface functions for doing necessary statistical calculation based on the stored robot data (including task execution, movement position and battery)
- Database and operation software testing.

Place of work is IAT Rostock, Warnemünde. Master Thesis is to be delivered in MS Word 2007 format or higher.

Day of Beginning: 02.04.2012

Mentors: Prof. Dr. Norbert Stoll
Dr.-Ing. Steffen Junginger
Dipl. Ing. Hui Liu

…….
Institutsdirektor

…….
Prüfungsausschuss

Institut für Automatisierieungstechnik

Direktor:

Prof. Dr.-Ing. Habil Kerstin Thurow

Universität Rostock
Fakultät für Informatik und Elektrotechnik

Postadresse:
R.-Wagner-Str. 31
18119 Rostock

Fon +49(0)381 498-7701
Fax +49(0)381 498-7702

Kerstin.Thurow@uni-rostock.de
http://www.iat.uni-rostock.de

Dedicated To

అమ్మ, నాన్నగారు & అక్క ...!!!

(Mom, Dad and Sister)

"Families are the compass that guides us. They are the inspiration to reach great heights, and our comfort when we occasionally falter" - Brad Henry

Acknowledgement

I would like to express my gratitude to Prof. Dr. Norbert Stoll for his initiatives in organizing the project and giving me the opportunity to work at CELISCA, Institute of Automation (IAT).

I would like to gratefully and sincerely thank Dr.-Ing. Steffen Junginger, Dipl. Ing. Hui Liu for their guidance, understanding, patience, and most importantly, their friendship during my thesis studies and previous work as well at CELISCA. Their mentorship was paramount in providing a well-rounded experience. The encouragement given to me by them not only made me a better person technically but also humanly. For everything you've done for me, I am always indebted to you, I thank you both heartily.

My sincere gratitude to Dr.-Ing. Thomas Roddelkopf and Dr.-Ing. Kai Rittersbuch who made me believe that I can do programming and all the other dear staff members at CELISCA.

All this would have never been possible and achieved, without the unconditional love and support from my parents BHK Reddy and BM Saroja, my elder sister B. Hima Bindu. I thank them for being my pride and morale.

My sincere obligation to all my dear friends for their enduring nature and for their thorough support throughout my life.

Table of Contents

Chapter 1: Introduction

1.1 Laboratory Automation 1

1.1.1 Advantages of Laboratory Automation 2

1.1.2 Applications of Laboratory Automation 2

1.2 Robots, Robotics and Laboratory Automation 2

1.3 H20 - The Humanoid 4

1.3.1 How H20 Works 5

1.3.1.1 Localization Landmarks and GPS Sensor 5

1.3.1.2 H20 Control Centre and Dr. Robot Localization Setup 6

1.3.2 Applications of H20 8

1.4 Database in Laboratory Environment 8

1.5 Project Necessity 9

1.5.1 Key Phases 10

1.6 Database Architecture 10

Chapter 2: Development Tools

2.1 Database Programming 12

2.1.1 MySQL 12

2.1.1.1 MySQL Workbench 14

2.1.1.2 HeidiSQL 15

2.2 Application Programming 17

2.2.1 Microsoft Visual Studio 2008 17

2.2.2 Microsoft Chart Controls 18

2.3 .NET Framework 19

2.4 C Sharp 20

Table of Contents

2.5 Structured Query Language (SQL) 21

2.6 ADO .NET 22

2.7 MySQL Connector 23

2.8 MySQL Namespaces 24

Chapter 3: Database and Software Development

3.1 Database Architecture 27

3.1.1 Two Level Client/Server Architecture 27

3.2 Database Design 28

3.3 Architecture and Design of Project Database 30

3.3.1 H20 Database Architecture 30

3.3.2 H20 Database Design 30

3.3.3 Attributes and Their Data Types 33

3.4 Database Programming 34

3.4.1 Functions Based On Data Manipulation 35

3.4.2 Functions Based On Search Conditions 35

3.4.3 Functions Based On Statistical Analysis 36

3.4.4 Functions Based On Graphical Representation 38

3.5 Software/Application Programming 39

3.5.1 Save Function 39

3.5.2 Establishing the connection 41

3.5.3 Data Binding 42

3.5.3.1 DataAdapter 45

3.5.3.2 DataSet and DataTable 46

Table of Contents

Chapter 4: GUI Designing and Testing

4.1 GUI Designing 50

4.2 Testing 60

4.2.1 Testing Process 60

Chapter 5: Conclusion and Future Work

5.1 Aim 64

5.2 Tasks Achieved 64

5.3 Conclusion and Future Work 65

List of Figures

Figure 1: Example for Laboratory Automation 1
Figure 2: Dispensing and Culturing by Mahoro 3
Figure 3: H20 from Dr Robot 4
Figure 4: Localization GPS Sensor using HLD1L Landmarks 5
Figure 5: Dr Robot H20 Control Centre 7
Figure 6: Dr Robot Localization/GPS Setup 7
Figure 7: Future Lab Concept 9
Figure 8: Project Architecture 10
Figure 9: MySQL 5.5 Command Line Client 13
Figure 10: MySQL Workbench 14
Figure 11: HeidiSQL Session Manager 16
Figure 12: HeidiSQL Main GUI 16
Figure 13: Microsoft Visual Studio 2008 Environment 18
Figure 14: Chart Types in Microsoft Chart Control 19
Figure 15: ADO .NET Object Model 23
Figure 16: Example of Two Level Client/Server Architecture 28
Figure 17: Example of Entity Relationship Diagram 29
Figure 18: H20 Two Level Client/Server Architecture 30
Figure 19: MobileRobotH20 Database with Multiple Entities 31
Figure 20: Entity Relationship Diagram for MobileRobotH20 Database 32
Figure 21: H20 Statistical Analysis 38
Figure 22: Saving into Database 42
Figure 23: Example for Simple Data Binding 43
Figure 24: Example for Complex Data Binding 44
Figure 25: Data Binding Hierarchy 44
Figure 26: Work Flow of User Interface 48
Figure 27: H20 Database Viewer UI 50
Figure 28: H20 Database UI 52
Figure 29: UI with Latest Information from Database 53
Figure 30: UI with available search conditions 54
Figure 31: UI with Battery One Voltage search ranges 55

List of Figures

Figure 32: UI with Battery Two Voltage search ranges 56
Figure 33: UI with Time related search conditions 57
Figure 34: Graph, Voltage vs. Time 58
Figure 35: UI with Working Area related search conditions 58
Figure 36: Data exported to Microsoft Excel 59
Figure 37: Printing of Data 59
Figure 38: MobileRobotH20 Database with Test Data 61
Figure 39: UI showing test data 61
Figure 40: Test data sorted by given Battery Voltages 62
Figure 41: Test data sorted by given time period 62
Figure 42: Graph between Voltage and time for test data 63
Figure 43: Statistical Analysis for Test data 63

List of Abbreviations

GUI – Graphical User Interface

RAM – Random Access Memory

SQL – Structured Query Language

H20 – Hawk 20

PTZ – Pan Tilt Zoom Camera

GPS – Global Positioning Sensor

MS – Microsoft

3D – Three Dimensional

VB – Visual Basic

DNA – Deoxyribonucleic Acid

XML – Extensible Mark-up Language

ODBC – Open Database Connectivity

OLE DB – Object Linking and Embedding Database

API – Application Programming Interface

Chapter 1: Introduction

1.1 Laboratory Automation

"Besides Magic, there is only automation and mechanization." – Federico Garcia Lorca

Automating a system is, in which a workplace or process has been converted to one that replaces or minimizes human labour with mechanical or electronic equipment. Automation can be viewed as a sign of quality as it indicates the degree of variance in a manufacturing process. Lower variance allows lower tolerances, better fit and finish.

Increased automation is a key for desired increased production. In the scope of industrialization, automation is a step beyond mechanization [1]. Mechanization needs human assistance with the machinery and their muscular efforts, but whereas automation greatly reduces the need for human efforts. Almost every process and system in a laboratory can be automated. Automation plays significant role in the global economy and in daily experience with its increased capabilities.

The main aim of todays various laboratories, are helping mankind when needed, in less time and to provide less expensive help. This can be achieved through laboratory automation itself.

Fig.1: Example for Laboratory Automation [2].

Chapter 1: Introduction

1.1.1 Advantages of Laboratory Automation

There are several advantages of automation in laboratory environment and in general. Few of them are

- Decrease in part cycle time
- Improved quality and reliability
- Better floor space utilization
- Reduced production cost
- Economical

1.1.2 Applications of Laboratory Automation

In present day, as known, laboratories are being widely automated. There are several important applications of Laboratory Automation. Few of them are

- Drug discovery
- DNA Sequencing
- High content screening
- Forensic applications
- Nanodispensing
- High throughput screening
- Transportation

1.2 Robots, Robotics and Laboratory Automation

"Robota, The subordinate labour" – Karel Čapek

Though the "concept" was clearly established by many historical realizations, to materialize, the "concept" had to await the advent of its underlying technologies during the course of 20th century.

Chapter 1: Introduction

Medial 20th century, period when the initial relations between human intelligence and machines were understood, ignited full-fledged research in Artificial Intelligence. Over this period, advancement in different technologies helped in advancement of Robots.

This virtuous circle over time produced that knowledge and understanding which gave birth to the field of *Robotics*, properly referred to as the science and technology of robots [3]. By the dawn of new millennium, Robotics has been widely expanded by advancement in its related technologies.

Laboratory robotics is being increasingly applied in pharmaceutical development to help meet the needs of increasing productivity, decreasing drug development time and reducing costs. For example, Microarray technologies use a robotic spotting device or "Microarrayer" to print DNA sequences onto a solid support, typically glass slides or membranes [4]. And there are also several pick and place robots which help in safe and faster transport of arrays, vials and tube racks. Using of Robots and Robotics in laboratory automation lead to faster and safer production capability, which in turn can be very economical.

Fig.2: Dispensing and Culturing by Mahoro [5].

Chapter 1: Introduction

1.3 H20 – The Humanoid

H20, the 3rd generation Hawk robot from Dr Robot®, is a wireless networked autonomous humanoid mobile robot. H20 is built on i90 base from Dr Robot® itself, which is ready to use mobile robot platform developed for applications such as remote monitoring, tele-presence and autonomous navigation.

The Key features of H20 are:

- 12" touch screen tablet on chest
- Dual arms with 6 joints
- 6 DOF animated head with dual camera
- Auto docking and recharging station
- Fully wireless networked (802.11g)
- Navigation Sensors
- High resolution PTZ camera
- Vision-landmark based indoor localization

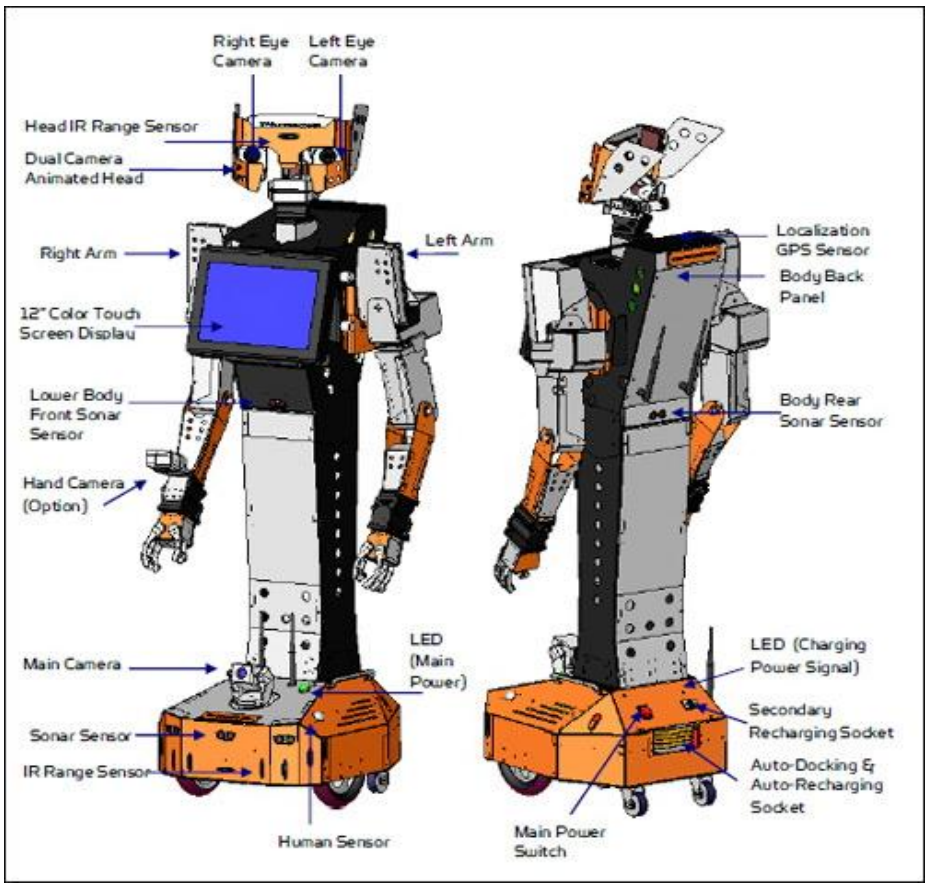


Fig.3: H20 from Dr Robot® [6].

1.3.1 How H20 works!!!

1.3.1.1 Localization Landmarks and GPS Sensor

H20 has state of the art indoor navigation system, built on localization landmarks. The localization GPS sensor present on the H20 is capable of recognizing these landmarks and plan its patrol path or charge path accordingly. The HLD1L type localization landmarks are capable of covering an area over 160 m^2. These landmarks should be placed on ceiling of height between 2.3m to 4.1m in the environment where H20 would be working. For ceilings of other heights, other types of landmarks such as HLD2L can be used.

The localization GPS sensor present on H20 is an installed Hagisonic StarGazer Robot Localization System by Hagisonic CO.LTD. It is a unique solution for indoor localization of mobile robots as H20. The IR projector present in the StarGazer Robot Localization System shoots Infrared rays on to the landmarks present on the ceiling which in turn reflect the rays to the StarGazer. And through further immediate processing StarGazer calculates the position and angle of H20.

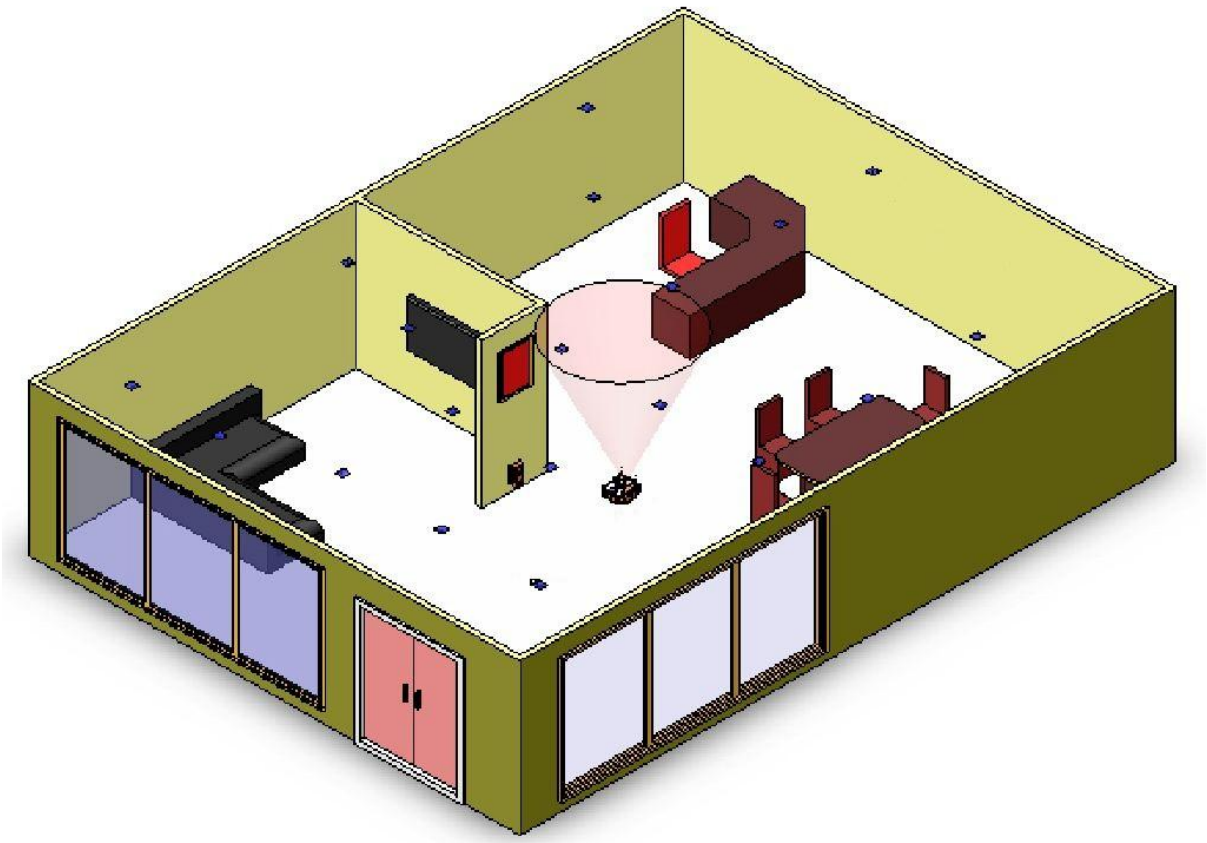


Fig.4: Localization GPS Sensor using HLD1L landmarks [17].

1.3.1.2 H20 Control Centre & Dr Robot Localization – GPS Setup

H20 Control Centre is proprietary software from Dr Robot® used for overall control of H20. It is also has a user friendly GUI (Graphical User Interface) where user can get the important information about H20 as battery information, sensory information, robot position, power status, live feeds from on-head cameras and main PTZ(Pan-Tilt-Zoom) camera. And also user can control i.e., move the robot using the joystick control option. This GUI also empowers user with capability of commanding H20 to GoCharge and AutoPatrol.

H20 Control Centre also starts few service programs as Dr Robot Motion & Power Service, Dr Robot File Share Service, Dr Robot Head Control Service and Dr Robot Arm Control Service. Every service has further functionalities where user can manipulate with H20 and also plan new paths for patrolling and charging. H20 also has Dr Robot H20 Remote Control software for remote monitoring and tele-operation with most of the functionalities as control centre has.

The speciality of H20 Control Centre is it has two watch-dog features which are very important for healthy working of H20 and long battery run. They are

- H20 Control Centre will automatically reset H20 if it does not receive any GPS information for 50 seconds while it is performing AutoPatrol or GoCharge tasks.

- If in any case H20 loses communication with Control Centre, it will enter stand-by mode and wait to reconnect.

Dr Robot Localization – GPS Setup is proprietary software from Dr Robot® used for mapping of the landmarks once they are attached to the ceiling of environment where H20 would be working. Mapping is important because it brings positional and orientation information from the localization sensors. Mapping of localization landmarks is a four step process which has several intermediate processes.

The positional and orientation information of whole H20 are calculated depending on the information from GPS sensor position data.

$\theta_{Robot} = 180 - \theta_{GPS}$, if $\theta_{GPS} \in (0,180)$
$\theta_{Robot} = -180 - \theta_{GPS}$, if $\theta_{GPS} \in (-180,0)$

$$\begin{bmatrix} X_{Robot} \\ Y_{Robot} \end{bmatrix} = \begin{bmatrix} 0.126 * Cos(\theta_{Robot}) + X_{GPS} \\ 0.126 * Sin(\theta_{Robot}) + Y_{GPS} \end{bmatrix}$$

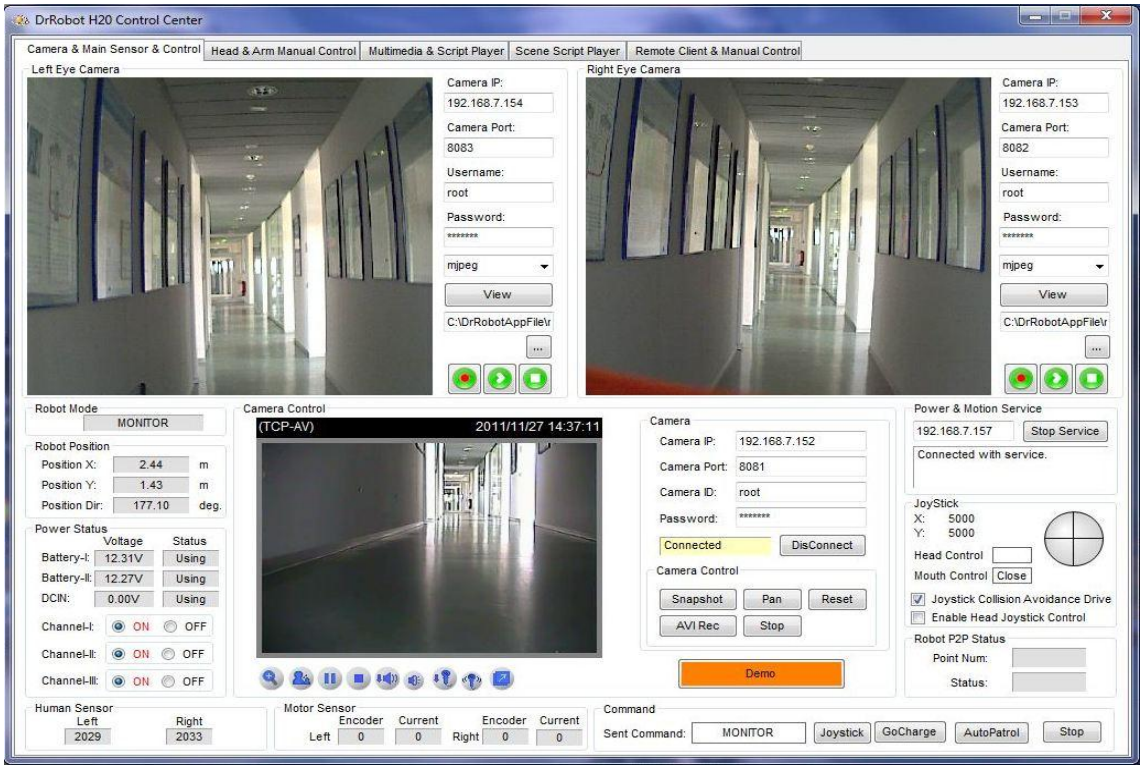


Fig.5: Dr Robot H20 Control Centre.

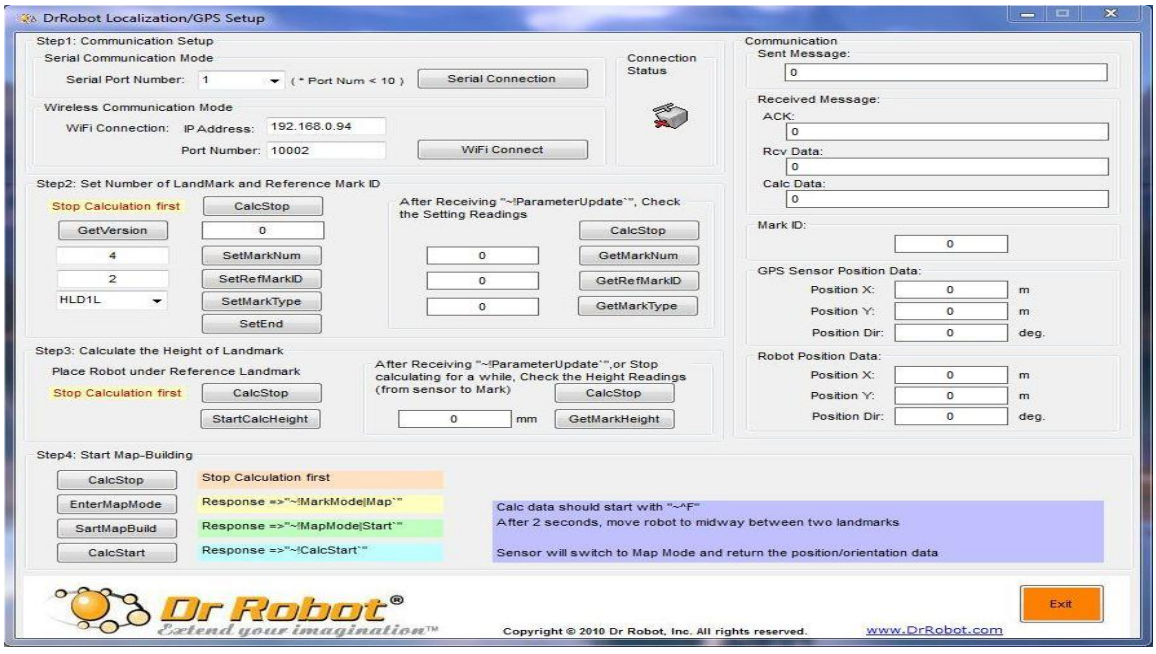


Fig.6: Dr Robot Localization/GPS Setup.

1.3.2 Applications of H20 in Laboratories and World

H20 has and would have many applications in near future. Few of the important applications are

- Can be used for surveillance.
- Can be used for transportation in laboratories.
- Can be used for remote monitoring and tele-operations.
- Can be used as guide in Museums, Showrooms and other Public places.
- Can be used as development platform for Research and Academic purposes in institutions.

1.4 Database in Laboratory Environment

Database, a collection of organized data mostly in digital form [7]. It plays a very important role in today's organized world where every detail has to be saved for future reference. Though not recognized by the user, it is Database which saves every information about a particular event of his. In 21st century, database is used almost everywhere, in shopping malls, hospitals, airports, educational institutions and etc.

Similarly, in laboratories also Database has great importance. It can be used to store the task log performed on various biological and chemical samples and also the results. It can keep records of errors that occur while working.

Many trial and reference laboratories lack adequate material management system. Considering this as a challenge many study and research groups have developed a database system that allows exchange of database between laboratories and trial coordination centres [8].

Today there are several dedicated database systems designed and developed for laboratories as BOXIT, a Database for Molecular Biology Laboratories and MEB-Lab Database holding information about laboratories capable of measuring exposure biomarkers.

1.5 Project Necessity

"Database Programming for Mobile Robots"

The main aim of this project is to develop an efficient and adequate database for Mobile Robots, which in this case is H20. And this independent database should be capable of storing and processing all the key data for multiple mobile robots.

As a part of future work, Future Lab, a concept owned by CELISCA, H20 would be working inside the laboratories and performing many tasks over time. The major task would be transporting laboratory equipment and apparatus, for example Microtiter Plates, between different working stations.

Though an easy task for humans, it is a challenging task for H20 and other mobile robots as well. There are several parameters that have to be regularly monitored for efficient working of H20 as Battery Information, Sensory Information and etc. And also other task regarding information of H20 can be saved in database.

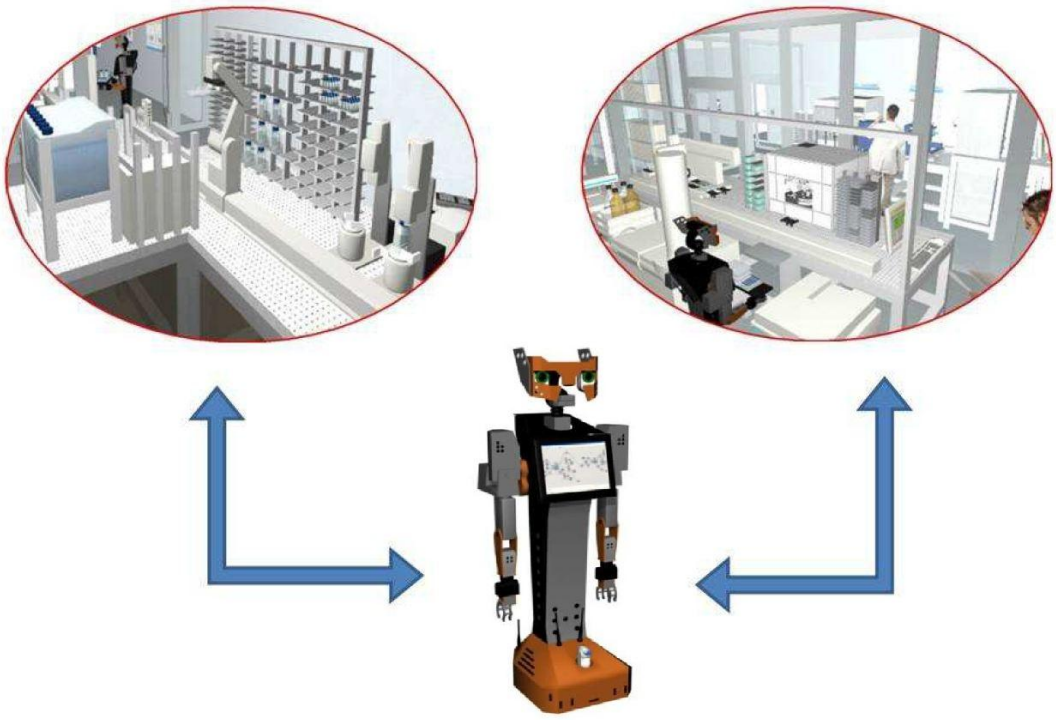


Fig.7: Future Lab Concept

1.5.1 Key Phases in Project

There are several phases included in this project. And the key phases of all are

- Concept development for robot database architecture
- Developing of database interface functions for sensory data
  - Storing
  - Deleting
  - Updating
  - Querying

- Developing of database interface functions Statistical calculations of stored data
  - Average Charge Time
  - Average Discharge Time
  - Number of Tasks performed

1.6 Project Architecture

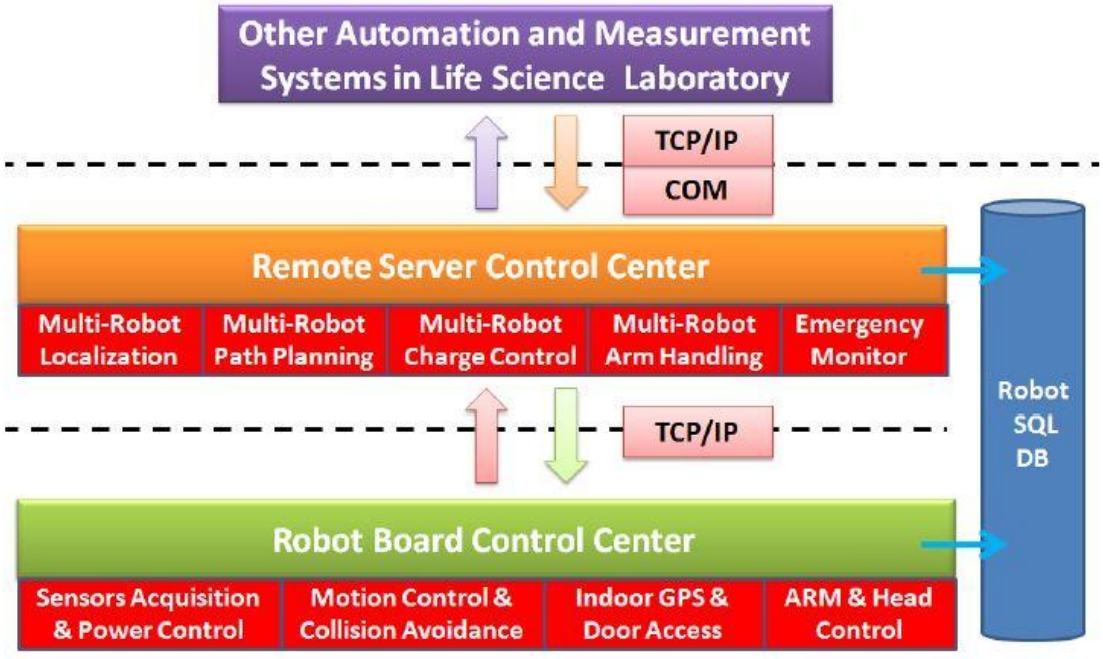


Fig.8: Project Architecture [9].

Chapter 1: Introduction

As shown above in Fig. 8, the main aim of this project is to store data into database. The data could be anything, beginning from positional information to sensor information. The later target after storing this data into a database is to make it accessible to user through a (GUI) graphical user interface.

They way these targets were achieved, are discussed in the following chapters. Where we come across through different development tools used for the project, the original work, application presentation and validation of the software with test data.

Chapter 2: Development Tools

"Right tools in Right hands brings Right results"

In recent times, almost for every project we need certain software tools for programming as every project has at least a little part of it. In my Master Thesis project, I also have used several programming tools. And all of them are briefly discussed in this chapter.

In this project as Database programming and Application programming are the main things done, I would like to categorize the software's used according to them.

Database Programming	Application Programming
MySQL	Microsoft Visual Studio 2008
MySQL Workbench	Microsoft Chart Controls
HeidiSQL	

And the languages, framework, connectors and namespaces that helped this software's working are discussed later in this chapter.

2.1 Database Programming:

Database programming means that with use of various tools, managing huge data which are vital information. In this project database programming plays an important role. And the vital information in this project are information from H20 related to Power, Sensors and Tasks. Several SQL queries were used in this project in order to obtain desired results from the data stored. As discussed earlier the tools that were used in this project for database programming are MySQL and their frontend user interfaces. In detail,

2.1.1 MySQL:

MySQL, is considered as world's most used open source relational database management system (RDBMS) that runs as a server providing multi-user access to several databases [10]. Written in C and C++ languages, works on several popular present day operating systems as Microsoft Windows, Solaris, Macintosh OS X, Linux and etc. And many of the Fortune 500 companies as Face book, Google, Adobe and etc., use MySQL as their database management system.

And especially MySQL on windows has several performance related advantages. MySQL provides various tools useful for developing and managing MySQL based Windows applications.

Main reasons behind using MySQL over the other are as follows,

- Scalability and Flexibility
- High Performance
- High Availability
- Strong Data Protection
- Management Ease

And the MySQL programming can be done using MySQL Command Line Client,

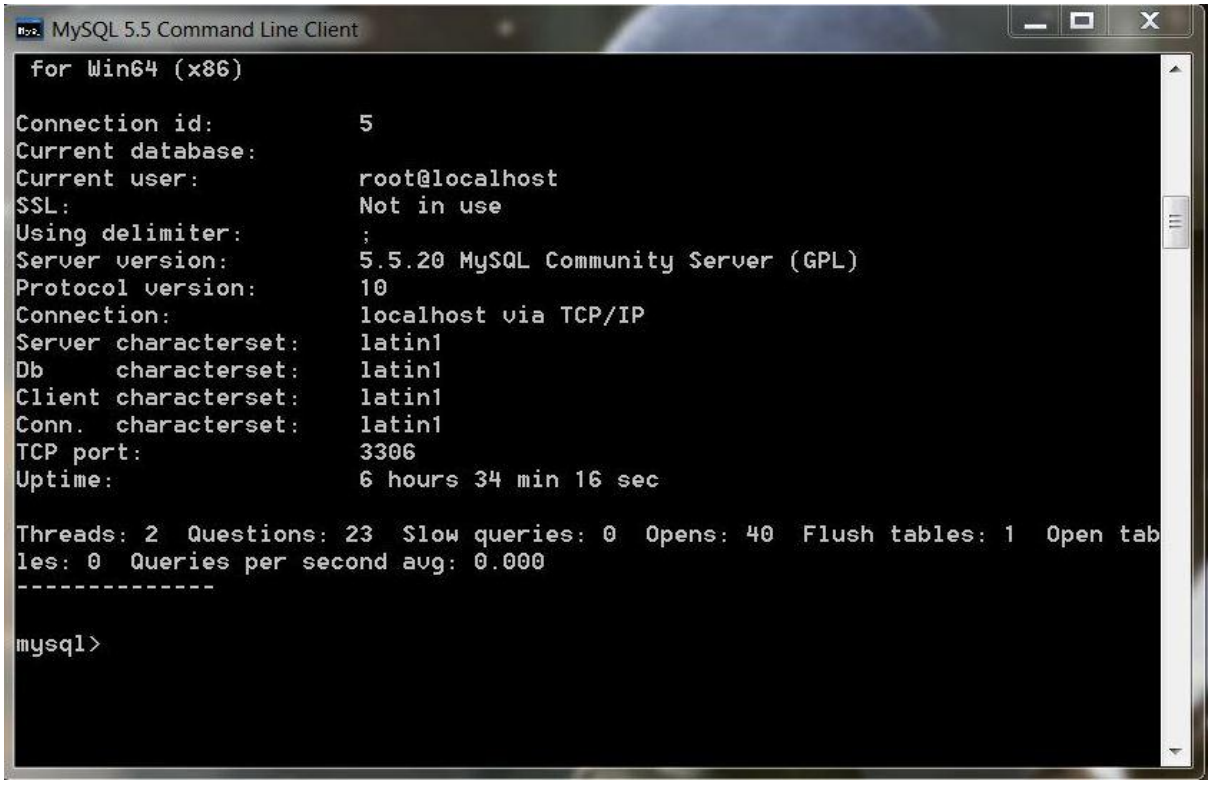


Fig.9: MySQL 5.5 Command Line Client.

However, as MySQL, a relational database management system, comes with no frontend graphical user interface system for managing the database. But there are several frontends and command line clients developed by MySQL themselves and third parties, serving the purpose. And few important of them and those used in this project are the following,

2.1.1.1 MySQL Workbench

MySQL Workbench, developed by Oracle and MySQL AB, is a graphical user interface for managing the database. Also referred as visual database design tool enabling design of database structure. It provides graphical tool for working with databases and servers containing number of them.

The three main functionalities of MySQL Workbench are:

- SQL Editor
- SQL Development
- Data Modeling
- Data Administration
- Server Administration

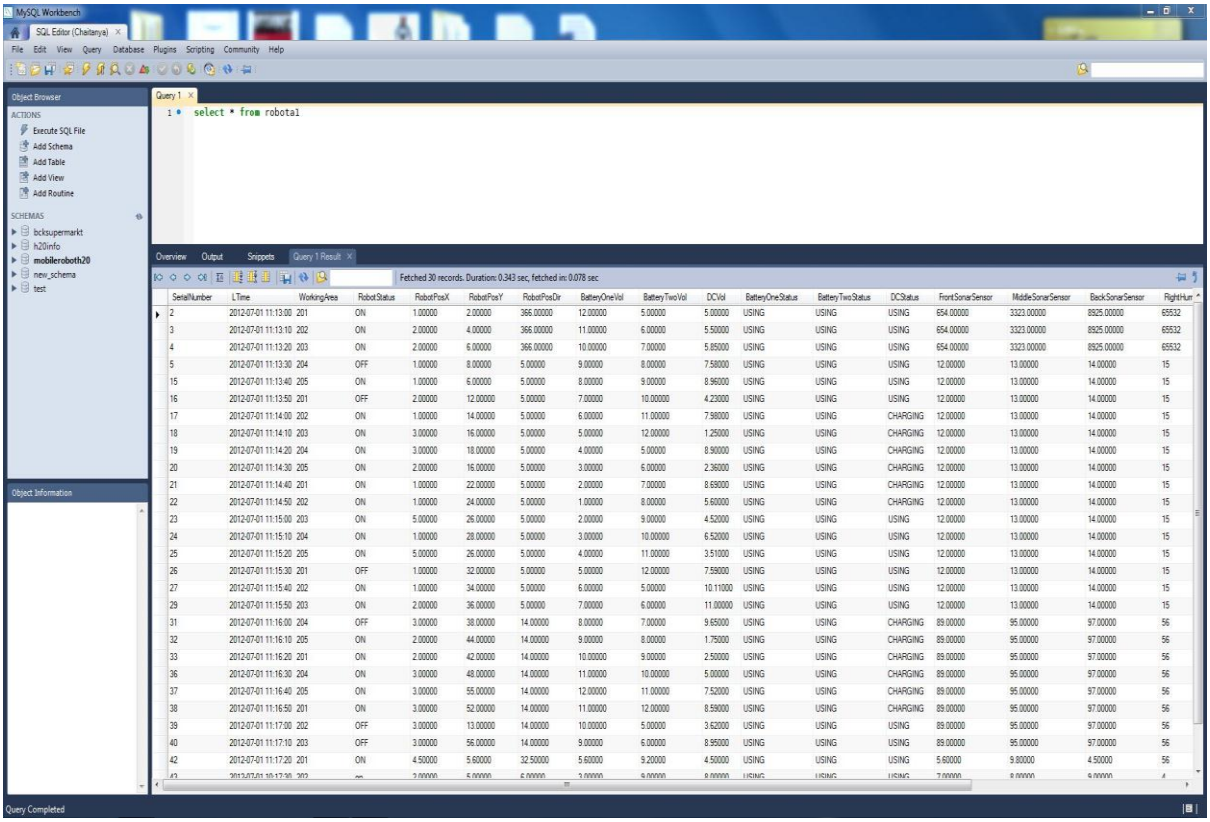


Fig.10: MySQL Workbench.

2.1.1.2 HeidiSQL

HeidiSQL is a free and open source frontend, developed for managing databases created using MySQL. It is a full featured front end written in Object Pascal that runs on machines with Windows operating systems. It is capable of connecting locally and remotely to MySQL servers, managing the data. It is also capable of other important functions used for data manipulation as adding, deleting, saving and sorting data.

As MySQL Workbench, HeidiSQL also has few important capabilities, which are as follows,

- Databases
- Tables
- Views
- Triggers
- Procedures
- Events
- Server Host
- Server Connection

Unlike MySQL, HeidiSQL needs a local or remote MySQL login with valid credentials to manipulate data. Once connected, session is created, within which a user can manage database and work with it. When done, session ends. User has to login again whenever HeidiSQL is needed. Whereas in MySQL Workbench, user needs no login as it is already associated with MySQL.

As HeidiSQL is under active development there are few connection issues that should be taken care of. And also complex conjoined statements would not return any result as MySQL Workbench would do. But at the same time HeidiSQL has its own advantages, for example, viewing a table is a matter of clicks in HeidiSQL whereas in MySQL Workbench a command statement has to be written. It has very user friendly frontend interface.

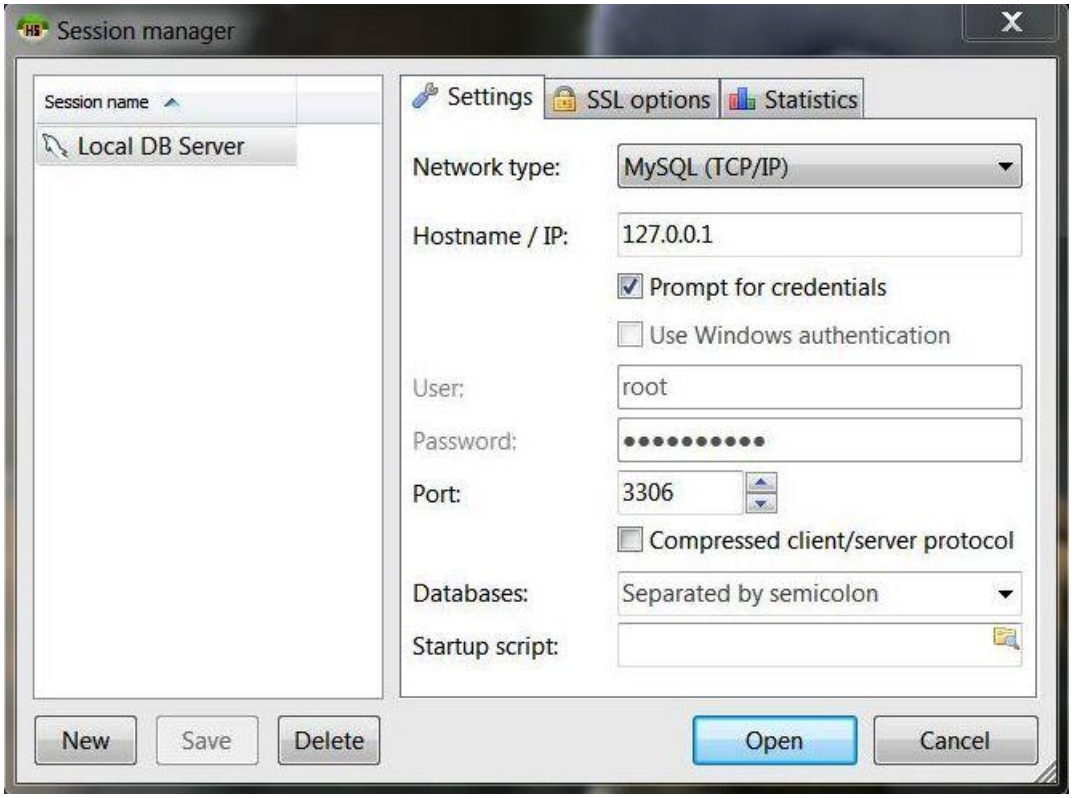


Fig.11: HeidiSQL Session Manager.

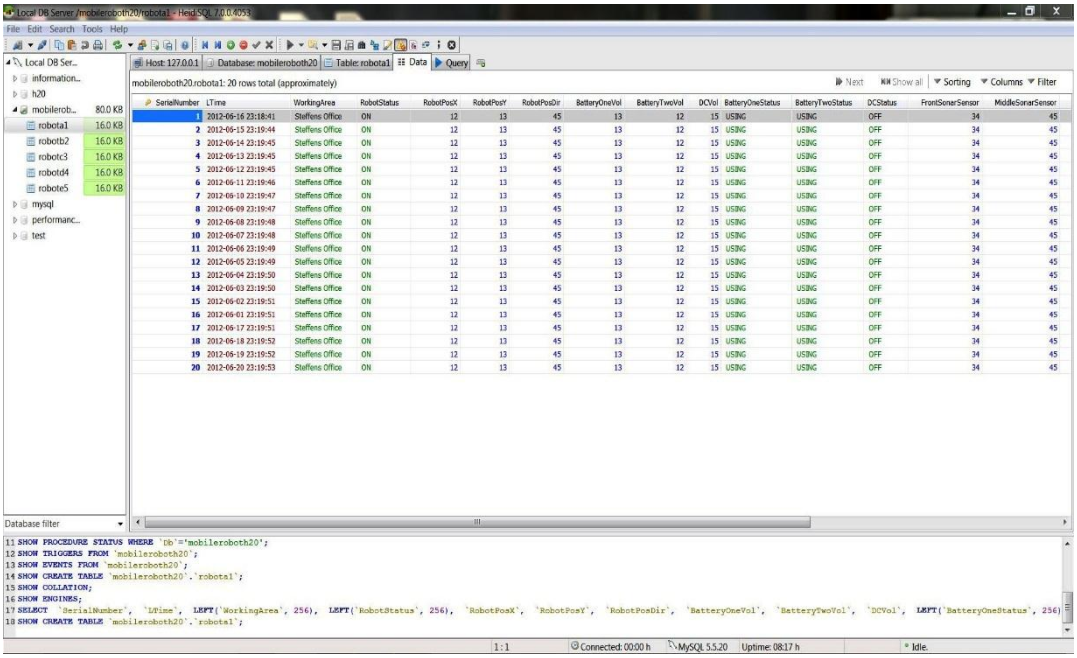


Fig.12: HeidiSQL Graphical User Interface.

Chapter 2: Development Tools

2.2 Application Programming

Application programming can be defined as any program written or designed to perform specific functions on the order of user or connect to other programs and execute desired function. In general application programming uses the services provided by operating systems and other supporting programs. Almost every application depends on the underlying API (Application Programming Interface) provided by the operating systems to carry out even the basic functions as opening a file, deleting a file and etc.

In this Master Thesis Project, Database Programming for Mobile Robots, application programming for database is done using the following development tools,

2.2.1 Microsoft Visual Studio 2008

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft [11]. There have been several versions of Microsoft Visual Studio, varying from Visual Studio to Visual Studio 2012 with different code names and supporting different .NET Framework versions. It is capable of developing various console and graphical user interface applications. It is also used to develop Windows Forms applications, web sites, web applications, and web services in both native codes together with managed code for all supporting platforms.

Microsoft Visual Studio 2008 version runs on .NET Framework 3.5. In this project applications are developed using Microsoft Visual C#, included in Microsoft Visual Studio 2008. The advantages of C# in whole application programming environment are as follows,

- Automatic Memory Management
- Coherent Libraries
- Smaller Memory Footprint
- Language Complexity
- Compilation Model

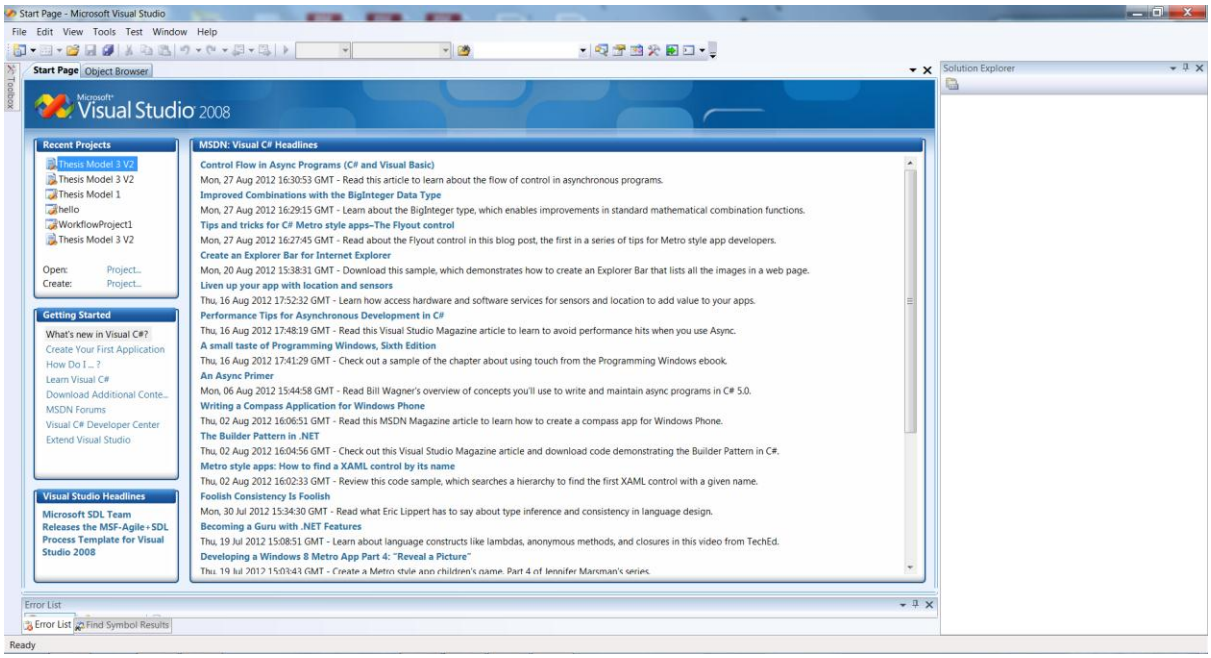


Fig. 13: Microsoft Visual Studio 2008 Environment

2.2.2 Microsoft Chart Controls

Microsoft Chart Controls are capable of simple, intuitive and visually compelling charts for statistical analysis. It is a rich chart control released by Microsoft for .NET web and Windows form applications. Chart controls can be easily customized according to the necessity. Few important features of MS Chart Controls are as following,

- Easy plot of complex data
- 3D Visualization Control (perspective, angle, rotation)
- Image Map Selection
- Custom Animated Tooltips
- Capture of Mouse Events

In this thesis project, MS Chart Controls were used to plot line graphs between a selected set of data. In programmer point of view, there are few added advantages with MS

Chart controls and their methods. Methods like PrintOut, Refresh etc., gives the programmer ease in programming.

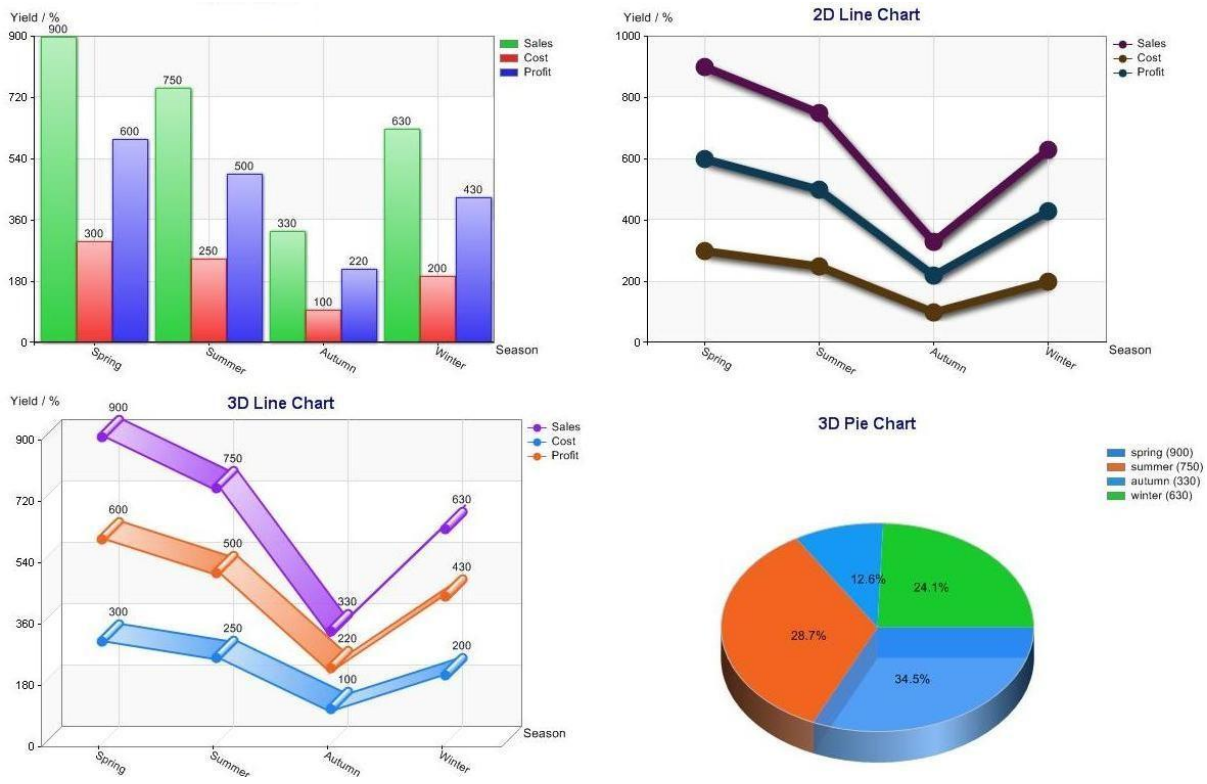


Fig.14: Chart Types in Microsoft Chart Controls

2.3 .NET Framework

The .NET Framework is an application developing platform introduced by Microsoft. The key inspirations behind the .NET Framework is it's envisioned to use as a means of incorporating incongruent operating systems[19]. A wide variety of application can be created using .NET framework on windows platform. It is capable of creating

- Web Applications
- Office Applications
- Windows Applications
- Work Flow Applications
- Smart Device Applications
- Database Applications (SQL Server)

Chapter 2: Development Tools

.NET programming can be done using many languages, beginning from C#, C++ to old languages as COBOL and etc. And the main advantage of their integration into .NET is they can communicate with each other.

People are always curious to know what is inside .NET Framework. It is a vast library where it has all the codes written for the objects and methods that are used in object oriented programming. And this library, depending upon the language and the applications used, is subdivided into small modules. Some of them are suitably further distributed into smaller sub modules. .NET Framework contains a Common Language Runtime (CLR), which is responsible of everything that happens to an application.

Developing an application using .NET means, using the objects that are provided by the framework, which in turn have pre-written code. .NET Framework 3.5 has been used in this project which is included in Visual Studio 2008. There are many advantages of using .NET Framework, and few of them are

- Direct Support for Security
- Consistent Programming Model
- Simplified Development Efforts
- Easy Application Deployment and Maintenance

2.4 C Sharp

C Sharp (C #) can be referred as a simple, model-purpose, general object oriented programming language. It is very useful in creating software components like this thesis project. Programmers familiar with languages like C++ and Java can easily learn C #. And C# is very usable for embedded and large system programming.

The major advantage of C# is, it comes with the .NET Framework API. And as known .NET framework comes with a package of large set of classes, TCP/IP socket programming, and graphics. Thus Developers can partly write in C# and another part in another .NET language (e.g. VB .NET).

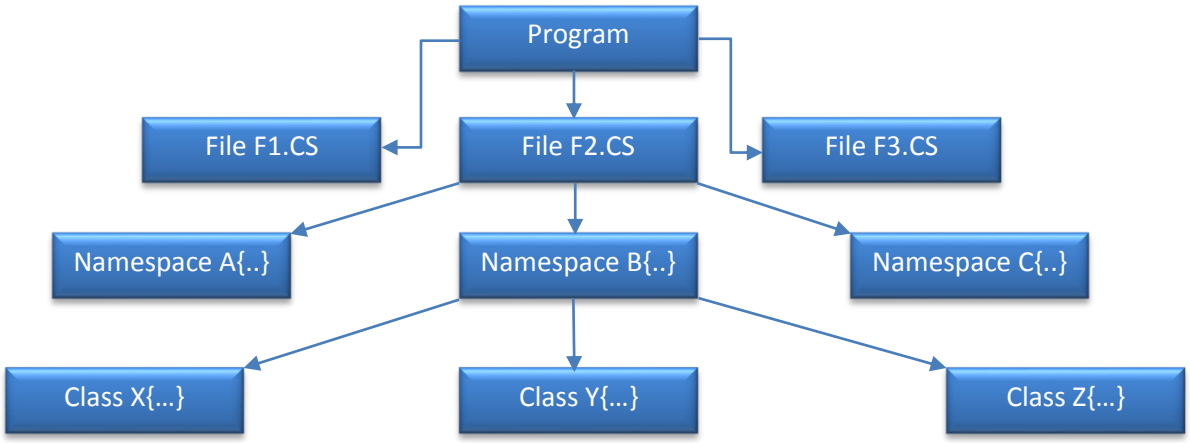
Chapter 2: Development Tools

C# has first appeared in 2000, which has been designed and developed by Microsoft ever since. It is hugely inspired by C++, Java, and Object-Pascal etc. And it has been majorly implemented in Visual C # and .NET Framework.

The key features of C# are as follows,

- Unified Object System
- Single Inheritance
- Interfaces
- Structs and Delegates
- Pre-processor directives

And the program structure of C Sharp is:



2.5 SQL (Structured Query Language)

SQL, elongated as Structured Query Language, is used to query the database and to retrieve the data from it. SQL has various looping techniques, logic offshoots, variables and etc. SQL is categorized under fourth generation languages that do not have conventional way of writing a program. The important advantage is, user defined variables can be easily introduced and used. And also the queries once used can be reused and also make them to run automatically using triggers.

Chapter 2: Development Tools

SQL has been the way to create, maintain and manipulate a database. Many statements like select, create, drop, insert etc. can be easily read and understood by a database programme. There are also different types of languages (or statements) present in SQL itself

1. Data Definition Language (DDL) Statements
2. Data Manipulation Language (DML) Statements
3. Data Control Language (DCL) Statements
4. Transaction Control Language (TCL) Statements

The advantages of SQL are:

1. Powerful database access language
2. Non procedural language
3. Automatically generated by SQL compiler
4. Portability across systems
5. Multiple views of data

2.6 ADO .NET

ADO.NET provides consistent access to data sources such as SQL Server and XML, and to data sources exposed through OLE DB and ODBC [12]. It can be considered as a bridge connecting data, data sources and applications using them. It is included in Microsoft .NET Framework, which can be used to manipulate data stored in database systems, as MySQL in this case.

ADO .NET has several data providers, which allow communication with different data sources, depending on the database being used. Regardless, of which data provider used, similar set of objects are used to interact with a data source. There are set of objects included in ADO. NET which help in interacting with data source and they are as follows,

As mentioned, ADO .NET interacts with different data providers, but the main functionality is to manage the data and get the desired outcome. Different data providers that ADO .NET interact with are as following,

- SQL Data Provider
- ODBC Data Provider
- OleDB Data Provider
- Oracle Data Provider
- Borland Data Provider

In the following figure (Fig.14) you can find the ADO .NET object model and the relation between Data Providers and Data Consumers.

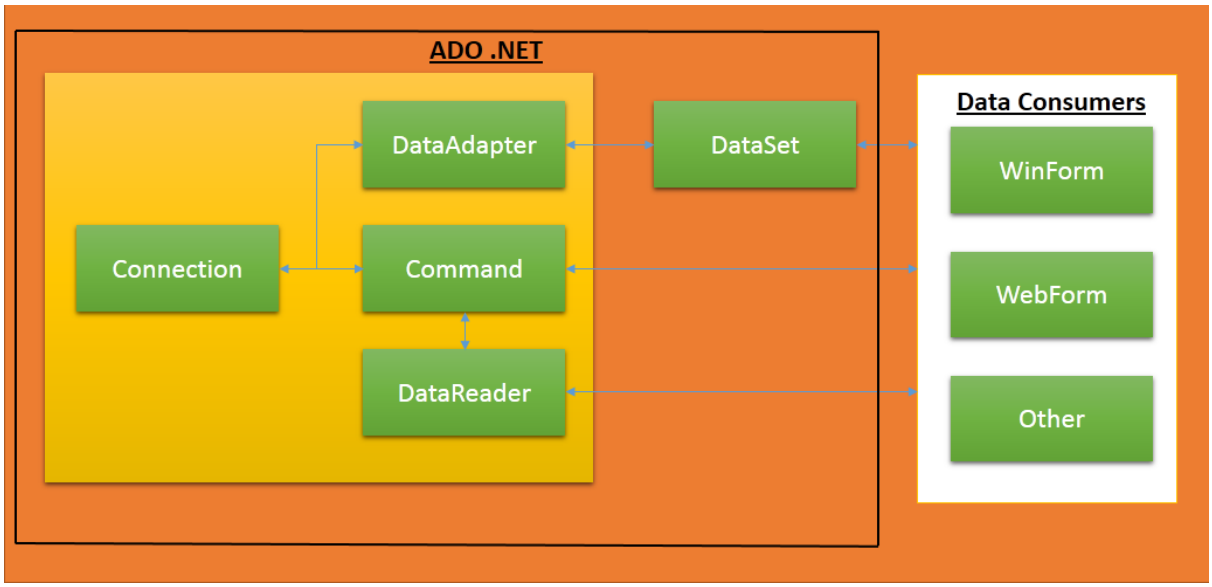


Fig.15: ADO .NET Object Model [13].

2.7 MySQL Connector

MySQL Connector/Net, provided by MySQL is a driver used to connect ADO .NET and data sources. With help of connector, a programmer can easily develop high performance and secured .NET applications which require data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET-aware tools. Programmers and developers can build applications using available .NET languages. Connector/Net is a fully managed ADO.NET driver written in C#.

The version used in this thesis project is MySQL Connector/Net 6.4. It includes support for MySQL Server 5.5, 5.1, and 5.0. Important new features include support for Windows

authentication (when connecting to MySQL Server 5.5+), table caching on the client side, simple connection fail-over support, and improved SQL generation from the Entity Framework provider [14].

2.8 MySQL Namespaces

MySql.Data.MySqlClient namespace of MySQL has been used in this master thesis project to use the in-built classes [18].

MySqlData.MySqlClient
- MySqlCommand
- MySqlConnection
- MySqlDataReader
- MySqlDataAdapter

In order to use this namespace this has to be mentioned in the program in the following pattern: using MySql.Data.MySqlClient;

MySqlCommand:

This class has been used to use the public instance methods available. These are capable of executing the queries depending upon the type they are and the expected return value

ExecuteNonQuery(): This method is used to manipulate the data inside the database.

ExecuteScalar(): This method returns a single value result query.

ExecuteReader(): This method is used to read the command text and execute it.

MySqlConnection:

Though MySQL connector is used to connect to the database, this class has to be used to provide the connection details in a connection string. This class has two constructors. The second constructor takes arguments, generally the connection string.

In order to make this connection string work important information has to be defined as server name, user name, password and etc., These are all put together to form a connection string but divided by a semi colon to differentiate each one.

MySqlDataReader:

This is the class used for accessing the data of a database. It runs any valid SQL select statement on the database and returns the results of the query to the client application. It cannot, therefore, be used to insert, update or delete records held in database tables.

MySqlDataAdapter:

This MySQLDataAdapter class is a relation between a DataSet and MySQL, which explains how data is retrieved and saved. MySQLDataAdapter provides this bridge by mapping Fill, which changes the data in the DataSet to match the data in the data source, and Update, which changes the data in the data source to match the data in the DataSet, using the appropriate SQL statements against the data source [17].

Chapter 3: Database & Software Development

In the modern age robotics, one of the main stream problems is to store the information and retrieve it when necessary. This is more problematic when the matter comes to a robot like H20, The Humanoid, which are capable of moving from one place to another, carrying things and detecting objects around. Number of things had to be considered for their effective usage. And one of the possible ways to make that happen would be using database management. A well maintained database, with querying could produce all the required data over a period of time or other parameter, which helps in deciding things about humanoids.

And when it comes to laboratory robotics, the integration of object-oriented databases into robot control programs, like H20 Control Centre in the case of H20, and strategies to optimize multi-step procedures could be a fruitful idea. Such database systems can be used by the laboratory scientists in using the robots for tasks. For example, in case of multiple robots, the information about a robot being used for transportation of objects like test tubes, vials etc. can be stored into a database. And this database can be viewed in later time and be used for decision making like which robot can be used for certain task, which has more capability of working and which robot has a safer path to achieve the task.

But only a simple object-oriented database can't be of a greater help. These program/ component/ application should also be capable of providing some more useful information. Information such as average time for a whole task too can be a useful for making decisions.

This whole thinking lays a perfect platform for this master thesis project, which could be one of the possible solutions for above mentioned situations in laboratory robotics. Developing a database and making it accessible for people in laboratory environments through a graphical user interface. The database developed in this project is not an ordinary database, but can be considered as Statistical Database as statistical analysis is done on the available database to provide user with easy decision making.

It took two important things to make this solution possible which are discussed in detail in this chapter. They are

- Database Development using Structured Query Language/ Database Programming
- Database Application development using Software Programming

Chapter 3: Database & Software Development

To start with database development, H20 database was developed depending on the necessity of regularly storing data of the mobile robots for future reference. Different information about H20 has to be maintained in a database like the following,

- Power
- Activity
- Position
- Power
- Sensors

In development of any database it is very important to have a good architecture and design as they are the important ideas behind maintaining the database. So database structure and design are discussed in general and as well as in this project's point of view.

3.1 Database Architecture

Database architecture can be identified and differentiated by examining the way application logic is spread throughout the whole system. Application logic consists of three main components:

- Presentation Logic
- Processing Logic
- Storage Logic.

The presentation logic component is responsible for formatting and presenting data on the graphical user interface for the client. The processing logic component handles data processing as querying the database. Finally, the storage logic component is responsible for the storage and retrieval from actual devices such as a hard drive or RAM. By determining on which level these components are being processed, the architecture of database can be identified and differentiated.

3.1.1 Two Level Client/Server Architecture

The two level client/server architecture is the most prominent and well known to the users. Present day very common implementation of this database architecture is, for example,

a Microsoft Windows based application or client program that access SQL server or Oracle. User has access to the database server via network through a Structured Query Language (SQL) based, objectively designed Graphical User Interface (GUI).

As every other thing, Two Level Client/Server Architecture also has its own advantages and disadvantages. It is very advantageous when there are limited number of users and the database is local. And also less remote data access where limited security is required. Given these conditions, Two Level Client/Server Architecture has capability of good understanding and maintenance.

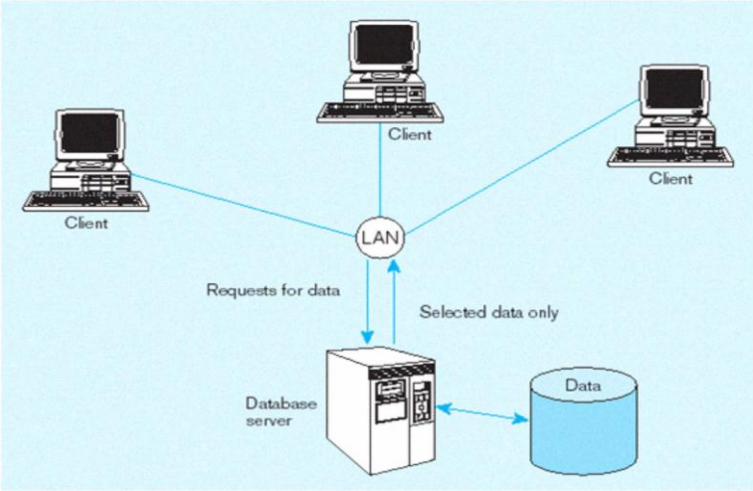


Fig.16: Example for Two Level Client/Server Database Architecture [15].

3.2 Database Design

Though Design and Architecture are widely confused with each other, in terms of database they differ entirely. Architecture is about the strategy, stages and purpose. But whereas design is the implementation of those strategies and plans. Database architecture and design are equally dependent on each other, leading to the final product.

Database design can be considered as procedure that helps building a new detailed database. It contains the various design concepts required for designing the database. It has the different attributes, keys and parameters used in a database. It also helps to generate the ER (Entity-Relationship) diagram. Though it is commonly assumed as the designer for the

database structure, it also can be used to store the different queries that are used to query the database and manage the system.

The design process has few steps included in successful designing of a database, and they are as following,

- Identify the main motivation of maintaining the database
- Organizing the data according to necessity
- Tabulating the data
- Column the data items
- Naming the primary keys
- Establish table relationships
- Refining the data and design
- Applying the normalization rules

And ER (Entity-Relationship) diagram is a graphical illustration that gives the exact relationship between the entities in the database and the database itself. Variety of symbols are used in ER diagrams and they are as follows,

- Boxes – Represents Entities
- Ovals – Represents Attributes
- Diamonds – Represents Relationships

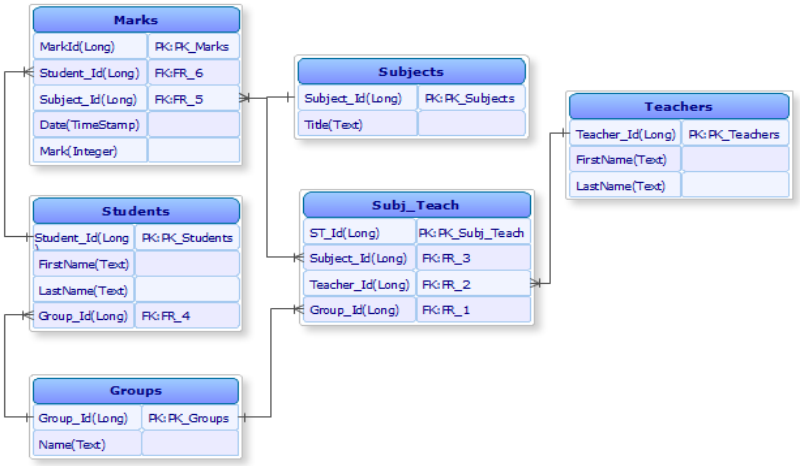


Fig.17 Example for ER (Entity-Relationship) Diagram [16].

3.3 Architecture and Design of Project Database

3.3.1 H20 Database Architecture

In this project the database architecture is Two Level Client/Server Architecture. The user has access to mobile robot, H20 data, through a front end graphical user interface which interacts with a local database server. Though the architecture appears to be three level as three machines are involved in the whole process, but the intermediate, real time happening processes are, only two.

The main idea and reason behind choosing a Two Level Client/Server Architecture is the given environment of work. The database is local and has fewer entities with a limited number of users and no high traffic, thus resulting in good maintenance.

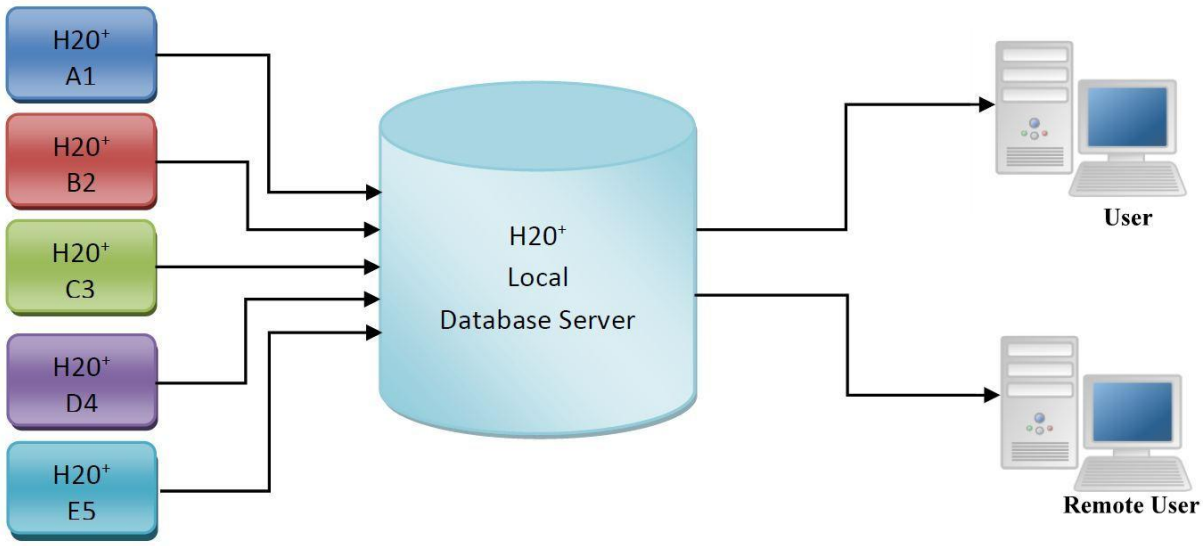


Fig.18: H20 Two Level Client/Server Architecture.

3.3.2 H20 Database Design

Database design in general is way different when it is compared with H20 Database design as it is a database for a laboratory robot. According to the actual requirement, the key necessity of H20 database is to be able to decide which H20 is ideal for which task. Considering this as main idea, the H20 database has been designed accordingly.

The activities and data usage situations are investigated and analysed before deciding the quantity/attribute, type and scope. The analysis is also done in the user's point of view. The

incoming data from different H20, which are different attributes from different entities, had to be organized and tabulated accordingly, which is done. As the data updates over regular intervals of time, the data is columned according to their properties and or attributes. And to maintain uniqueness of the table or record, Primary key has been defined. Also the relationship between different entities and their attributes are established. As a final step, the data is constantly refined which leads to better database.

Following are the important properties of the project,

- Database: MobileRobotH20
- Entities: Robot A1, Robot B2, Robot C3, Robot D4, Robot E5
- Attribute: Working Area, Position, Power Data, Sensor Data, Robot Activity

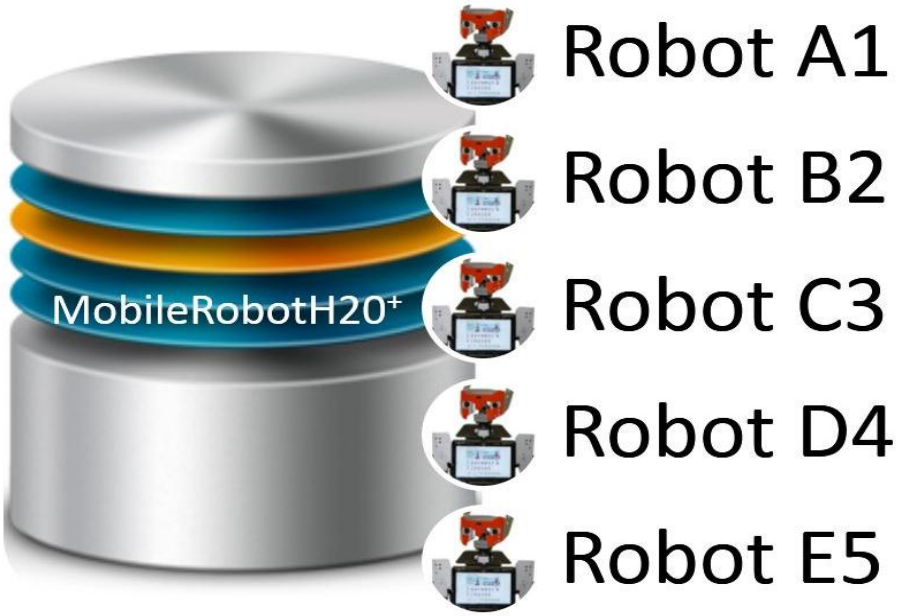


Fig.19: MobileRobotH20 with Multiple Entities.

As mentioned in the properties, the MobileRobotH20 has five different entities, which are five different H20. Each H20 has five different attributes and four of them again have few sub attributes each. Different attributes and their sub-attributes are mentioned below,

Position – RobotPosX – X Co-ordinate Information from GPS Sensor.
RobotPosY – Y Co-ordinate Information from GPS Sensor.
RobotPosDir – Facing Direction Information from GPS Sensor.

Robot Activity – RobotStatus – Information about H20 Working Status.
RobotP2PStatus – Information about H20 Point to Point Task.
RobotShortestPath – Information about the path for Point to Point Task.

Power Data – BatteryOneVol – Battery One Voltage Readings.
BatteryTwoVol – Battery Two Voltage Readings.
DCVol – DC Input Voltage Readings.
BatteryOneStatus – Battery One Usage Status.
BatteryTwoStatus – Battery Two Usage Status.
DCStatus – DC Input Usage Status.

Sensor Data – FrontSonarSensor – Information from Sonar Sensor on H20 Base.
MiddleSonarSensor – Information from Sonar Sensor on front body of H20.
BackSonarSensor – Information from Sonar Sensor on rear body of H20.
RightHumanSensor – Information from Right Human Sensor on H20.
LeftHumanSensor – Information from Left Human Sensor on H20.

The ER (Entity-Relationship) Diagram (considering only one H20 - Robot A1), representing the relationship between different entities and their attributes can be seen below,

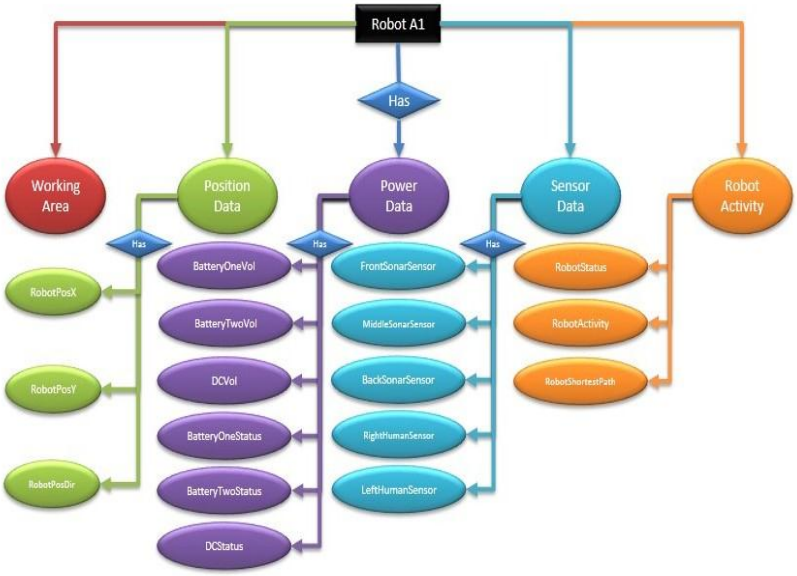


Fig. 20: ER (Entity-Relationship) Diagram for MobileRobotH20 Database.

3.3.3 Attributes and Their Data Types

As well-known widely, Data Types are used for identifying the attributes and to decide what values can be held by them. All together there are totally twenty attributes which are used in this project. Attribute and their data type are mentioned below,

No.	Attribute	Data Type	Length
1	SerialNumber*	INTEGER	50
2	LTime	TIMESTAMP	-
3	WorkingArea	INTEGER	50
4	RobotStatus	TINYTEXT	-
5	RobotPosX	DECIMAL	10,5
6	RobotPosY	DECIMAL	10,5
7	RobotPosDir	DECIMAL	10,5
8	BatteryOneVol	DECIMAL	10,5
9	BatteryTwoVol	DECIMAL	10,5
10	DCVol	DECIMAL	10,5
11	BatteryOneStatus	TINYTEXT	-
12	BatteryTwoStatus	TINYTEXT	-
13	DCStatus	TINYTEXT	-
14	FrontSonarSensor	DECIMAL	10,5
15	MiddleSonarSensor	DECIMAL	10,5
16	BackSonarSensor	DECIMAL	10,5
17	RightHumanSensor	DECIMAL	10,5
18	LeftHumanSensor	DECIMAL	10,5
19	RobotP2PStatus	TINYTEXT	-
20	RobotShortestPath	TINYTEXT	-

* - Primary Key

3.4 Database Programming

Apart from database designing, database programming is also an important part of database maintenance. The main concern during the programming should be the performance of database. And also keep complex queries far from trivial to translate from the requirements. In this section database programming techniques implemented in this thesis project will be discussed.

SQL, structured query language is used to program/or query a database. Created by IBM, is used to define and manipulate objects and data in a database. It is the mainstay of most of modern day database systems. SQL can be divided into two main classes [16],

- Data Manipulation Language (DML) – SQL for retrieving and storing data
- Data Design Language (DDL) – SQL for creating, altering and dropping tables.

Both of these classes had been used in this thesis project. Database and the tables used in this project have been created by programming using SQL command line client. Though there are variety of MySQL graphical user interfaces present to create a database, the conventional way of doing it, through MySQL command client has been chosen to eradicate any mistakes. As the available front end graphical user interfaces are third party applications.

After building a successful database, as discussed, it takes yet again good SQL programming ability along with efficient Software programming to provide user with an intriguing and well-performing database application. All the important functions used in this Master Thesis project, which are written in SQL, are discussed here.

All the SQL functions used in this project are categorized according to their complexity and necessity. Of which some are directly triggered by user input and some are intermediately triggered.

- Functions based on Data Manipulations
- Functions based on Search Conditions
- Functions based on Statistical Analysis
- Functions based on Graphical Representations

3.4.1 Functions Based on Data Manipulations

For the ease of use, few very basic functions were written using SQL. For example, to delete any unwanted record in a table, exporting the table data to other format, to load the data into a table view and etc. Two such functions written in SQL for this project are to delete any unwanted record from a data table and exporting the data to Microsoft Excel.

```
DELETE From robota1 WHERE SerialNumber = @SerialNumber;
```

Delete:

This query helps in deleting a record from the table, irrespective of its order in the table. And also the whole table is updated after the deletion is done.

Export to Excel:

```
SELECT * From robota1 INTO OUTFILE 'd:/Data.csv' FILEDS TERMINATED BY ',';
```

This query exports and make a copy of the select table data from the database to Microsoft Excel in the CSV format. "FIELDS TERMINATED BY ',"" helps in separating the data into respective columns in Microsoft Excel.

3.4.2 Functions Based on Search Conditions

In this section, queries for search based on different criteria are discussed. The database can be loaded into the GUI based on the values input for the search criteria, provided by user. The available criteria for search are on Battery Voltage, Robots information in certain time period and on the work area of robot.

Battery Voltage:

```
SELECT * From robota1 WHERE BatteryOneVol>='" + textBox1.Text + "' and BatteryOneVol<='" +
textBox2.Text + "' ORDER by BatteryOneVol in ASEC;
```

In this query, user has the flexibility to look at the data according to the voltage ranges which are input by one. And the whole data which is updated in the datagridview of user interface is ordered by Battery Voltage and in ascending order.

Time and Date:

```
SELECT * From robota1 WHERE LTime>='" +  dateTimePicker1.Value.ToString ("yyyy-MM-dd") + "' and
LTime<='" + dateTimePicker2.Value.ToString ("yyyy-MM-dd") + "' ORDER by LTime in ASEC;
```

Here, the whole data is reloaded into the datagridview of user interface, which belongs to a certain time period specified by the user. And the whole reloaded data is ordered by ascending time.

Working Area:

```
SELECT * from robota1 WHERE WorkingArea='" + comboBox1.Text + "'
```

Sometimes user would like to know about the robots that are involved in a work in a specific room. This function gives user the ability of searching the data depending on a specific room.

3.4.3 Functions Based on Statistical Analysis

Though functions based on data manipulation and search conditions are useful for the user, the functions that are based on Statistical Analysis are the main work of this Master Thesis project. These functions turn the normal data base into a statistical database used for statistical analysis. And they are created on the idea of providing user a database application with edge of statistical analysis. These functions are capable enough to statistically calculate the data and returned the processed information in to the text boxes of a user interface.

Number of Tasks:

```
SET @last_task = 0;

SELECT SUM(new_task) AS tasks_performed

FROM (SELECT IF(@last_task = RobotShortestPath, 0, 1) AS new_task,

@last_task := RobotShortestPath FROM robot" + comboBox1.Text + ") AS tmp”
```

This query can be considered as a compound SQL statement because of intermediately passed sub queries.

It returns with number of tasks done by a selected robot. When this query is passed, the data table of specific robot is checked. Whenever a new change appears in the column RobotShortestPath attribute, it is counted as one and zero if not. The sum of these entire one’s

gives the total number of tasks done by a specific robot. To assure no mistake every two consecutive rows of the RobotShortestPath are compared with each other.

Average Battery Charge/Discharge Time:

```
Create Table B1ChgTime (a int(225), b text(100));
Insert into B1ChgTime (a,b)
(Select IF(@last_task = BatteryOneStatus, 0, 1) AS StatusChanged,
@last_task := BatteryOneStatus FROM robot" + comboBox1.Text + " ORDER BY LTime);
Select (count(b) / sum(a))*10 as number from B1ChgTime where b = 'CHARGING';
Drop table B1ChgTime;
```

This query returns the average battery charging time used by a robot. This also is a compound statement. The main advantage of this function in user interface is, the user can determine which robot can work for a long time with uninterrupted battery backup and assign a task respectively.

Time for Respective Task:

```
SET @last_task = 0;
Create Table Temp" + comboBox1.Text+ "(a Text(50), b Text(50));
INSERT into Temp" +comboBox1.Text + "(a,b)
(SELECT IF(@last_task = RobotShortestPath, 0, 1) AS new_task,
@last_task := RobotShortestPath FROM robot"+comboBox1.Text+" ORDER BY LTime);
 SELECT DISTINCT b AS Task, ((count(b)*10)/sum(a)) AS AvgTime FROM Temp"+comboBox1.Text+"
GROUP BY b ORDER BY 1 DESC;
DROP TABLE Temp"+comboBox1.Text+"";
```

This query returns the different tasks done by each robot and the average time utilised for each task.

Average Battery Voltage:

```
SELECT AVG(BatteryOneVol) FROM robot" + comboBox1.Text + ";
```

This query returns the average battery voltage of both the batteries mounted on H20.

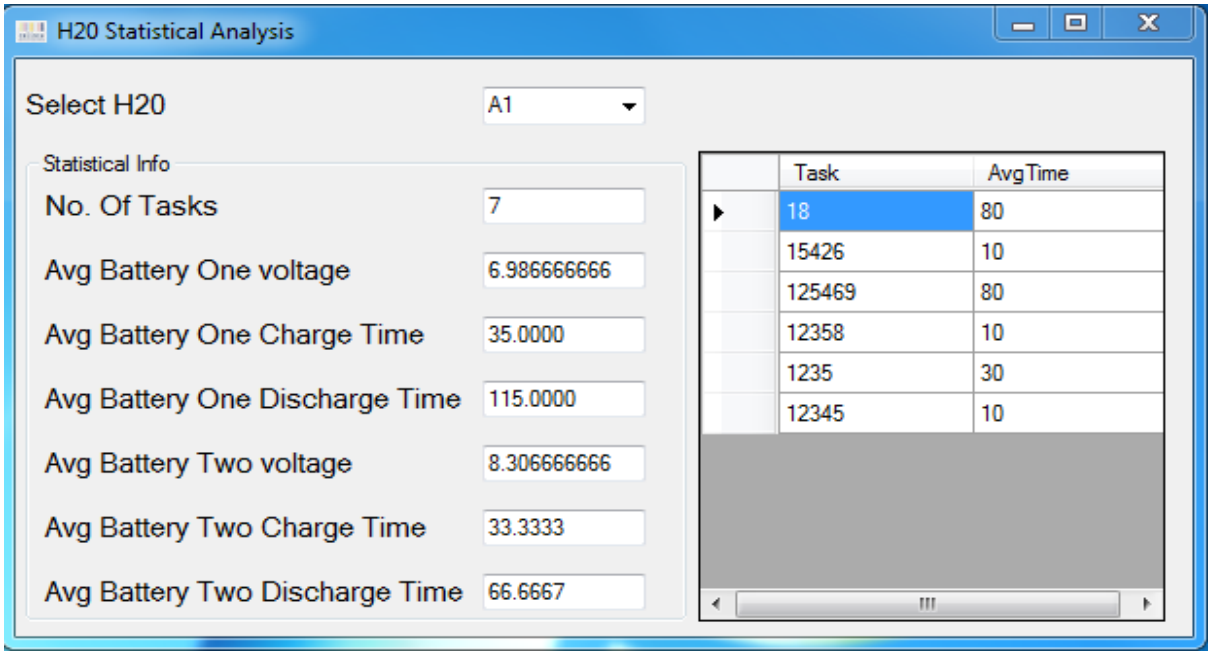


Fig. 21: H20 Statistical Analysis

3.4.4 Functions Based on Graphical Representation

In this section we discuss how the SQL is used to fill the data inside Microsoft Charts. The query language is used for creating temporary data tables whenever the program is started. So that these data tables can be used to store data required for charts. And the chart is used to plot a graph between Battery Voltages and Time.

As every temporary table has four columns, the battery voltage information and time is inserted into them whenever a query depending on time period is passed to the database. The data binding service uses these temporary tables and the column data inside the table as source of information to plot graph. Each column is assigned to each axis on a chart. Whenever user wishes to see the graphical representation of battery behaviour over time, these charts are can be used. When once the program is closed, all the previously created tables are dropped off the database.

The real advantage of these charts is the very clear statistical representation of data. This idea of temporary database leads to effective use of Microsoft Charts and consumes less space on server.

3.5 Software Programming

As one of the main aim of this thesis project is to provide user with a nice and intriguing user interface for database, software programming is carried out. Software programming is necessary to connect the front end user interface with the database. Software programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions.

In this section of the chapter, various techniques of software programming for connecting front end graphical user interface and database are discussed. Different methods are introduced in this section. Several software programming techniques are used in this project like ADO. NET programming and logical programming which includes SQL programming inside.

The software programming was broken into a step by step procedure according to the work flow. So the complete procedure of connecting user interface with the database of H20 is divided into tasks.

1. Saving the data from H20 into database.
2. Establishing the connection between database and data binders.
3. Data binding.

3.5.1 Save Function

A save function was required to be written in order to save data into the MobileRobotH20 database. This function is included into another class which is capable of communicating with H20 and acquire the data. This function accepts an incoming string with robot data and robot name as shown below,

```
public void Save(string incomingrobotdata, string robotname)
        {
                .
                .
        }
```

The incoming string of data about robot is split into parts using a string splitter and assigned to respective database attributes. And the string about robot name is used for identification and redirecting the data to respective entity. Following is the format of the incoming data string.

```
connect2DataBase.Save("201 ON 24.1 35.3 46.4 57.5 68.6 79.7 USING USING CHARGING 213.2 314.3 415.4 516.5 617.6 ACTIVE 12-8-7-6" , "F8");
```

And following is the string splitter,

```
char[] splitter = new char[] { ' ' };
string[] RobotInfo = incomingrobotdata.Split(splitter, StringSplitOptions.None);
for (int i = 0; i < RobotInfo.Length; i++)
        {
          // Assign string reference based on induction variable.
          string value = RobotInfo[i];
         }
```

The whole incoming string is read and split whenever a space is observed in the string. Each data which has been split is assigned to an array which will be used in future for saving the data into the database. The main idea behind using an array is, array allows using loop and other techniques which are very essential when dealing with large amount of data. After the data string is split, the condition with robot name is checked. And it is satisfied, the entire data is inserted into their respective data table in the database. And if the condition isn't satisfied, a message is sent to check the data string and format.

On a whole, the capabilities of save function are,

- Ensure connection with hardware.
- Checking the data string.
- Splitting the data string.
- Saving into database.

3.5.2 Establishing Connection

The main aim of a database dependent application should be successfully connecting to the database and be capable of retrieving data from it. ADO.NET, provided by .NET framework can be used as a bridge between the database dependent application and the data source, allowing user to manipulate the data and to read and present the data from database by using a DataReader or a DataAdapter.

To connect to MySQL, SqlConnection object of the .NET Framework Data Provider for MySQL is used. In order to use the connection object, all the necessary information for the connecting to database is provided in a connection string. The Connection object you use depends on the type of data source.

The connection initialization information that has to be passed as a parameter from a data provider to a data source is stored or written in the format of connection string. Once the given connection string is valid, the data source applies the options and opens the connection.

The format of a connection string is a semicolon-delimited list of parameter values. And the parameters have to be declared before the connection string is declared. Data Provider for MySQL has elements from older style of syntax and is generally more flexible with common connection string syntax. Mostly there are equally valid connection string syntax elements, but some syntax and spelling errors can cause problems. So care should be taken while declaring or passing the connection string with valid connection information.

When the connection string is valid, the data source opens a bridge between database and application. This enables user with the capability of data manipulation. But it is always important the connection is closed after it is used. Close method of connection object can be used to disconnect.

If a connection is left unclosed for a while, pool size is filled and if there is no usable connection available, there happens a stagnation of request until connections are back in to pool. A new function for opening and closing the connection were written for effective use of the connection,

```
// to open the connection
this.OpenConnection();
.
.
.
//to close the connection
this.CloseConnection();
```

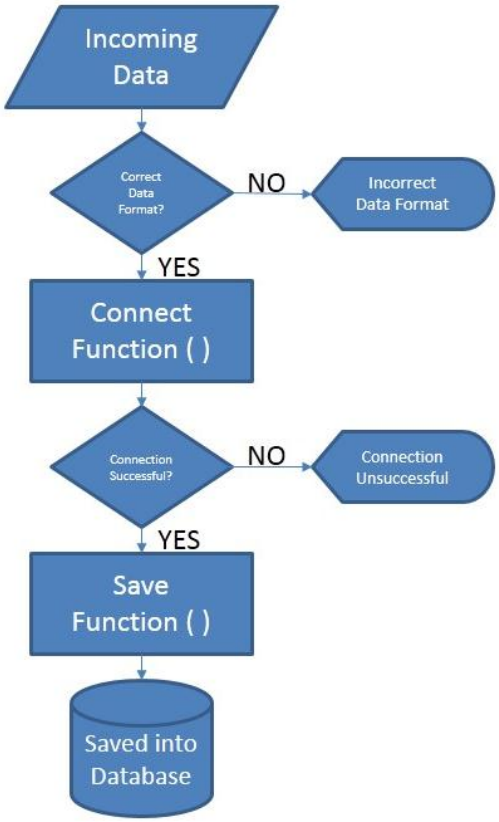


Fig. 22: Saving into Database

3.5.3 Data Binding

Data binding helps application developers to establish a link between the controls on a form and the data in the database. This link is capable of reading and writing. In general, data binding is used in database dependent applications to read or write the data to database. Data binding in windows forms allows developer to access data from databases as well as in other structures, such as arrays and collections.

Data binding can be useful to manually bind the data base and application if there is a complex input scenario. With help of data bound input controls, when an exception is thrown, it can be understood that there has been an error in input data. Errors like this can be handled

with various validation / pre-commit events on the data controls but it can be easier to manually validate and input correct data.

Data binding is of two types which are available and are also used in this project. They are Simple and Complex Data Binding.

Simple Data Binding: Is the ability to bind to a single value in a dataset or single value from a column of a database. These values are shown in single text boxes or on labels etc.

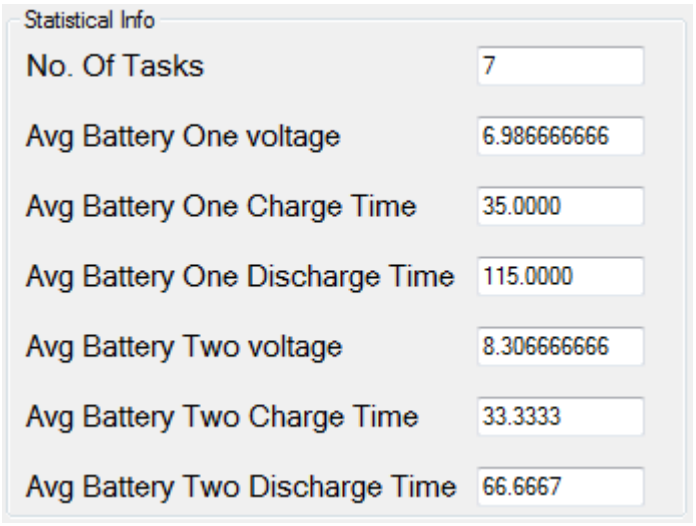


Fig. 23: Example for Simple Data Binding.

As shown above in Fig. 21, is an example for Simple data binding where the information is shown in text boxes. Such data binding, though regarded as simple, might need a complex query to get the result.

Complex Data Binding: Where as in Complex data binding, it is the ability to bind to a set of data. For example a whole record or a whole table from a database. These values are generally shown in data grid view, combo box or etc.

Data bindings as these sometimes take very easy query to return whole data in a database table. For example, SELECT * FROM TABLE NAME can return such result in data grid view.

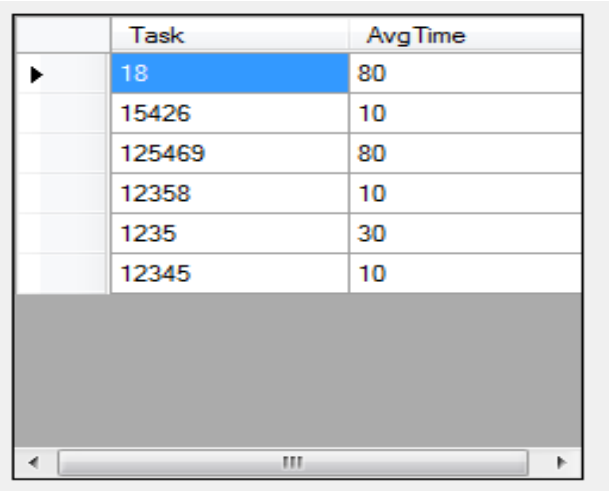


Fig. 24: Example for Complex Data Binding.

Data binding in Visual C# can be done using variety of data structures provided by ADO .NET. These objects are capable of holding data from database and present them. The available data structures are,

- DataColumn
- DataTable
- DataView
- DataSet
- DataViewManager

In the section following next, the data binding objects used in this project and the data adapter class which help the data binding object are discussed. The figure (Fig. 23) given below shows the hierarchy in the data binding objects and the data adapter class.

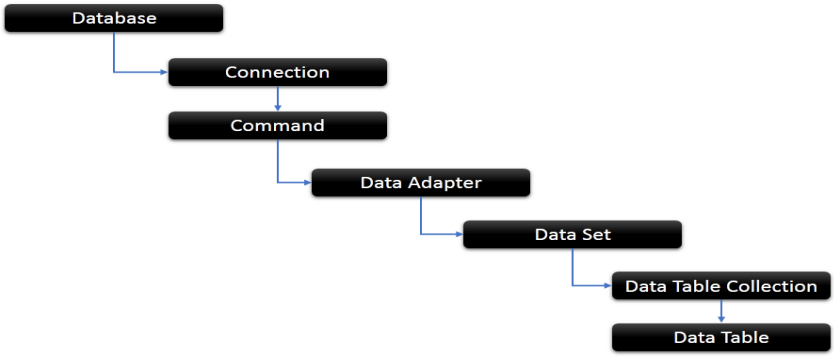


Fig. 25: Data Binding Hierarchy.

3.5.3.1 DataAdapter

DataAdapter is capable of accessing data with the ADO.NET model. It acts as the bridge between the database and the objects in the ADO.NET object model. The DataAdapter object's Fill method provides the capability to fetch the results of a query into a DataSet or a DataTable so that the data is easily accessed. The DataAdapter object is also used to submit any changes to database that have occurred in the DataSet.

DataAdapter object is a part of Command object and has variety of properties and methods. The SelectCommand, UpdateCommand, InsertCommand and DeleteCommand are the objects present in the DataAdapter class that represent the SQL query that is used to populate, update, add and delete a record into the DataSet object. With each of these properties stored procedures or SQL statement are called so that they perform the required operation. Once any data manipulation is done or completed, the UpdateCommand can be called in the DataAdapter object, and ADO .NET will use the command object created earlier to update the database.

In this project as well, DataAdapter object was used to hold the data and to present in dataset and also the DataAdapter object properties were used to delete a row from database and update the table.

```
{
        .
        .
        .
        //Prepare adapter to run query
        AdapterA1 = new MySqlDataAdapter(query1, connection);
        .
        .
        .
        //Get query results in dataset
        AdapterA1.Fill(DS);
        .
        .
        .
        //Delete Command
        string query2 = "DELETE FROM robota1 WHERE SerialNumber=@SerialNumber";
        AdapterA1.DeleteCommand = new MySqlCommand(query2, connection);
        AdapterA1.DeleteCommand.Parameters.Add("@SerialNumber", MySqlDbType.Decimal, 10,
"SerialNumber");
        AdapterA1.DeleteCommand.UpdatedRowSource = UpdateRowSource.None;
        .
        .
        .
}
```

3.5.3.2 DataSet

The DataSet object can be considered as a collection of data. It contains numerous DataTable objects that are stored in the DataSet's object table's collection. The major advantage of DataSet is, even a whole database can be presented with it.

The data stored in the DataSet object is disconnected from the database. Any changes that are made are simply stored in a DataRow. The whole object can be passed to the database when changes are wished. The DataSet provides a GetChanges method that can be used to

```
DataTable GetA1DTInfo()
{

//Prepare query to get all records from items table
string query1 = "Truncate Table a1p";

.
.

//Prepare adapter to run query
MySqlCommand cmd1 = new MySqlCommand(query1, connection);
.
.
cmd1.ExecuteNonQuery();
.
.
AdapterA1 = new MySqlDataAdapter(query3, connection);

DataSet DS = new DataSet();

//Get query results in dataset
AdapterA1.Fill(DS);

// return with info
return DS.Tables[0];

}
```

retrieve only the modified rows from a DataSet.

The DataTable in a DataSet can be filled with the incoming data from data source to which we are connected through a DataAdapter. The DataSet object can serve as an offline database, which has data stored in it.

Chapter 3: Database & Software Development

In our project data binding has an important role to play as it has to build a bridge between database and the front end graphical user interface with help of DataSet and DataTable. All the above shown codes are implemented in this project for successful data binding.

In the following chapter the underlying philosophies, techniques and the other procedures for designing a graphical user interface are discussed and presented.

Chapter 4: GUI Designing & Testing

"A picture is worth a thousand words. An interface is worth a thousand pictures."

- Ben Shneiderman

Human–computer interaction (HCI) is the study of how humans interact with computer systems. Many techniques are employed for this sort of interaction, including computer science, psychology, engineering, graphic design and etc. HCI covers all aspects of the way in which user interact with computers. In everyday life, users come into contact with an increasing number of computer-based technologies.

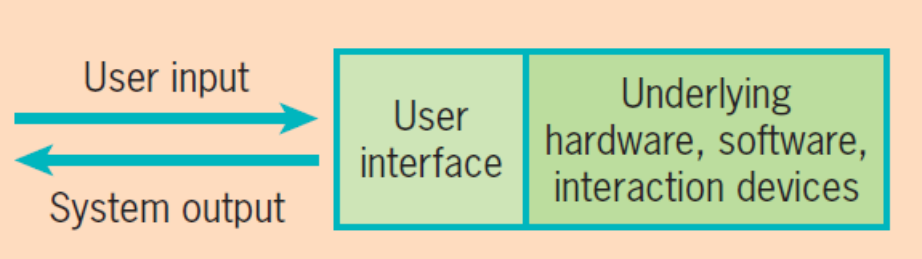


Fig. 26: Work of a User Interface.

User interacts with the computer through a user interface. As a developer, it is a responsibility to design good user interface — interface that is easy to use and easy to understand, that meet the needs of the intended users, as looking into data, manipulating data and ability to statistical analysis, in the case of this thesis project.

This chapter introduces the details about the techniques used to develop the user interface. The key idea behind the design was to make a user friendly user interface, where the user has total control. For designing an effective user interface, the decision on techniques were taken considering users point of view of the user interface. Questions were prepared so that they could be answered by the interface design. Summing up all these, the designing of interface was broken into two philosophies,

- What user would like to see in a database application?
- What user wants to do with what he is seeing in a database application?

These philosophies acted as motivating factors for the user interface design in this master thesis project.

Chapter 4: GUI Designing & Testing

In order to design a good user interface good tools are required. And also object oriented programming. Object oriented programming is capable of unifying a program around its data, i.e., objects and a set of definite interfaces to that data. An object-oriented program can be considered as data controlling access to code. There are many advantages of object-oriented programming, they are,

- Minimalism
- Modularity
- Ability to modify
- Extensibility
- Salvage ability
- Maintainability

In this master thesis project Visual C Sharp was used for object-oriented programming available in Microsoft Visual Studio 2008. Visual C Sharp has edge over other object-oriented programming languages like Visual Basic. VC# has capability of implementing interface in derived class which is defined in base class. This capability of VC# has been widely used in this project. The reasons behind choosing Microsoft Visual Sharp for this project over the others are,

- Automatic garbage collection
- Pointers no longer needed
- Reflection capabilities
- Definition of classes and functions can be done in any order
- Declaration of functions and classes not needed
- Un-existing circular dependencies
- Classes can be defined within classes
- Great support from Microsoft/MSDN

In this thesis chapter, to present user the design of graphical user interface visual references as screenshots of the software have been used. As only programming languages are difficult to understand and not everybody's cup of tea, visual referencing has been used. This idea is realized from the object-oriented programming itself. If .NET framework is not available in Microsoft Visual Studio, it would take an application developer ages to develop any application. Because of the pre written codes for many of the objects present in .NET

framework used for application programming, like buttons, forms, radio buttons, combo boxes and etc. any developer is able to develop applications and meet deadlines in time. So inspired from this idea, the Graphical User Interface of this database application has been introduced and explained in the following sections.

4.1 GUI Designing

As mentioned earlier that the spine of designing were two philosophies from users point of view, the designing process was carried in a way as replying them. The designing procedure was categorized and carried out depending upon those philosophies.

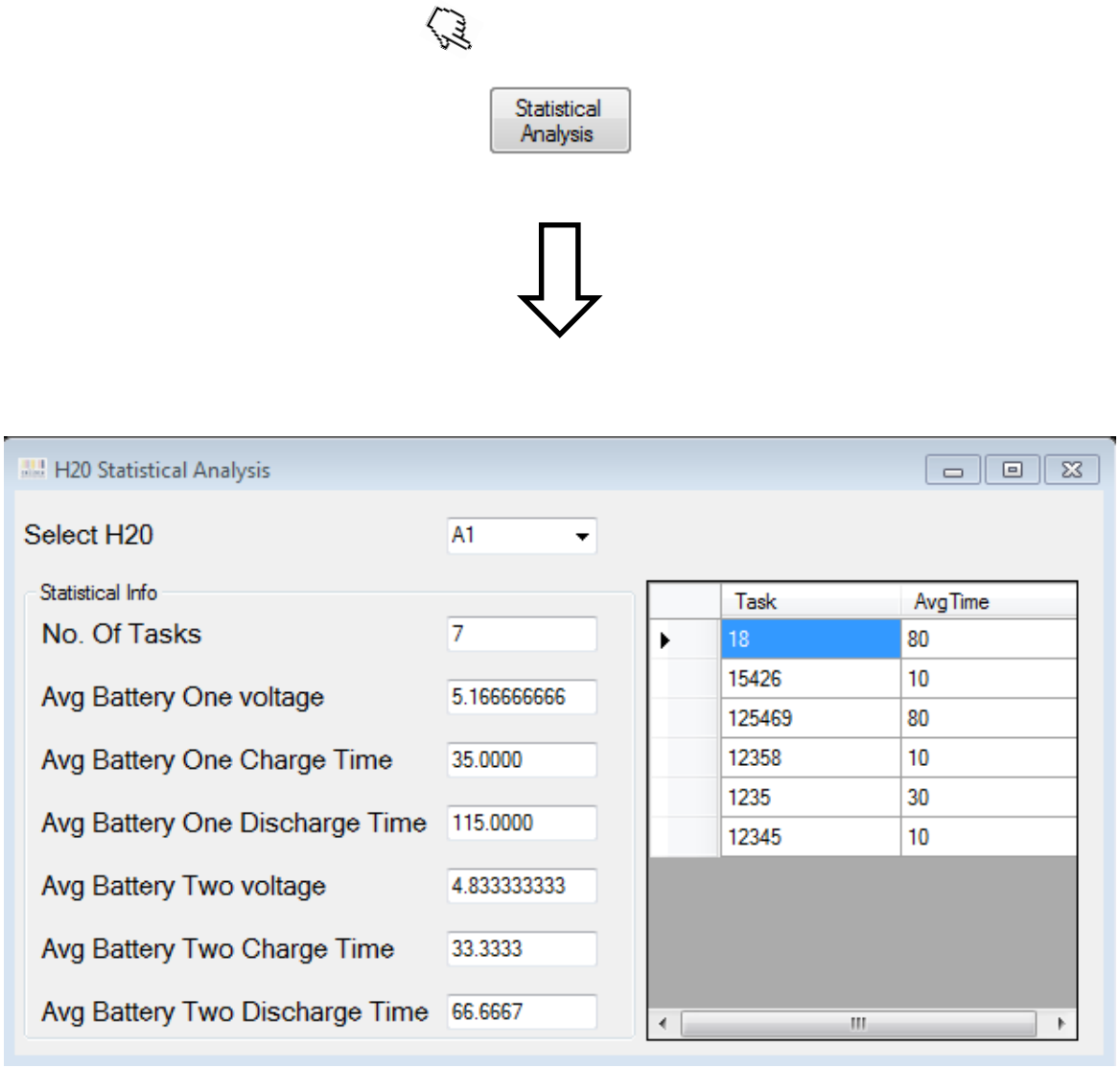
Any user using any application would be interested in knowing what that specific application has to offer. Designing an interesting main user interface of any application is very important. Not many times, but it has effect on users thinking about application. Always a good main interface leaves a good impression on the user. So as a reply to the first philosophy or question of what user would like to see in a database application, H20 Database Viewer Main interface was designed.

Fig. 27: H20 Database Viewer GUI.

Chapter 4: GUI Designing & Testing

H20 Main User Interface offers user to choose between viewing the database of individual robot and/or to see the statistical analysis. User interface with Statistical Analysis is of major importance in this project. This interface with all the information for a user can be of very importance. Especially in the laboratory environment, this helps user to make a faster decision on which robot to use for which task.

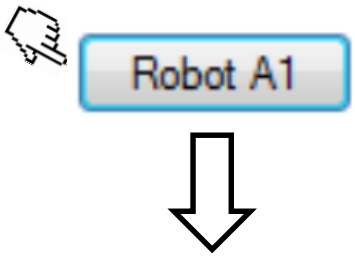
Let us walk through it and see its capabilities. The user can select the robot of his choice from the available list. Once selected, the data is returned into the text boxes and the data grid view as well. The figure (Fig. 25) below is the graphical user interface for Statistical Analysis done on H20 that provides user with statistical information



The information shown in these textbox's are the average battery voltage calculations and number of tasks done by the specific robot on a whole.

The information shown in this datagridview is the different tasks done by the specific robot and the average time used for each in seconds.

As an answer to first philosophy, the other available option is to view the database of individual robot. Here the idea behind providing user with a button each for each robot is ease of access. Sometimes not only people from laboratory environment but also ordinary people with no much knowledge of computer options like list box or combo box may want to use this application. This roots for individual button for individual robot.



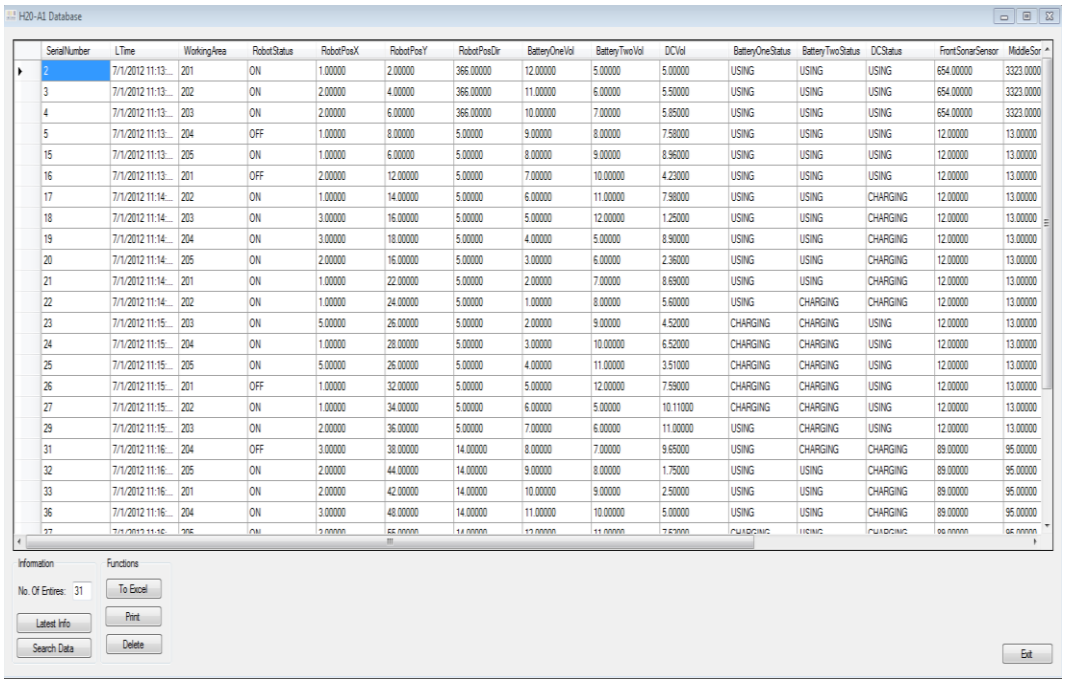
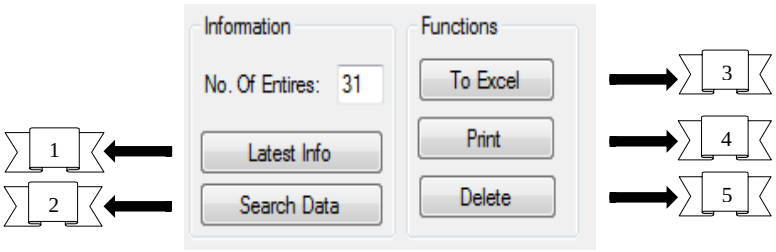


Fig. 28: H20 Database UI.

In this above user interface, user can access to see all the data stored in the database about the specific robot selected. And the other functions available on this interface were developed considering the second philosophy, what user wants to do with what he is seeing in a database application? Some special functions were decided to be included which gives user some additional choices like to export, print or delete the data. And also user can look into the data by sorting with search criteria.



1. To see the latest information about the robot (Fig. 27).
2. To look into data with search criteria (Fig. 28).
3. To export the data to excel (Fig. 36).
4. To print out the data (Fig. 37).
5. To delete a record from the database.

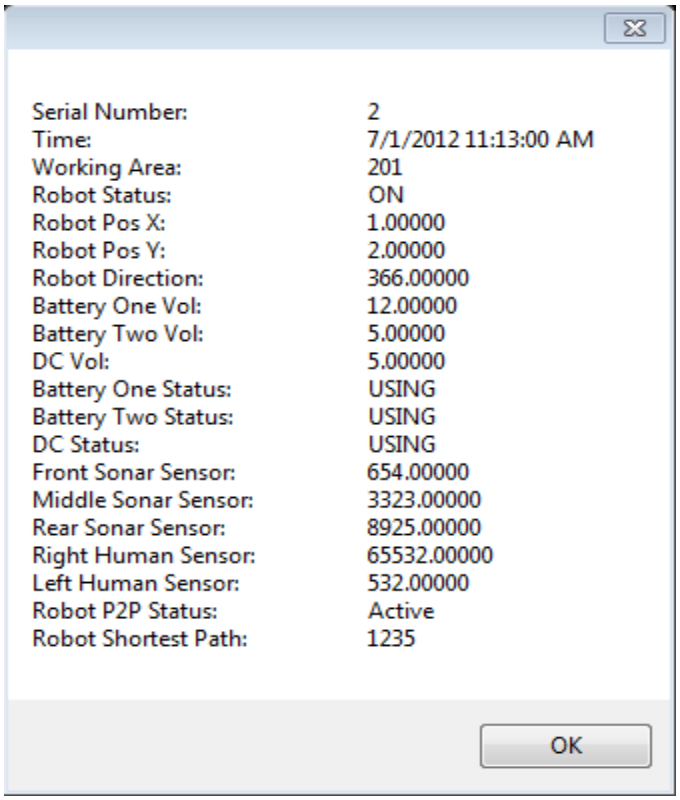


Fig. 29: User Interface with latest information about H20.

The necessity of this function arouse from providing ease to user in learning the latest information about H20 with just a single click. Even there is a scroll button present the UI (Fig. 28) to go through all the data; it is not good enough when user would need some data just to know about whereabouts of the H20.

The idea of providing user with possible search and sort over a criteria arouse from database itself. As known database is a large collection of data which is updated at a regular intervals of time. For a user it is quite impossible to go through all the data from top to bottom to get to see what he is looking for. And this idea also comes from our everyday usage of search engines on internet. For example, search engines like Google, Bing and etc. sort the data whenever we input the search criteria and return with possible match returns, thus providing user an edge. So search conditions were included H20 so that user can sort the data according to his necessity. Though available search conditions are limited for now, they can be improved in the future work. Included search conditions in this thesis project are,

- On Voltage
- On Time
- On Working Area

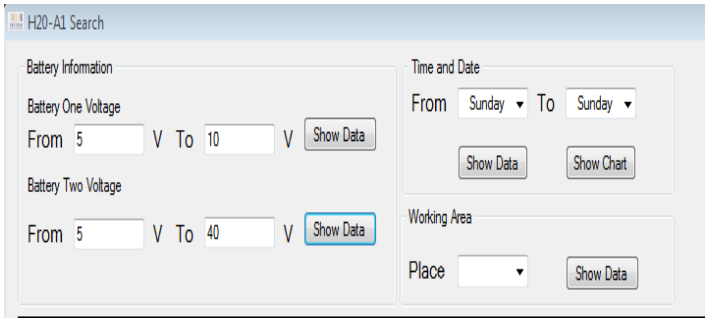


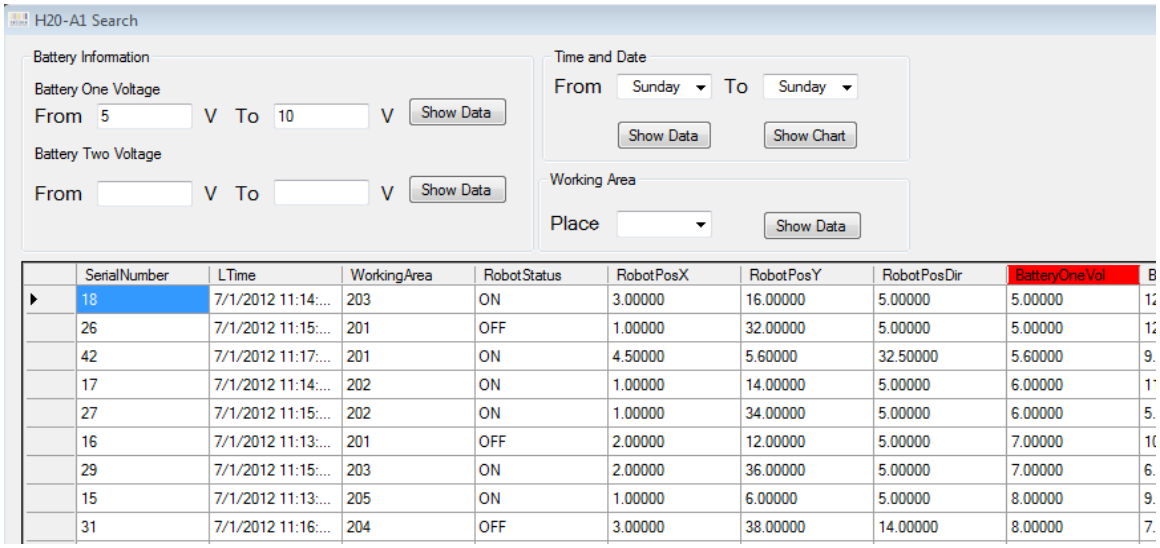
Fig. 30: User Interface with available search conditions.

The major idea of any search engine is just the same, give the user what he is looking for. So the same has been implemented in this thesis project too. This is also a part of designing a good user interface, where user has to feel that he not only can see the data but also sort it as he likes and to observe the behaviour at certain point of time, voltage or in working area. The possible search conditions could be on the status of the robot, the path robot has been following and much more new search conditions if much more attributes to an entity are added.

On Voltage: In the following figures (Fig. 31 & Fig. 32) you can see how the user can sort the data according to voltage search criteria depending on voltages of different batteries mounted on H20. On any mobile robot working on an alternative power source, the power conditions of the power source should be of concern. This search condition help user in deciding few things about H20 and its working,

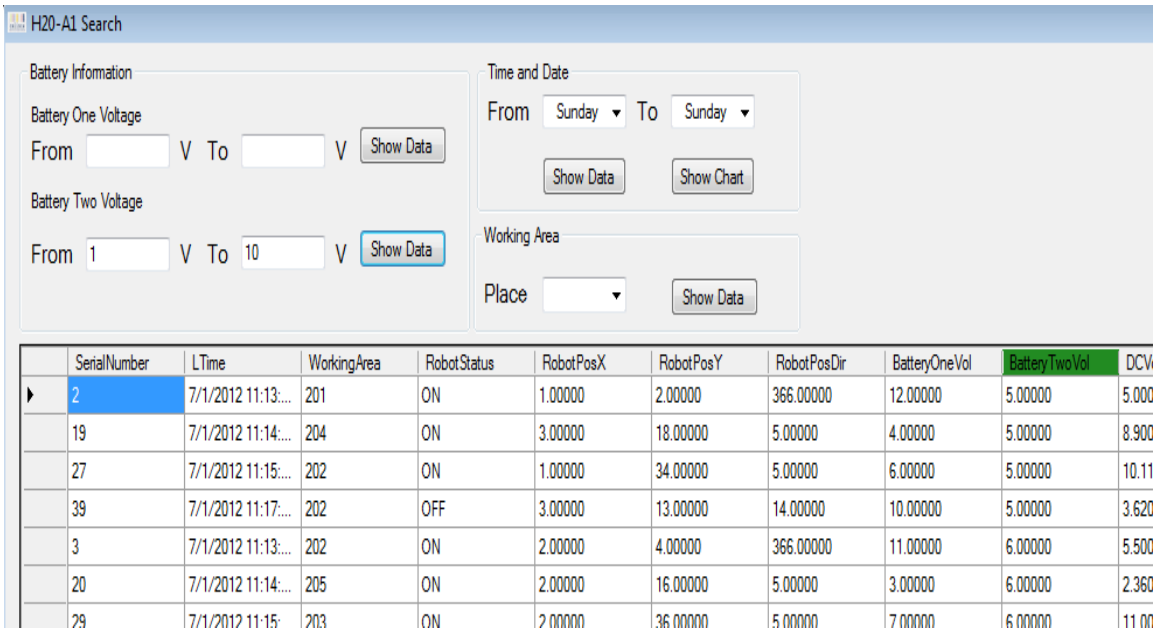
- How long is battery running?
- How fast is a robot operating in given voltage ranges?
- Which voltage range is good enough for working?
- What are the voltages when the H20 should be put to charging or not?

This information is trivial because in the beginning when the batteries, which act as alternative power source, are new, the working of H20 could be efficient. But over a period of time when the batteries performance start depleting, this voltage search conditions could be of greater help. When certain battery voltages are input the whole data is sorted in between the voltages in ascending order. With the help of ascending sort method, user can see if his initial input of start voltage is good enough for assigning work to H20 or not and also how long is battery supporting H20's work till the end voltage.



SerialNumber	LTime	WorkingArea	RobotStatus	RobotPosX	RobotPosY	RobotPosDir	BatteryOneVol
18	7/1/2012 11:14:...	203	ON	3.00000	16.00000	5.00000	5.00000
26	7/1/2012 11:15:...	201	OFF	1.00000	32.00000	5.00000	5.00000
42	7/1/2012 11:17:...	201	ON	4.50000	5.60000	32.50000	5.60000
17	7/1/2012 11:14:...	202	ON	1.00000	14.00000	5.00000	6.00000
27	7/1/2012 11:15:...	202	ON	1.00000	34.00000	5.00000	6.00000
16	7/1/2012 11:13:...	201	OFF	2.00000	12.00000	5.00000	7.00000
29	7/1/2012 11:15:...	203	ON	2.00000	36.00000	5.00000	7.00000
15	7/1/2012 11:13:...	205	ON	1.00000	6.00000	5.00000	8.00000
31	7/1/2012 11:16:...	204	OFF	3.00000	38.00000	14.00000	8.00000

Fig. 31: User Interface with Battery One Voltage search ranges.



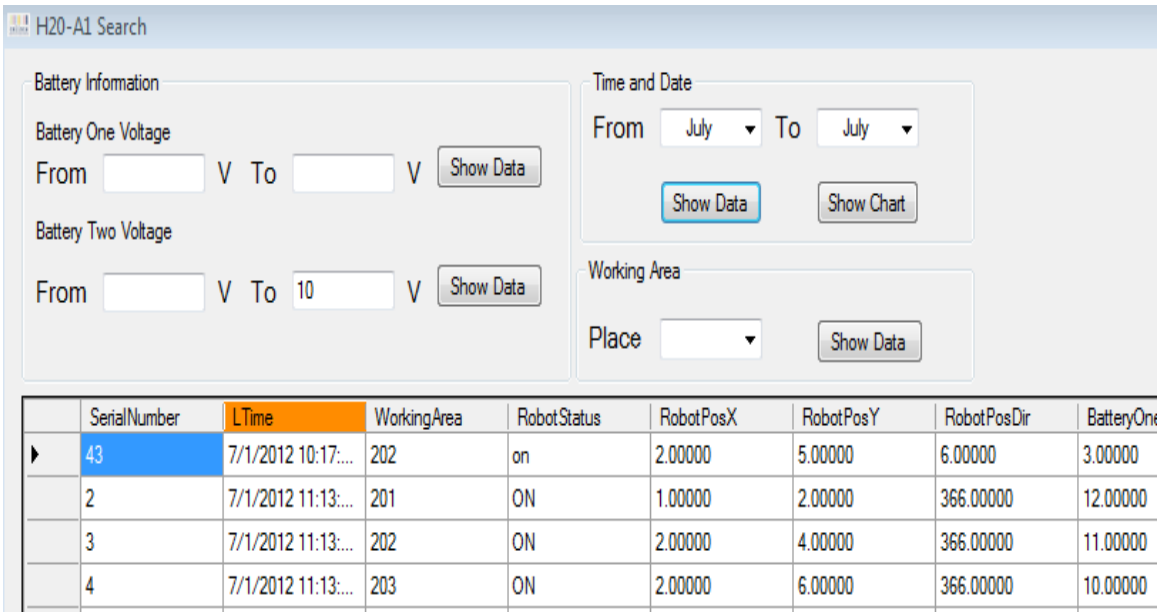
SerialNumber	LTime	WorkingArea	RobotStatus	RobotPosX	RobotPosY	RobotPosDir	BatteryOneVol	BatteryTwoVol	DCV
2	7/1/2012 11:13:...	201	ON	1.00000	2.00000	366.00000	12.00000	5.00000	5.000
19	7/1/2012 11:14:...	204	ON	3.00000	18.00000	5.00000	4.00000	5.00000	8.900
27	7/1/2012 11:15:...	202	ON	1.00000	34.00000	5.00000	6.00000	5.00000	10.11
39	7/1/2012 11:17:...	202	OFF	3.00000	13.00000	14.00000	10.00000	5.00000	3.620
3	7/1/2012 11:13:...	202	ON	2.00000	4.00000	366.00000	11.00000	6.00000	5.500
20	7/1/2012 11:14:...	205	ON	2.00000	16.00000	5.00000	3.00000	6.00000	2.360
29	7/1/2012 11:15:	203	ON	2.00000	36.00000	5.00000	7.00000	6.00000	11.00

Fig. 32: User Interface with Battery Two Voltage search ranges.

On Time: In the following figure (Fig. 33), the data is sorted on time condition. This idea of time criteria for search ails from the same idea of providing user with better information. Searching the database on time criteria has a lot of added advantages when compared with search on voltage criteria. With voltage search criteria only the performance of battery can be understood but with time search criteria the user can know more information. If user inputs a certain period i.e. in between two different days, he can discern the information about H20 like,

- Information about battery condition
- Information about tasks carried out
- Information about the working area
- Information about robot status
- Information about point to point task

And in special conditions, if working in the night as a surveillance robot, basing on human sensor readings, the presence of any human can be assumed. Soon more properties as feed from camera would be of help in this condition.



SerialNumber	LTime	WorkingArea	RobotStatus	RobotPosX	RobotPosY	RobotPosDir	BatteryOne
43	7/1/2012 10:17:...	202	on	2.00000	5.00000	6.00000	3.00000
2	7/1/2012 11:13:...	201	ON	1.00000	2.00000	366.00000	12.00000
3	7/1/2012 11:13:...	202	ON	2.00000	4.00000	366.00000	11.00000
4	7/1/2012 11:13:...	203	ON	2.00000	6.00000	366.00000	10.00000

Fig. 33: User Interface with time related search conditions.

If observed a special function not available for any other search criteria, only for time, a graphical representation in Microsoft Charts is present (Fig. 34). This helps user knowing the battery behaviour over a period of time. Though user can learn from the user interface with grid view (Fig. 33), it is always easier for user to asses using a graph representation. From the graph user can know,

- Life of Battery
- Usage of Battery

And as information from both the batteries are included in the same graph, this also can be used for comparing those and to learn which battery has been working better. And which one's life is depleting soon. With this information user can decide on the power source.

- Can it be used any further?
- Should it be sent to maintenance?
- Should it be replaced with a new one?

If the problem persists even after replacing with a new one, from the graphical representation user can consider of other parameters that could be affecting both in the hardware and software. And this chart or graph also comes with print options.

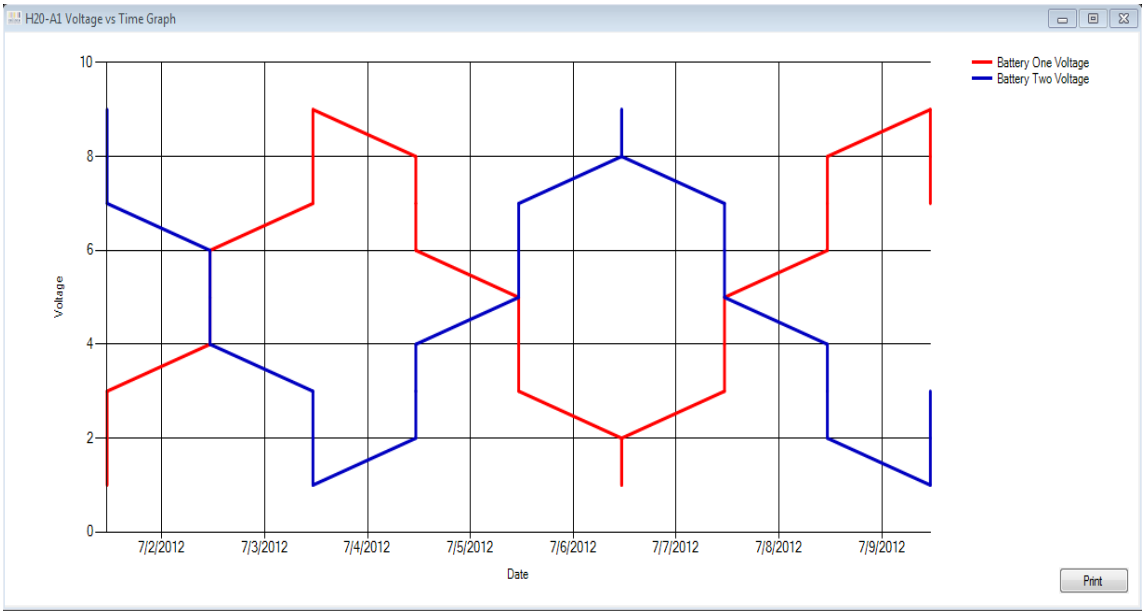


Fig. 34: Graph, Time vs. Voltage.

On Working Area: Search criteria depending on a respective working are is also important because it gives the user with information about a certain area where H20 has worked. With this information user can study H20. Information like,

- Information about tasks done in that room.
- Information about any obstacles depending on point to point activity status.
- Information about how long H20 has been working.

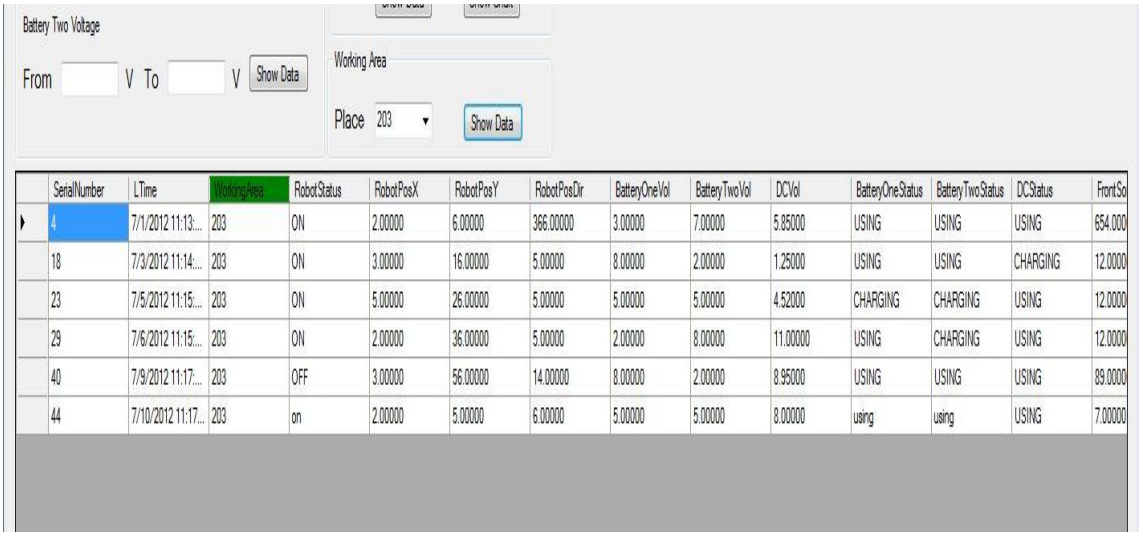


Fig. 35: User Interface with Working Area related search conditions.

Chapter 4: GUI Designing & Testing

These functions included in main user interface are from an idea of helping user with storing the data. These help user in saving data both digitally in computer using Microsoft Excel and also in archives using the print. These both can be used for future reference. And also an added advantage with exporting the data to excel is, it also can be used as a resource for a database if necessary.

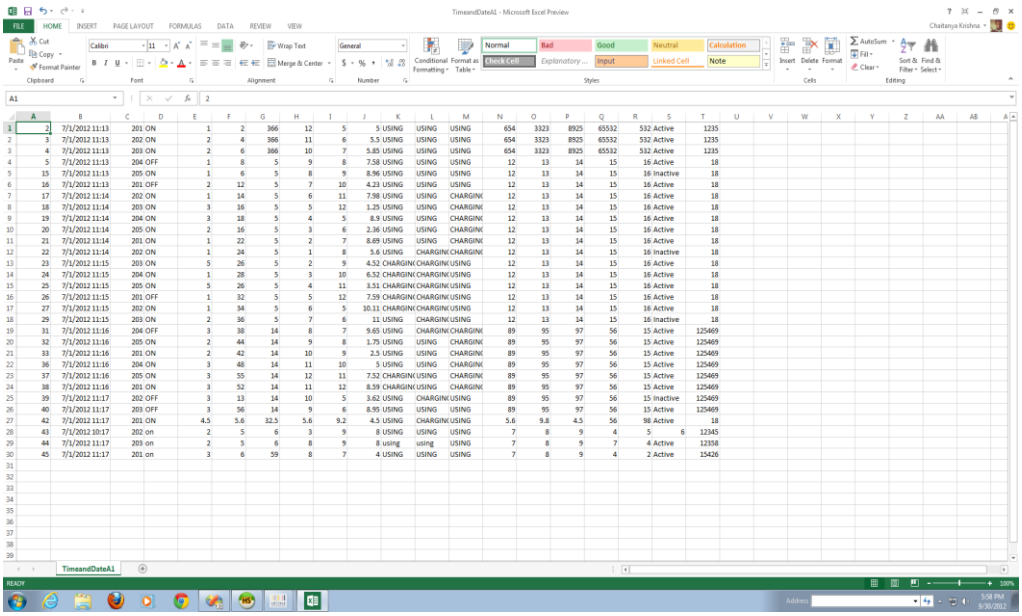
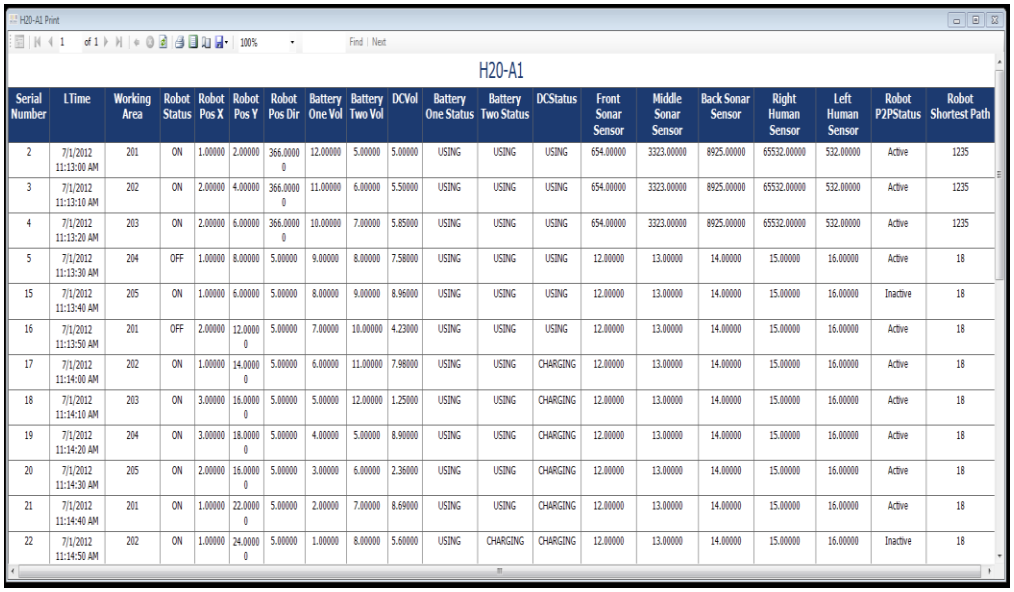


Fig. 36: Data exported to Microsoft Excel.



H20-A1

Serial Number	LTime	Working Area	Robot Status	Robot Pos X	Robot Pos Y	Robot Pos Dir	Battery One Vol	Battery Two Vol	DCVol	Battery One Status	Battery Two Status	DCStatus	Front Sonar Sensor	Middle Sonar Sensor	Back Sonar Sensor	Right Human Sensor	Left Human Sensor	Robot P2PStatus	Robot Shortest Path
2	7/1/2012 11:13:00 AM	201	ON	1.00000	2.00000	366.00000	12.00000	5.00000	5.00000	USING	USING	USING	654.00000	3323.00000	8925.00000	65532.00000	532.00000	Active	1235
3	7/1/2012 11:13:10 AM	202	ON	2.00000	4.00000	366.00000	11.00000	6.00000	5.50000	USING	USING	USING	654.00000	3323.00000	8925.00000	65532.00000	532.00000	Active	1235
4	7/1/2012 11:13:20 AM	203	ON	2.00000	6.00000	366.00000	10.00000	7.00000	5.85000	USING	USING	USING	654.00000	3323.00000	8925.00000	65532.00000	532.00000	Active	1235
5	7/1/2012 11:13:30 AM	204	OFF	1.00000	8.00000	5.00000	9.00000	8.00000	7.58000	USING	USING	USING	12.00000	13.00000	14.00000	15.00000	16.00000	Active	18
15	7/1/2012 11:13:40 AM	205	ON	1.00000	6.00000	5.00000	8.00000	9.00000	8.96000	USING	USING	USING	12.00000	13.00000	14.00000	15.00000	16.00000	Inactive	18
16	7/1/2012 11:13:50 AM	201	OFF	2.00000	12.00000	5.00000	7.00000	10.00000	4.23000	USING	USING	USING	12.00000	13.00000	14.00000	15.00000	16.00000	Active	18
17	7/1/2012 11:14:00 AM	202	ON	1.00000	14.00000	5.00000	6.00000	11.00000	7.98000	USING	USING	CHARGING	12.00000	13.00000	14.00000	15.00000	16.00000	Active	18
18	7/1/2012 11:14:10 AM	203	ON	3.00000	16.00000	5.00000	5.00000	12.00000	1.25000	USING	USING	CHARGING	12.00000	13.00000	14.00000	15.00000	16.00000	Active	18
19	7/1/2012 11:14:20 AM	204	ON	3.00000	18.00000	5.00000	4.00000	5.00000	8.90000	USING	USING	CHARGING	12.00000	13.00000	14.00000	15.00000	16.00000	Active	18
20	7/1/2012 11:14:30 AM	205	ON	2.00000	16.00000	5.00000	3.00000	6.00000	2.36000	USING	USING	CHARGING	12.00000	13.00000	14.00000	15.00000	16.00000	Active	18
21	7/1/2012 11:14:40 AM	201	ON	1.00000	22.00000	5.00000	2.00000	7.00000	8.69000	USING	USING	CHARGING	12.00000	13.00000	14.00000	15.00000	16.00000	Active	18
22	7/1/2012 11:14:50 AM	202	ON	1.00000	24.00000	5.00000	1.00000	8.00000	5.60000	USING	CHARGING	CHARGING	12.00000	13.00000	14.00000	15.00000	16.00000	Inactive	18

Fig. 37: Printing of data.

4.2 Testing

In order to validate any application, testing it is required. To determine the quality of an application, testing has to be done under several test cases to guarantee the quality and capability of the application. It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end users.

In simple words, software testing is an activity to check whether the actual results match the expected results and to confirm that it can be reused and is bug free. In testing there are few principles to be followed,

- Finding out the presence of defects
- Impossible comprehensive testing
- Initial Testing
- Collecting the defects or bugs
- Implementation of debugging methods
- Testing is context dependent
- No Errors – No Way

4.2.1 Testing Process

There are totally three types of testing procedures that this database application has undergone. They are Unit Testing, Integration Testing and System acceptance Testing. Each testing procedure will be discussed in detail.

Unit Testing:

Unit testing is something which is done on developers end. The whole code is broken into parts and tested individually. Proper unit testing done during the development stage saves both time and money in the end. In this thesis project as well the whole project was broken down into individual parts and checked. Unit testing was done on database and software programming to ensure that they are working. Test data was sent to database using save function which is included in another class. And the target was achieved as the data has been saved in database. This implies working ability of database and the programming required to connect application and database.

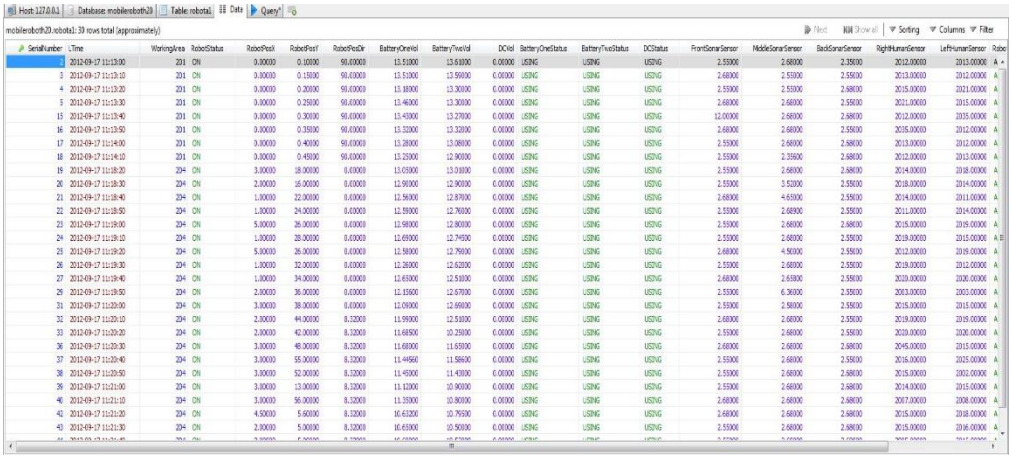


Fig. 38: MobileRobotH20 database with test data.

The above figure ensures the working of two different units of Save Function and Establishing connection. When once data has arrived from hardware, the save function was able to check if the incoming data string is correct or not. And as it was correct, then a connection was established between the save function and database to pass on the data to database. And the data has been saved to the database thus ensuring the functionality of all the units. Thus, Unit Testing was successful.

Integration Testing:

Integration testing is different from unit testing. In this testing procedure the data transfer between modules is verified. In case of this application, integration testing was done to between the front end graphical user interface and the database. This test would result the working of data binding methods. So the user interface was run and the application was capable of retrieving the data and also manipulating it. Thus, integration test was successful.

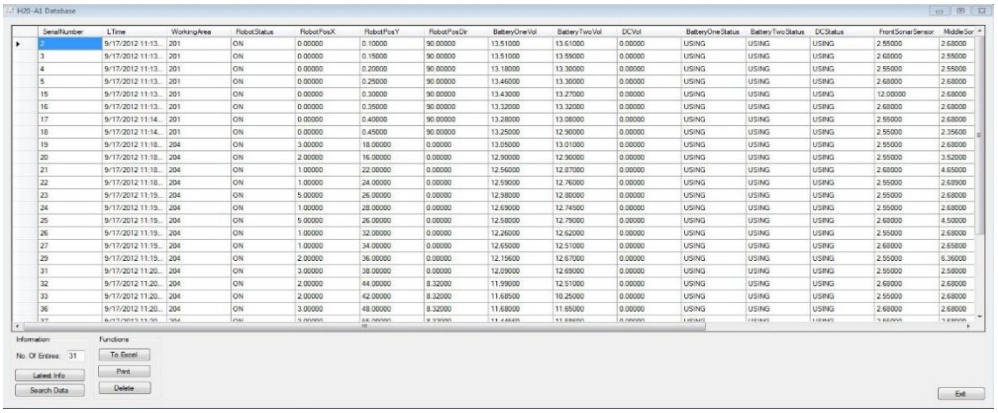


Fig. 39: User Interface showing the test data.

System and Acceptance Testing:

System and acceptance testing is entirely different from the previous tests. This is an end to end testing where the application will be tested by the third person i.e., other than the developer. There are two types of System and Acceptance Testing. Alpha Testing and Beta Testing. Alpha testing was done on this system, which is an end to end test done by the fellow developers or employees. Beta testing is test done on customers end. So the results of alpha testing system were also achieved as expected. Results after system and acceptance testing are displayed below.

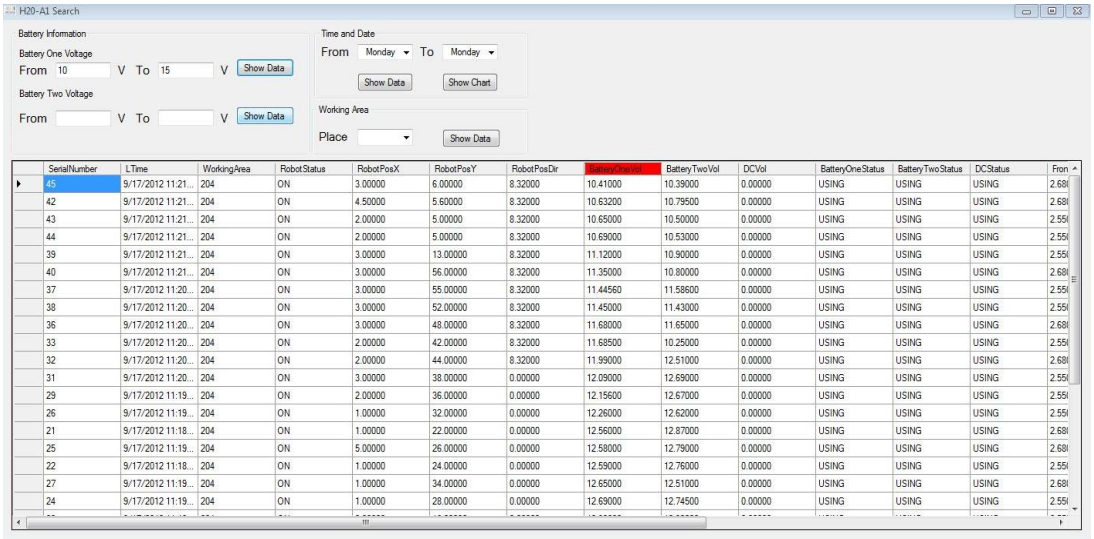


Fig. 40: Test data sorted for given Battery Voltages.

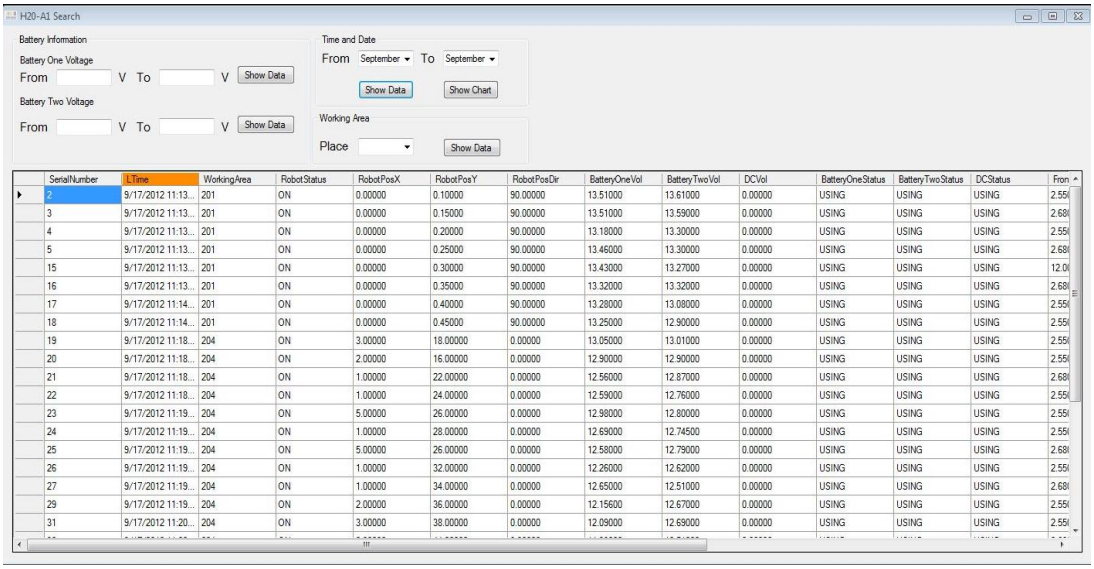


Fig. 41: Test data sorted for given dates.

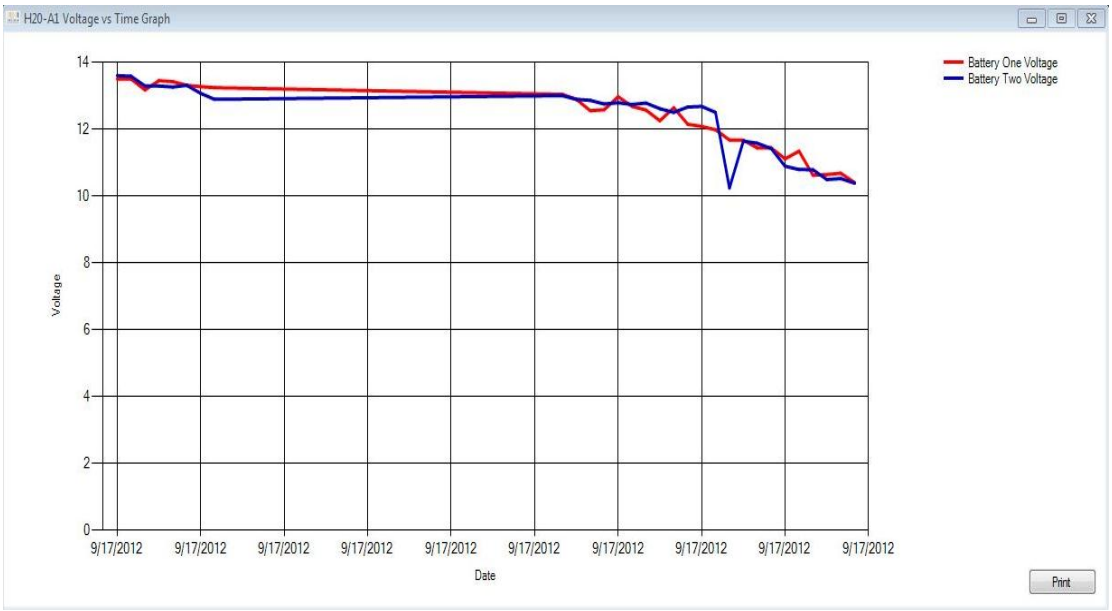


Fig. 42: Graph between Voltage and Time for test data.

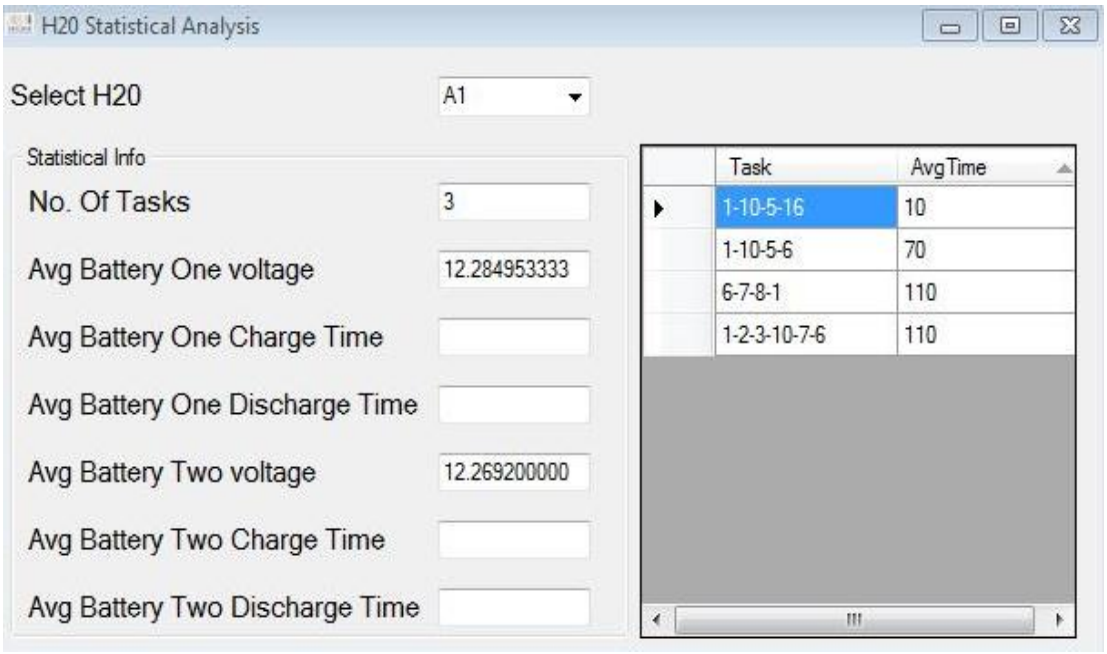


Fig. 43: Statistical Analysis for test data.

Chapter 5: Conclusion & Future Work

5.1 Aim

The main aim or goal of this thesis project was to develop a database, where all the important

- Power Data
- Sensor Data
- Position Data
- Activity Data

Could be effectively stored and used for future reference and database interface functions for storing, deleting, updating and querying the data. And to develop a graphical user interface in windows form where user can interact with the data and benefit from the options available for data manipulation, data sorting, visualizing the power behaviour and if needed print the data and/or to export the data to another source.

5.2 Tasks Achieved

The main tasks for successful completion of this thesis project, which were accomplished are:

- Studying the mobile robot H20 and know about the information that would be stored in database.
- Developing a database with good data design and data modelling.
- Using MySQL to develop functions for
    - Storing
    - Deleting
    - Updating
    - Querying
- Developing database interface functions for statistical calculation.
- Developing a user friendly database based graphical user interface.

Chapter 5: Conclusion & Future Work

5.3 Conclusion & Future Work

After learning about H20 and its working to understand the kind of data that would be stored in the database, a detailed study was carried to know how the database and SQL works. In order to present this database to a user, a database dependent graphical user interface was created using object oriented programming tools like Microsoft Visual C# 2008. And it was tested under laboratory conditions which yielded desired results with statistical calculations for the data stored in the database and with visualization using the Microsoft Charts.

And as a part of future work, this work can be extended to other type of mobile robots. And also with the capabilities of storing huge information, not only about sensors, power and activity but also the feed from the cameras, maintenance of the robots and etc.

Bibliography

[1] Importance of Industrial Automation - http://robotics.youngester.com/2010/05/why-industrial-automation-is-important.html

[2] Figure 1: Example for Laboratory Automation – http://www.rapidmicrobiology.com/news/3184h0.php

[3] About Robotics - Bruno Siciliano, Oussama Khatib, Editors. Page 1, Springer handbook of Robotics

[4] Katrine Verdun, Richard Rouse and Gary Hardiman - DNA microarray fabrication and processing: Automation in the laboratory. BIOGEM, Division of Biology, University of California San Diego.

[5] Figure 2: Dispensing and Culturing by Mahoro - http://www.houseofjapan.com/robots/mahoro-lab-android-automates-dangerous-lab-work

[6] About Database: Jeffrey Ullman and Jennifer widom 1997: First course in database systems, Prentice-Hall Inc., Simon & Schuster, Page 1, ISBN 0-13-861337-0.

[7] Julia Hannemann, Irene Krämer, Editor: Dr. med. Ralf Herold - Material Database for Laboratories: http://www.kinderkrebsinfo.de/e2260/e2163/e2253/e3736/index_eng.html

[8] Figure 8: Project Architecture. Hui Liu, Norbert Stoll, Steffen Junginger, Kerstin Thurow – A common wireless control system for Mobile Robots in Life Science Laboratory.

[9] About MySQL: Schumacher, Robin; Lentz, Arjen. "Dispelling the Myths". MySQL AB. Archived from the original on 6 June 2011. Retrieved 17 September 2012.

[10] About Microsoft Visual Studio: http://msdn.microsoft.com/en-us/vstudio/aa718325.aspx

[11] About Ado .NET: http://msdn.microsoft.com/en-us/library/h43ks021

[12] About MySql connector version 6.4: http://dev.mysql.com/doc/refman/5.5/en/connector-net-versions.html

Bibliography

[13] Figure 15: Example for Two Level Client/Server Architecture: Syed Sohail Ahmed, Usman Haider and Aiman Nadeem. Client/Server Model of Security with Bar-coded Identity Cards (http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05565129).

[14] Figure 17: Example of ER Diagram from http://www.mediainovacijos.com/public/var/userfiles/files/esybiu_rysiu_diagrama.png

[15] Figure 16: Example for Two Level Client/Server Architecture. Joseph M. Hellerstein, Michael Stonebraker and James Hamilton: Architecture of a Database System.

[16] Different SQL classes: http://cplus.about.com/od/thebusinessofsoftware/a/whatissql.htm

[17] Information about MySqlDataAdapter class and its instance methods from: http://www.linuxtopia.org/online_books/database_guides/mysql_5.1_database_reference_guide/connector-net-architecture-adapter.html

[18] About MySql Namespaces from: http://dev.mysql.com/doc/refman/5.0/en/connector-net-ref-mysqlclient.html

[19] About .NET Framework from Beginning MS Visual C# 2008 by Karli Watson, Christian Nagel, Jacob Hammer Pedersen, Jon D. Reid, Morgan Skinner and Eric White.

Statement of Declaration

I hereby declare that this thesis is solely written by me and reflects my own work. I confirm that any part of this work has never been submitted before for a degree or a scientific publication. All quotations, ideas, pictures extracted from external sources are cited.

Chaitanya Krishna Reddy Bommareddy
Date: 8th October, 2012.