Chapter 1

INTRODUCTION

1.1 General Introduction

If an individual, say an athlete, is injured in the knee utmost care has to be taken in-order to recover as soon as possible. For this purpose, the amount of stress on the knee due to an athlete's day-to-day activity needs to be monitored at every instant of time. This is shown in Figure 1.1. If the athlete bends his injured knee too much accidently, then chances of damage to the injured knee would increase and the speed of recovery would take more time.

To facilitate proper correction for speed recovery from damage caused at joints, in this case knee, we propose a prototype which uses accelerometer sensors placed on both the side (one above the joint and the other below the joint) of the joint. Whenever the bend around the joints is more, the athlete is alerted and hence can make the necessary adjustments. In this case the speed of recovery around the joints will be much faster. Further an Android application is built to receive the data via Bluetooth from the microcontroller unit to which the accelerometer sensors are interfaced. This data will help the physician to monitor the athlete's knee during rehabilitation.

Figure 1.1 Monitoring Human Knee Joint

In this project the acceleration readings are provided by accelerometers on 3 different axes namely x, y and z. These readings are obtained by monitoring changes in the positions of two accelerometers, placed on either side of the joint of an individual. Any movement that occurs at the knee joint where the accelerometers are placed will cause these sensors to generate an output voltage corresponding to the change in gravity. These voltages are fed as input to the microcontroller. The microcontroller further processes this information and delivers an output which is transmitted via Bluetooth to an Android based application. This application is responsible for receiving this data and to display it graphically so that it is visually better and is faster to analyze than textual information.

1.2 Statement of the Problem

The main idea of this project is to aid the process of speedy recovery for athletes and people who have undergone knee surgery by constantly monitoring the different activities performed by the patient. With this information a physician will be able to monitor the patient and provide further suggestions to improve the rehabilitation process. For this purpose, two 3-axes accelerometer sensors are placed around the knee joint and they are interfaced to Renesas RL78 microcontroller which converts the analog voltage into a digital value and based on this value the different activities such as walking, running, sitting, standing, etc. performed by the patient are identified and transferred to the Android application which displays this data graphically with the help of a pie chart. Also an emergency button is p38rovided to handle unforeseen situations which when pressed send alerts to the intended recipients via SMS and Email along with the location of the patient by using GPS. If the stress on the patient's knee exceeds due to the body posture, then a buzzer is triggered and also a voice alert is played on the Android application. Further an LCD is used to demonstrate the working of this prototype.

1.3 Objectives of the Project

The objectives of this project are as follows:

- Constantly monitor the different activities performed by the patient which applies stress on the knee joint, such as walking, running, sitting, and standing.
- Alert the patient when the stress on his knee is high due to a posture or activity which is prohibited.
- The data read by the sensors must be displayed graphically so that it looks better in visual and can be analyzed easily.
- The previous data that was read when the system was used must also be stored for the physician to analyze.
- The patient can alert the intended recipients within limited time when the patient is in need of desperate help.
- Use Bluetooth for communication between the Android application and the microcontroller.

1.4 Project Deliverables

The deliverables of this project are as follows:

- A wearable prototype for monitoring the different activities performed by the patient that affects the knee joint.
- An emergency button to aid the patient in pain which triggers the following alerts.
- An Email alert will be sent to the intended recipient along with the location of the patient by using GPS.
- A SMS alert is also sent to the number that is registered on the Android application.
- An Android application which is installed on the patient's phone to receive data and to display it as a pie chart which shows the frequencies of the activities performed by the patient.
- When the patient makes a prohibited movement which affects the knee's recovery, the buzzer is triggered and a voice alert is played on the Android application.
- An option to clear the existing data in the android application is provided.

1.5 Current Scope

As of now the system developed is a prototype in which the microcontroller needs constant power supply for it to run. Also the prototype been highly customized for a single patient. This project helps athletes, old people who have knee problems and people who have undergone knee surgery by constantly monitoring the different activities performed by the patient which involves their knee. Also only certain postures such as walking, running, sitting and standing have been recorded to demonstrate the concept.

1.6 Future Scope

The most important aspect to be considered in the future is to make the entire prototype wearable and to ensure that the fabrication costs involved due to it is low so that the product can be released in the market. This is because the cost of fabrication is quite high as it needs to ensure that the device is small, efficient and can be worn without causing any discomfort. Further the system must extend its variety in identifying the different postures and activities performed by the user.

Chapter 2

PROJECT ORGANIZATION

2.1 Software Process Models

The purpose of this chapter is to ensure that the project is organized by explaining the software process development model that is used and what are the roles and responsibilities of the team members. Extreme programming, a type of agile methodology, is best suited for this project as the product is developed in small increments by using test driven development with the help of pair programming. Basically there are four phases in the project namely designing, coding, testing, and listening. The process of developing the software is divided between two people and the task of interfacing the hardware and coding in Embedded C is divided between two people. Once the individual modules have been unit tested and developed, then they are integrated together. As late changes in the requirements are allowed, the idea to choose Extreme programming process model to organize the project is justified.

The overall process in XP is shown in Figure 2.1 and an iteration in XP is shown in Figure 2.2 respectively.

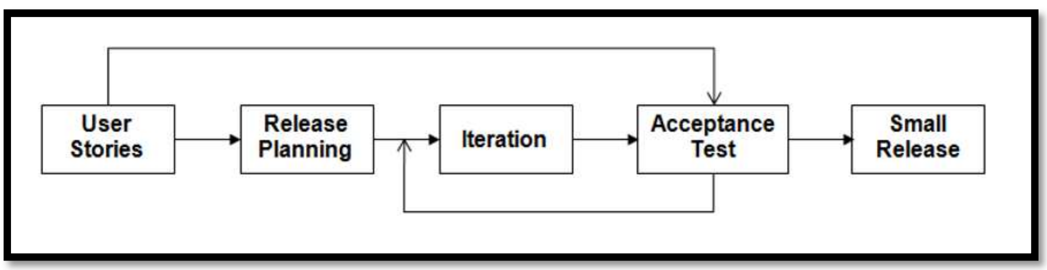

Figure 2.1 Overall process in XP

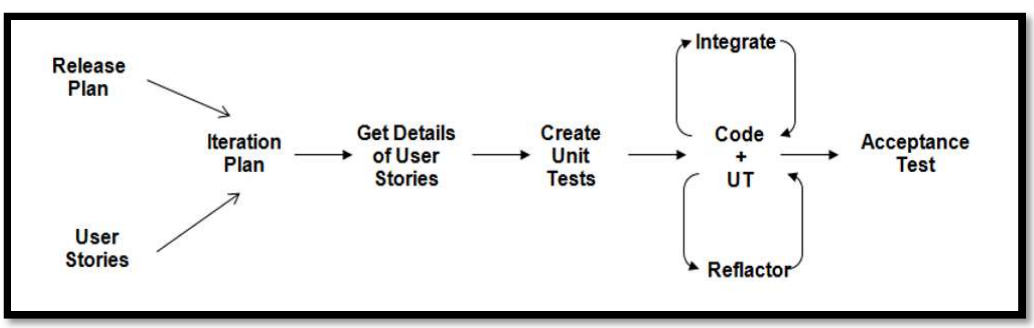

Figure 2.2 An iteration in XP

2.2 Roles and Responsibilities

- Sanjana K.S was instrumental in conducting brain storming sessions to determine the possibilities of extending and improving the project. Also she was responsible for purchasing the necessary hardware components, assembling the circuit and interfacing the accelerometer sensors and the LCD. She also helped in designing the prototype.
- Vidit Jain's role was to prepare the report and also helped Vignesh in developing the Android application. Also he suggested the idea of using a microcontroller than any IOT device such as Raspberry pi as internet connectivity cannot be expected to be present all the time. Further he also carried out the literature survey.
- Vignesh P is the team leader and was responsible for coding the Android application and also to perform unit testing of SMS module, Bluetooth Module, and Email module. Also he determined what sensors and other hardware components will best suit the project and played an important role in delegating work among the team members and prepared the project management plan.
- Vishal H's role was to perform integration testing and deployment. Further he coded the necessary hardware programs using Embedded C for interfacing the Bluetooth and for flashing the Embedded C code into the Renesas RL78 microcontroller in addition to determining the software requirements specifications.

Chapter 3

LITERATURE SURVEY

3.1 Introduction

The aim of this project is to monitor the activities performed by a patient which involves the knee joint, on an Android application, by using two accelerometer sensors and a Renesas RL78 microcontroller with a Bluetooth module. The accelerometer sensors generate an analog voltage output depending on the orientation of the sensor along x, y and z axis i.e. three dimensional accelerometer sensors. This output is then converted into a digital value by using an ADC and displayed on a LCD interfaced with the microcontroller. Later this value is sent via Bluetooth to an Android application which generates a pie chart indicating the various users' physical activities such as walking, sitting, running etc.

We propose this system as this idea was brainstormed by referring to various IEEE papers, Wikipedia and RL78 hardware manual as shown in Figure 3.1 and by using suitable ideas from them and also we have added some additional attractive features to the system such as providing an emergency button which when pressed triggers a buzzer and also ensures that the Android Application sends an Email alert with the location of the user by using the user's phone's GPS.

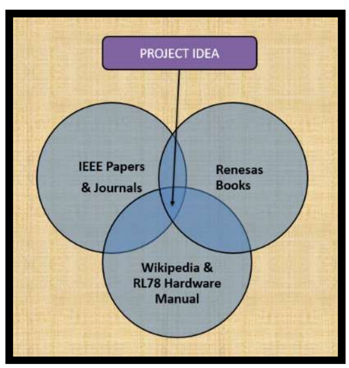

Figure 3.1 Project Idea

3.2 Related works

The field of Body Sensor Network (BSN) is very useful in constantly monitoring the body's movements in rehabilitation activities [1]. In our case we consider that the user might want to recover quickly after a knee operation and user along with the doctor needs to constantly monitor the different activities performed by them to ensure that there is not much of stress or strain applied on the knee due to excessive activity. Thus this data is very crucial for physiotherapists who ensure that the effectiveness of the rehab program is quite high and user achieves a timely recovery [2].

As our concentration is on monitoring the movements in the knee joint we have used a 3-axes accelerometer which gives more precise readings [1]. The first reference paper uses a 2-axes accelerometer sensor with an ATmega8 microcontroller as this paper aims at fall detection thus the readings of the sensors need not be very accurate and thus they have used a 2-axes variant. However, in our system the readings need to very accurate as it is used by the Android application to determine whether the user is sitting, standing, walking etc. Thus the aim of [1] is to estimate fall detection but the idea of using accelerometer sensors has been quite efficient and we would thus be using it in our system too. We found that the accelerometer sensors suit our requirements [3] [6] [7].

The aim of [2] is to determine the knee joint angle using gyroscope and flex sensors for rehabilitation purposes. From this paper we got the idea of monitoring body joints as Body sensor networks (BSN) is a field which is rapidly growing and needs relatively easy to use and comfortable wearable computing to monitor the body. Thus we are concentrating on monitoring the knee joint movement rather than the angle as it wouldn't be an effective way to monitor the user's movements. Instead we would be using a pie chart that attractively displays the data and yet effectively monitors the user's knee to ensure quick recovery. Renesas RL78 is a 16-bit microcontroller widely used in various industrial and medical applications [4] [5]. Thus we have chosen it for our system.

Further [3] gives an overview of the advantages and characteristics of using accelerometer sensors. This was a motivation for us to use this sensor in our system too. Also we wanted the user to actively monitor his knee by providing him with a very efficient GUI. Thus we came up with idea of an Android application which communicates with the Renesas Microcontroller [4]. The Android application also provides persistent storage by keeping a history of previous readings that it received via Bluetooth. This makes sure that ADL i.e. activity of daily living [1] is monitored on a long term basis. This is feasible by interfacing a Bluetooth chip with the microcontroller and thus it transfers the sensors data via Bluetooth and hence it's not dependent on internet connectivity. Also we have used an emergency button which when presses trigger a buzzer which can be deactivated by pressing the reset button. This ensures that if the user is in tremendous pain he can gather the attention of the people around him and also an Email alert is sent to the doctor from the Android application along with the patient's GPS location. In addition to this a SMS alert is also sent to the intended recipients.

The system that we have proposed is shown in Figure 3.2. It efficiently takes ideas from different papers and integrates them to achieve maximum benefit and also ensures that the design is cost effective but at the same time efficiently solves the problem at hand. Thus the tradeoff between performance and cost has been effectively managed.

3.3 Conclusion

The system that we have proposed is shown in Figure 3.2. It is light weight and can be used on a daily basis and will help in quick recovery. This is quite different from other knee joint monitoring systems as most of them that are available in the market are quite sophisticated to understand and use. Also it efficiently takes ideas from different papers and integrates them to achieve maximum benefit and also ensures that the design is cost effective but at the same time efficiently solves the problem at hand. Thus the tradeoff between performance and cost has been effectively managed. Further we are developing an Android application which communicates with the microcontroller via Bluetooth and thus doesn't need internet connectivity. Also persistent storage is provided by the application. The 3-axis accelerometer sensors give precise information when there's a knee movement. The UI is simple and yet performs all the functionalities efficiently. In case of any unforeseen events an emergency button and a buzzer has also been provided along with the LCD which constantly displays appropriate messages to keep the user informed if there are any technical glitches. By sending an Email alert (in addition to the SMS alert to the patient's relative) to the doctor along with the GPS position of the user it ensures that the user receives immediate care.

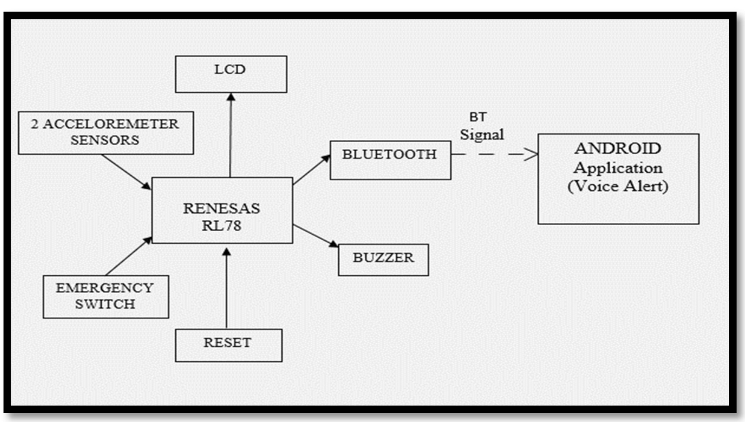

Figure 3.2 Proposed System

Chapter 4

PROJECT MANAGEMENT PLAN

4.1 Project Schedule

This chapter deals with scheduling the project using Gantt chart and risk identification and steps taken to mitigate these risks.

The overall project schedule is briefly summarized in Table 4.1.

Sl. No.	Task Name	Duration
1.	Problem Definition	6 days
2.	Project Plan	7 days
3.	Requirements Gathering and Analysis	7 days
4.	Design	7 days
5.	Implementation	34 days
6.	Testing	4 days
7.	Documentation	6 days

Table 4.1 Project Schedule

Table 4.1 has been summarized from the Gantt chart drawn for the project. A Gantt chart is effective in scheduling a project and graphically representing the deadlines and the tasks which are pending and those which have been completed. The Gantt charts are shown in Figure 4.1 and Figure 4.2 respectively on the next page. As we can see the main tasks include problem definition, project plan, Requirements gathering and analysis, Design, Implementation, Testing and Documentation. Each of these tasks in turn have some sub tasks under them.

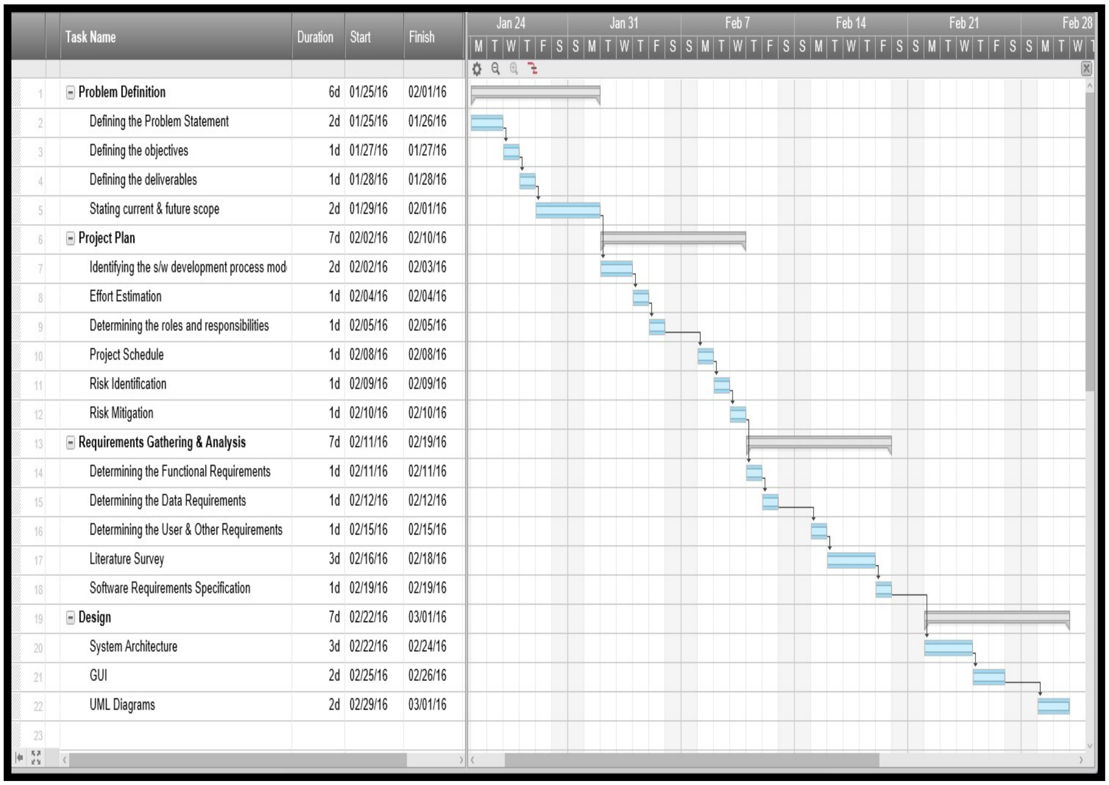

Figure 4.1(a) Gantt chart

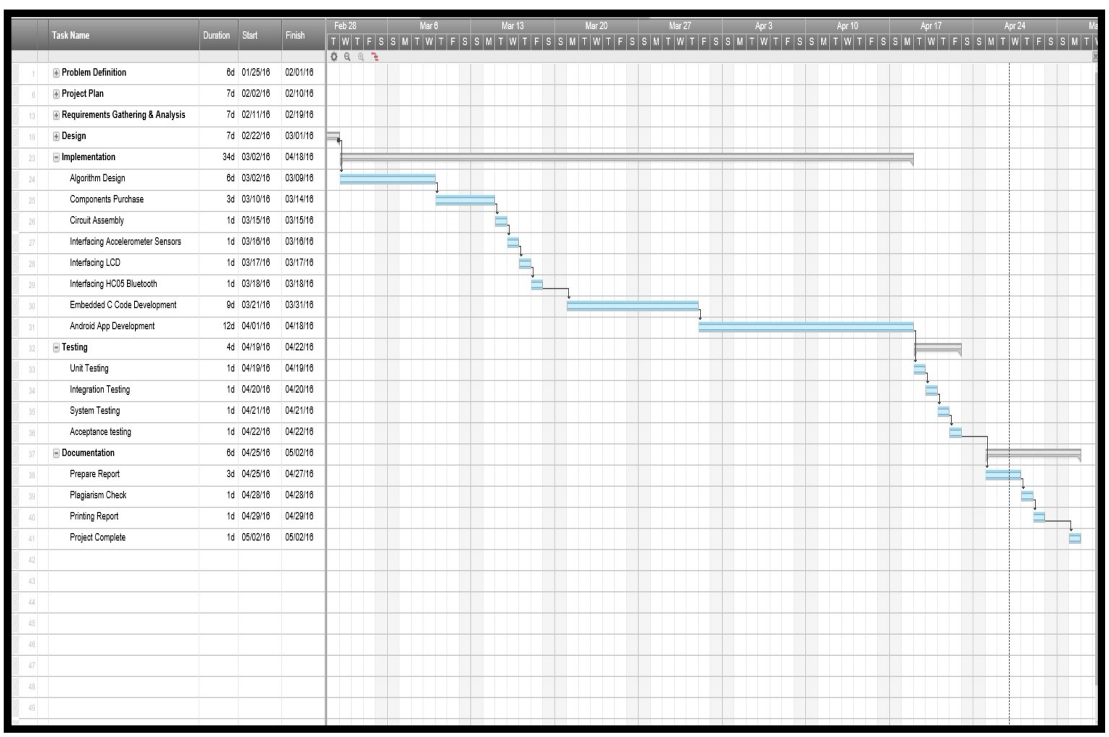

Figure 4.1(b) Continuation of Gantt chart

4.2 Risk Identification

The top five risks for this project 2 along with the mitigation steps taken for each of them are summarized in Table 4.2.

Sl. No.	Risk Item	Risk Mitigation Techniques
1.	Developing the wrong software functions.	Early user manuals, User surveys and developing the software in iterations (prototyping).
2.	Developing the wrong user interface. (Android Application)	Task Analysis, User Characterization, Scenarios and Prototyping.
3.	Changing project requirements.	Developing the software in small increments.
4.	Lack of technical and management skills.	Adequate training and also allocating tasks to people based on their capabilities and strengths, enhanced communication between team members
5.	Use of non-effective hardware components.	Clearly understanding the requirements and analyzing the scenarios, requirements scrubbing, cost benefit analysis.

Table 4.2 Risk Identification and Mitigation

Chapter 5

SOFTWARE REQUIREMENT SPECIFICATIONS

5.1 Product Overview

The aim of this product is to build an Android application which can monitor the activities of the patient which involves the knee joint with the help of Renesas RL78 MCU and accelerometer sensors. A HC05 Bluetooth module is used for transferring data between the MCU and the Android Application. When the patient is in tremendous pain due to unforeseen situations an Emergency button is provided which alerts the intended recipients via Email (along with the location of the patient using GPS) and SMS. A LCD is used to demonstrate the working of the entire prototype. A buzzer is triggered and a voice alert is played on the Android phone when the patient is in a prohibited posture which might affect the knee joint. Further the prototype to be built is wearable.

5.2 External Interface Requirements

5.2.1 User Interfaces

The user interface includes an Android Application and a LCD screen. The android application will be highly interactive, efficient, and attractive but would yet be simple and possess a sleek look. Also this interface will be highly user friendly and will perform two important tasks. One of them is that it would produce an audio message after the Bluetooth signal has been received from the wearable computing kit for an emergency alert. Another task is that it would enable the user to keep a track of the activities performed as this would be helpful for analysis and speedy recovery. The LCD screen displays the voltage value that is produced as the output by the microcontroller for the corresponding sensor inputs. The Android application shall be implemented using Java and Eclipse IDE with Android plugin. The LCD code is written using Cube Suite+ IDE in Embedded C and is flashed into the MCU using Flash Magic.

5.2.2 Hardware Interfaces

The hardware interfaces will include the need of wireless internet connectivity and GPS to send Gmail Alerts with the location of patient in case of unforeseen emergency events by pressing the Emergency button. Rensas RL78 Microcontroller is the hardware device which is responsible for co-coordinating various activities with respect to the project and is the core component. Further accelerometer sensors are used to measure the change in position of the knee and its output is fed to the input of the microcontroller. A Bluetooth module must also be interfaced. The LCD screen displays the output voltage of the microcontroller. A buzzer is triggered when the knee bends beyond a threshold angle.

5.2.3 Software Interfaces

The values are sensed from the accelerometer sensors which are then fed to the microcontroller. An Embedded C program is written to covert the sensor value to a corresponding voltage value as its output and also to display it on the LCD. Also this program with the help of the Bluetooth module sends messages pertaining to the activities performed by the patient to the Android application. Further Java and Eclipse IDE will be used to develop the Android Application to send both, the intended Gmail Alerts with the GPS location to the doctor and the SMS alerts.

5.2.4 Communication Interfaces

SMTP is the communication protocol used to send Email Alerts to the doctor which uses port number 25. Also a Bluetooth signal is used as the communication interface between the wearable computing kit and the Android Application running on the mobile device of the user. Also a voice message is played out to the user upon the receipt of the Bluetooth signal indicating the excessive angle of bend in the knee joint and serves as another type of communication interface between the Android Application and the user.

Figure 5.1 summarizes the external interfaces.

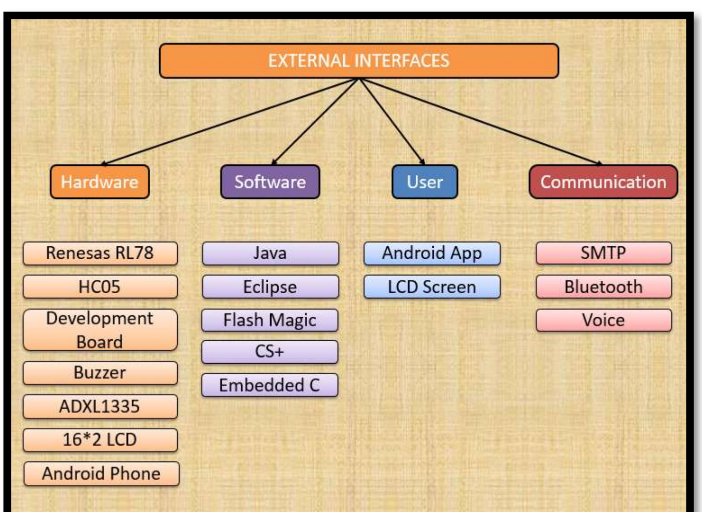

Figure 5.1 External Interfaces

5.3 Functional Requirements

5.3.1 Functional Requirement 1.1

The activity performed by the patient such as walking, running, sitting and standing needs to be constantly monitored as they apply stress on the knee joint

5.3.2 Functional Requirement 1.2

The MCU must have an Embedded C code flashed in it to detect the activity performed by the patient and it needs to be transferred via Bluetooth to the patient's phone.

5.3.3 Functional Requirement 1.3

In case if the angle of bend in the knee joint exceeds drastically due to some unforeseen event an automated message should be sent via Bluetooth which should play a voice alert on the phone and also the buzzer must be triggered.

5.3.4 Functional Requirement 1.4

If the user experiences a shock or sudden pain in the knee, then an Emergency button must be provided which when pressed must send alerts.

5.3.5 Functional Requirement 1.5

On pressing the Emergency button, an Email alert must be sent to the relative of the patient along with the location of the patient by using the phone's GPS.

5.3.6 Functional Requirement 1.6

On pressing the Emergency button, a SMS alert must also be sent to the physician.

5.3.7 Functional Requirement 1.7

An LCD must be present to display and demonstrate the working of the prototype.

5.3.8 Functional Requirement 1.8

The android application must accumulate data in a cumulative manner (unless the data is cleared) so that it can help the physician to analyze it for speeding up the recovery process.

5.4 Software System Attributes

5.4.1 Reliability

The reliability of the product depends on the lifetime of the accelerometer sensors and the accuracy of the measurement. As lifetime and accuracy is high the system is reliable. Further the android application would be rigorously tested to ensure that the application does its intended tasks in real time and doesn't generate fake alerts due to some bugs.

5.4.2 Availability

The basic functionality of the system except the Gmail Alerts is always available as long as the device is worn by the user as it depends on Bluetooth connectivity. For Gmail Alerts internet connectivity and GPS is needed. For SMS alerts balance talk time amount is needed.

5.4.3 Security

The scope for security in this product is more over concerned with privacy. Only the authorized phone of the user must be capable of receiving the alerts and more over the data readings stored in the android application must be kept safe and thus authentication should be added. The Android application shall not leave any cookies on the customer's phone containing the user's password. The Android application shall never display a user's password. It shall always be echoed with special characters representing the typed characters.

5.4.4 Portability

One of the features of Java is that it is architecturally neutral i.e. the code is machine or platform independent. Thus the same application would run on different Android phones and its backward compatible with respect to a specific Android phone version.

5.4.5 Maintainability

The Android application needs to maintained and updated in case of any bugs. Updated applications will be free from bugs and also some design oriented changes can be incorporated to make the interface more user-friendly and attractive.

5.4.6 Performance

The product's performance is measured in terms of responsiveness, efficiency and user experience. A product with high performance will be delivered. Performance is high if the throughput is high, latency is low, response time is less and system dependability exists. The changes in the patient's activity must be measured with accuracy in real time. Performance is also affected by the speed of internet connectivity for sending Email alerts.

5.5 Performance Requirements

The only way in which this system will meet its performance targets is for them to be specified clearly and unambiguously. It is a simple fact that if performance is not a stated criterion of the system requirements then the system designers will generally not consider performance issues. In order to assess the performance of the product the following are clearly specified:

- **Response Time-** It depends on the microcontroller used as it needs to execute the embedded program to convert the sensor value into a voltage signal and then needs to transmit the Bluetooth signal. The response time needs to be low.
- **Workload-** Even if the workload on the system is increased the performance doesn't degrade as Rensas RL78 microcontroller has a 16-bit CISC architecture. RL78 is designed specifically for ultra-low power applications enabling customers to build compact and energy-efficient systems at lower cost.
- **Scalability-** The product is highly scalable as it can be worn by various users if they have any problem in their knee and all that they need to do is to install the Android application on their phone.
- **Platform-** The microcontroller is Rensas RL78 that uses 16-bit CISC architecture and the Android application is built using Java and a framework like Android Studio.

5.6 Database Requirements

A database is used to provide persistent offline storage as it keeps a log of the activity performed by the patient. The database used needs to be a relational database, for example MySQL can be used and integrated with the Android application.

5.7 Design Constraints

1. **Space**: The amount of space occupied by the embedded program must be less as it needs to be incorporated into the Rensas RL78 microcontroller. Further the android application must be of size less than 100MB.
2. **Application memory Usage**: The amount of memory used in cache as well as main memory usage must be low while executing the embedded program. This is to meet the requirements of the microcontroller.
3. **Budget**: The amount of money that can be spent on the hardware must be less than Rs3000/- so that the overall price is quite reasonable.
4. **Application Quality:** The quality of the product must be good and also it must be comfortable to wear it on the knee.

5.8 Other Requirements

- **Android version:** The application developed must be simulated on the latest Android version and also it needs to be backward compatible thereby supporting a variety of phones and Android OS versions.

Chapter 6

DESIGN

6.1 Introduction

In order in to optimize the design process and also to ensure that the design is efficient and the final product designed meets the desired requirements and achieves the goals and objectives of the project, it is essential to divide the project into various components or modules whose description has to be defined clearly.

Thus this chapter initially describes the different modules present in the project. Later the system architecture is described which is followed by the description of the GUI and then various UML diagrams like class diagram, sequence diagram and data flow diagram are given.

The different modules needed and their description for the Knee Joint Movement Monitoring system are given below

- **Accelerometer sensors Module**: This module is responsible for identifying the change in the angle of bend in the knee of the user. Accelerometer sensors are interfaced with the Renesas RL78 microcontroller as they produce an analog voltage output which depends on the orientation of the sensors with respect to the x, y and z axis.
- **Microcontroller Module:** This module is the most important module in the project. An embedded c program is written and flashed to the microcontroller. This program ensures that the data is being read from the accelerometer sensors and also make sure that this analog voltage is converted to a digital value and is sent to the Android application using the Bluetooth module. Further suitable messages are constantly displayed on the LCD screen which is also interfaced to the microcontroller. Depending on the voltage value the user’s activity is identified as when the user is sitting the two accelerometer sensors will be perpendicular to each other and while standing they will be parallel and similarly the position of the sensors changes for walking and running. Thus using the voltage values given by the sensors the user activity is determined by the Android Application.

- **Bluetooth Module**: This module is responsible for interfacing a Bluetooth chip with the Renesas Microcontroller to ensure that data communication takes place between the Android Application running on the user's phone and the Microcontroller.
- **Database Module**: This module's function is to insert the data into the Android Application and provide persistent storage for all the activities performed by the user and will be helpful when the user or the doctor needs to monitor the user's physical activities which affect the patient's knee.
- **Email Module**: This module's function is to send an Email alert to the intended recipient's Gmail account along with the current position of the user which is obtained using GPS when the user presses the emergency button.
- **SMS Module**: A SMS must be sent to the intended recipient when the emergency button is pressed.
- **Android Application Module**: This module's function is to pictorially represent the angle of bend in the knee using a pie chart. Different activities such as walking, running, sitting and climbing are taken into consideration and are plotted as a pie graph which indicates how many times each activity is performed by the user. Further the application has a login page for authentication and also plays out a voice alert when the user presses the emergency alert button which is provided and interfaced with the Renesas RL78 microcontroller. Also the Android Application has a record of all the activities performed by the user in the past.

6.2 Architecture Design

The system architecture is given in Figure 6.1. The dotted line between the Bluetooth module and the Android application indicated the Bluetooth signal used for communication between the two and also a voice alert is played when the emergency button is pressed. All incoming arrows to Renesas RL78 indicate input features and all outgoing arrows indicate output features respectively. The dotted line between Android application and Email Alert indicates the usage of Internet connection to send the alert. The dotted line between the Android application and the SMS alert indicates the usage of some money in sending the message. Figure 6.2 shows the flow diagram for this project.

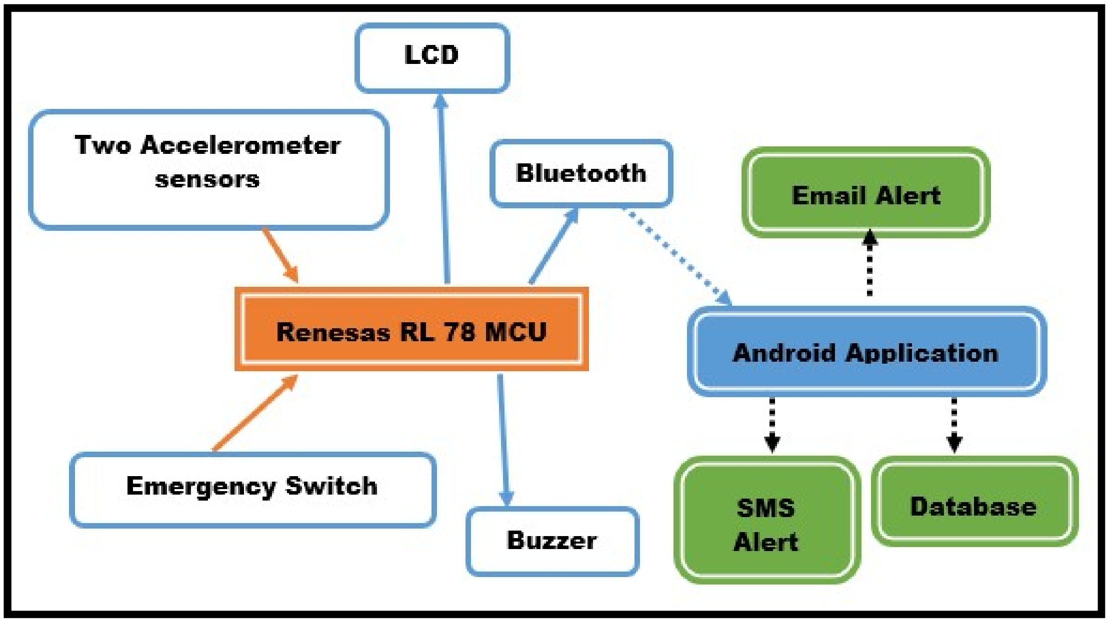

Figure 6.1 System Architecture

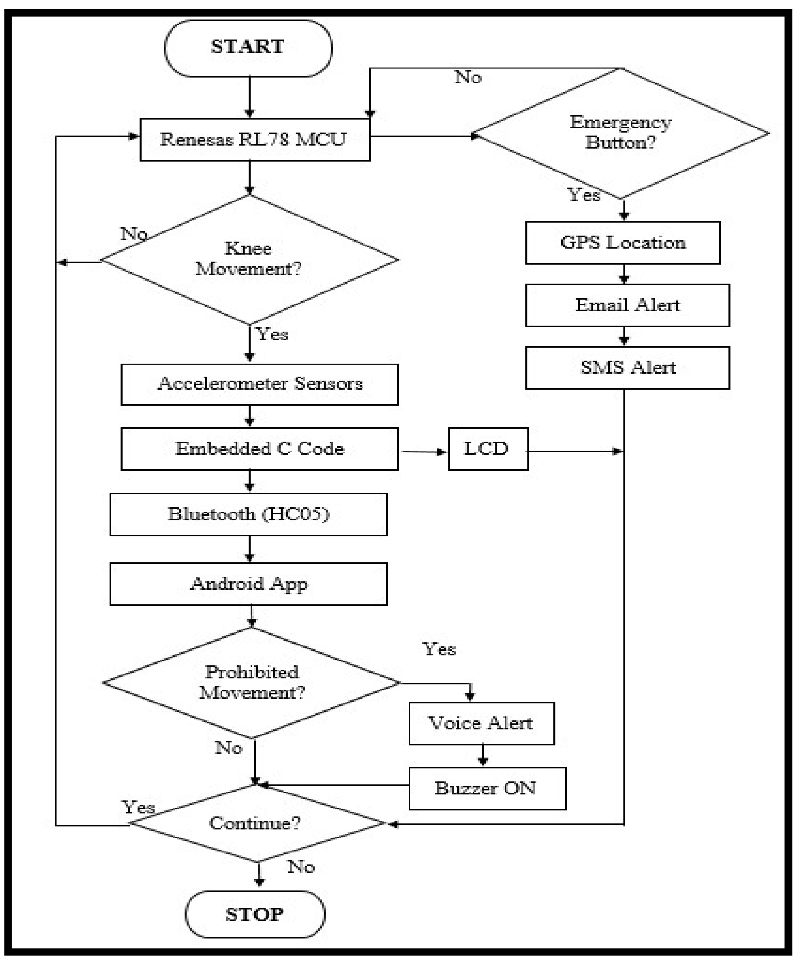

Figure 6.2 Flow Diagram

6.3 Graphical User Interface

The main graphical user interface is an Android Application which communicates with the Renesas RL78 microcontroller via Bluetooth. This application initially has a welcome login page for authentication with two buttons, namely login and clear. When the user enters the correct username and the password the login succeeds and redirects the application to the main page. If either the username or the password is invalid, then it prompts the user to enter the correct credentials. The pie chart is generated depending on the voltages values of the 2 accelerometers received via Bluetooth from the microcontroller. Thus the user's physical activities such as standing, walking, sitting, running etc. are identified on the basis of the voltage values and a pie chart is generated to indicate the count of each activity performed by the user. The GUI has been planned to be designed very efficiently so that its user friendly and at the same time has a good look and feel. The different functions used are *loginButton, clearButton, viewData, processData, and generatePie*. Thus the Android application acts as the software graphical user interface.

Also there is an LCD which is directly connected to the Renesas Microcontroller which provides a restricted user interface as it allows the user to only view the data being displayed on it. The LCD is used to display some initial configuration messages, the digital voltage values and also some success or failure messages regarding the transmission of data via Bluetooth and whether or not different components such as the accelerometers and the Bluetooth have been interfaced properly or not and also it displays an alert message when the user presses the Emergency button which triggers a buzzer and this alert is stopped when the user presses the reset button. This functionality is achieved using embedded c program which is developed using Cube Suite+ and flashed into the microcontroller using Flash Magic software. The different functions used in the embedded C program are *initLCD, displayLCD, clearLCD, buzzTrigger, initBluetooth, Ebutton and Rbutton*. These functions perform the tasks defined above. Thus the LCD along with the buzzer, emergency button and the reset button acts as a hardware graphical user interface.

6.4 Class Diagram

The relationship among various classes and their interdependencies are effectively modeled using the class diagram by using association, aggregation, composition and generalization which is shown in Figure 6.3.

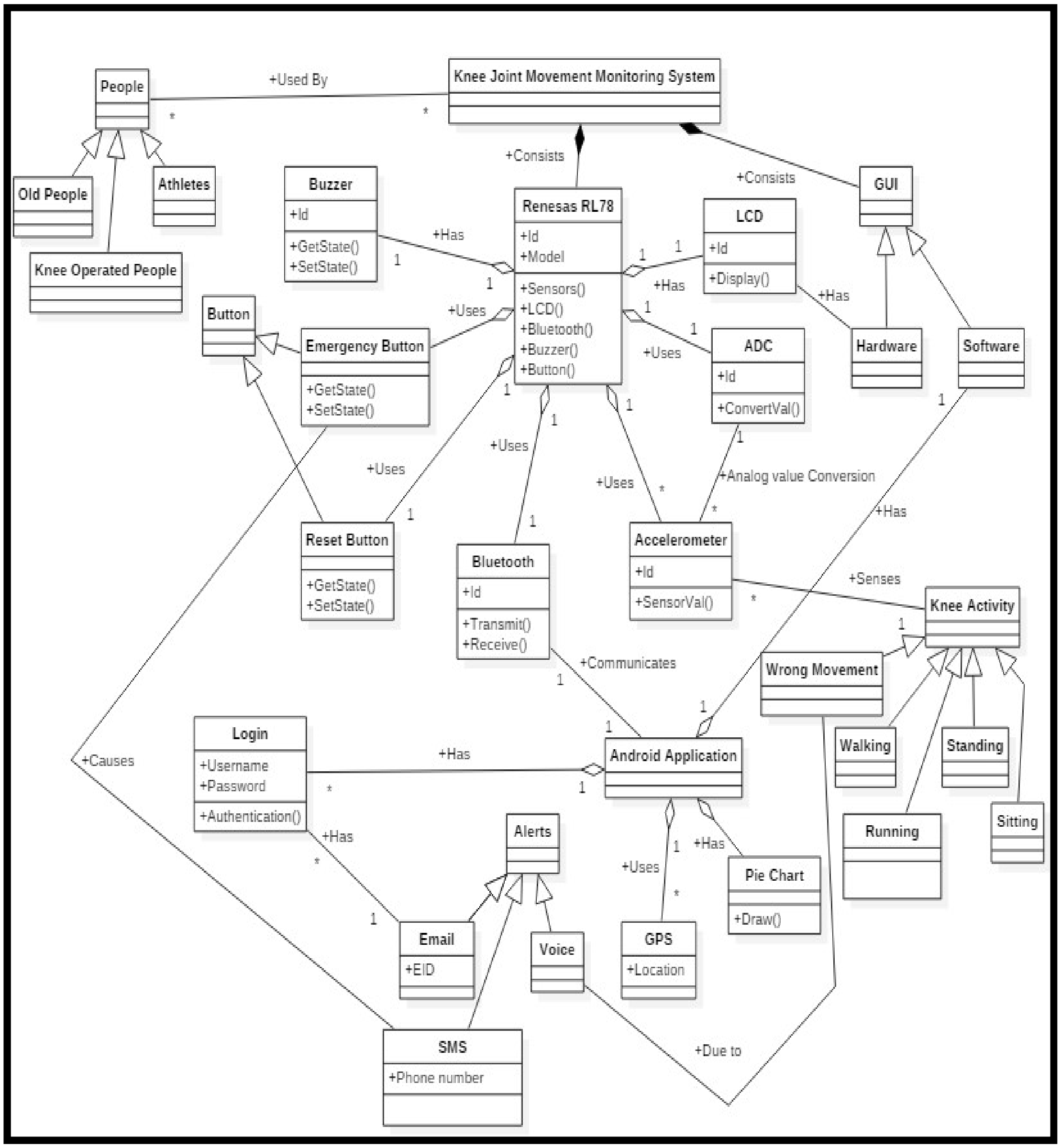

Figure 6.3 Class Diagram

6.5 Sequence Diagram

The sequence diagrams for this project are given below in Figure 6.4, Figure 6.5 and Figure 6.6.

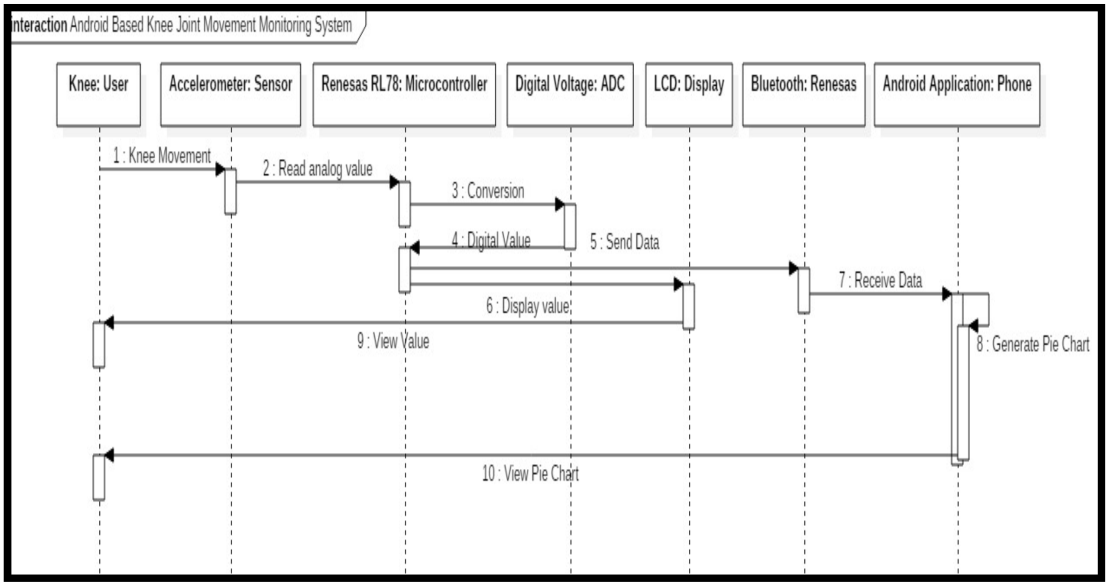

Figure 6.4 Sequence Diagram for the Overall System

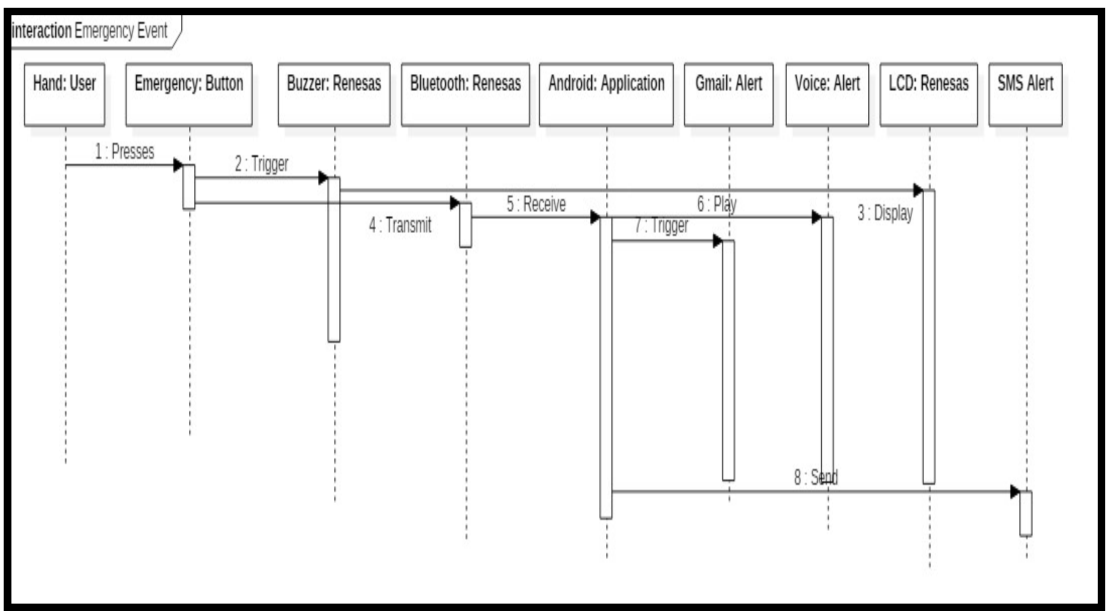

Figure 6.5 Sequence Diagram for Emergency Event

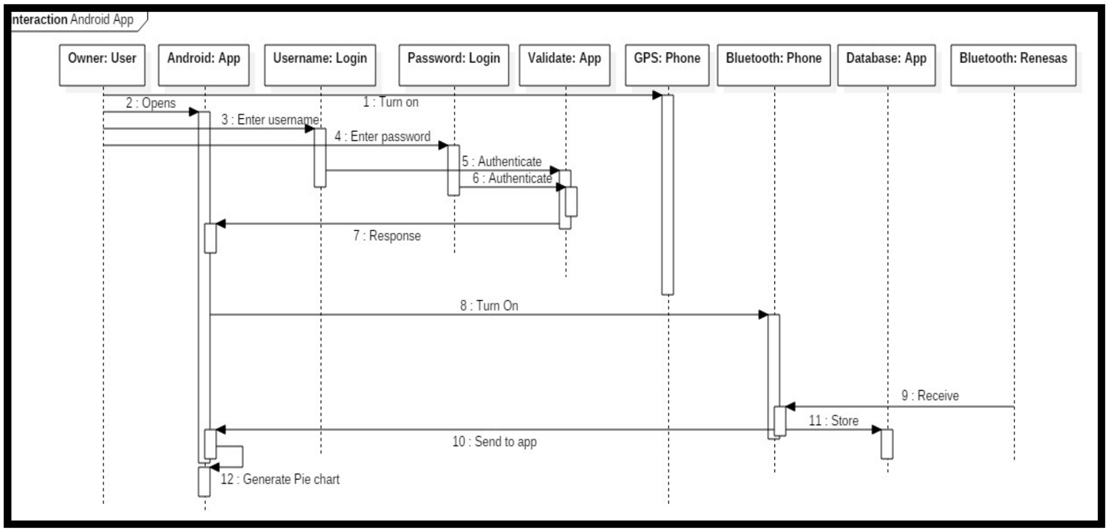

Figure 6.6 Sequence Diagram for Android Application

6.6 Data Flow Diagram

The data flow diagram for this system is given in Figure 6.7.

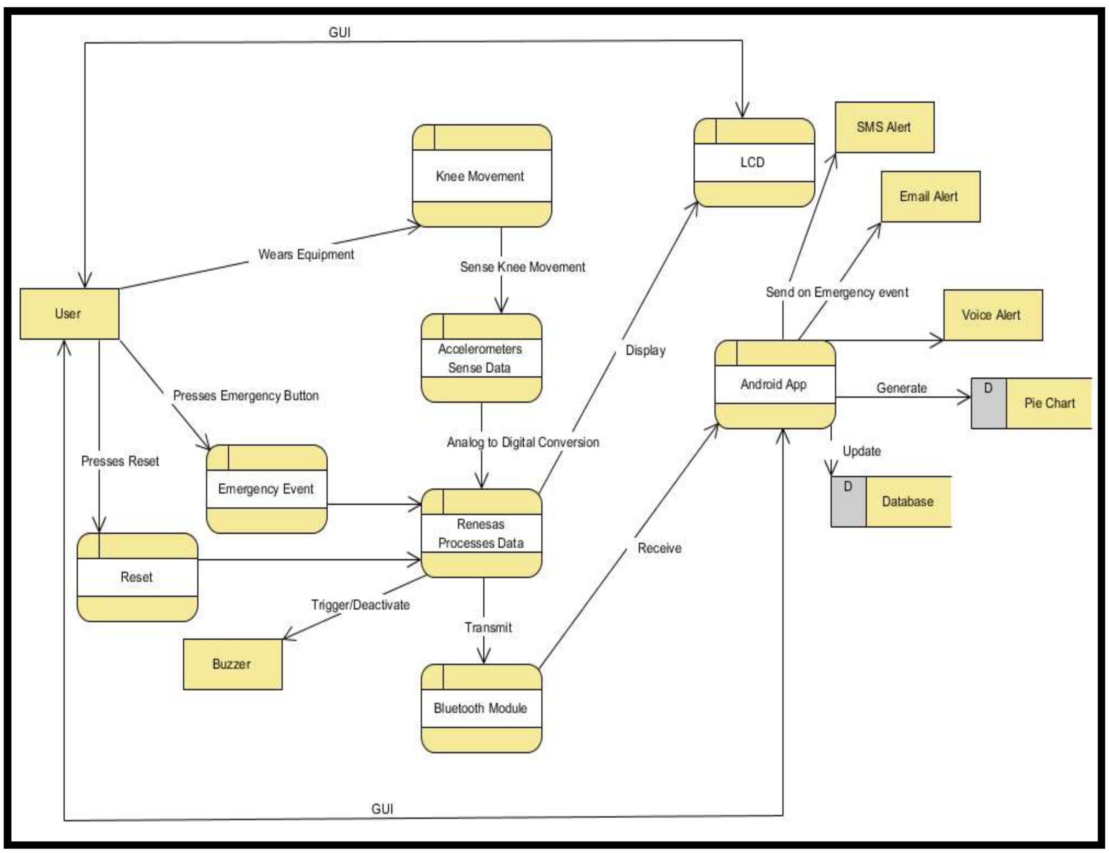

Figure 6.7 Data Flow Diagram

6.7 Conclusion

As the design has been completed and a thorough understanding of what needs to be implemented is known with the help of the system architecture and the UML diagrams such as class diagram, sequence diagram and data flow diagram, coding can now be started and it's discussed in the next chapter. This chapter deals with the specifications of what needs to be rather than how it is done. The aspect of how the features need to be implemented is given in the next chapter which deals with the coding details.

Chapter 7

IMPLEMENTATION

7.1 Tools Introduction

The hardware components needed to build this prototype are as follows.

- **Renesas RL78/G12 microcontroller** is the core component of this system. RL78 is a 16-bit microprocessor which has a CISC architecture. Flash memory is the type of memory that is present inside this MCU. We are using a 30 pin variant of this MCU along with 16kb ROM capacity. It has a variety of household applications. Thus this MCU is named according to the Renesas Standard as R5F102AAA and belongs to the G12 family. The meaning of this name is shown in Figure 7.1. The Renesas development board is used for the project and is shown in Figure 7.2.

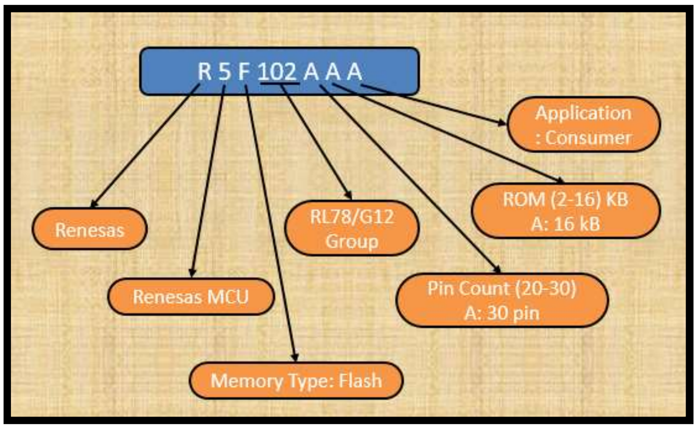

Figure 7.1 Meaning of the microcontroller's name

Figure 7.2 Renesas RL78/G12 Development Board

- **3 axes Accelerometer sensors (ADXL335)** are the ones used in this prototype to determine the activity of the user by placing one of them above the knee joint and the other one below. They work on the basis of gravitational force rather than acceleration force. We are using a capacitive variant of this sensor. Thus whenever there is a motion in the knee, the movable plate of the capacitor moves towards one of the fixed plates which produces a difference in capacitance across the fixed plates which is amplified to produce an analog voltage as the output. This is shown in Figure 7.3.

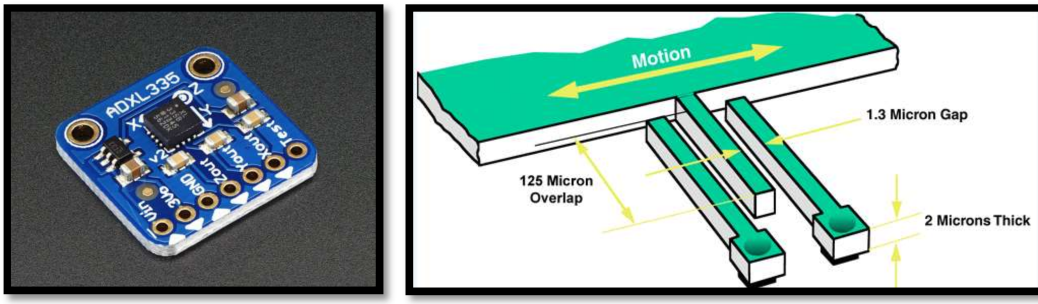

Figure 7.3 ADXL335 Accelerometer sensors (3 axes-x, y, and z)

- **HC05 Bluetooth module** is a type of SPP (serial port profile) module and is interfaced to the RL78 MCU using UART (Universal Asynchronous Receiver Transmitter) to provide communication between the android application and the MCU. Figure 7.4 shows the HC05 Bluetooth module.

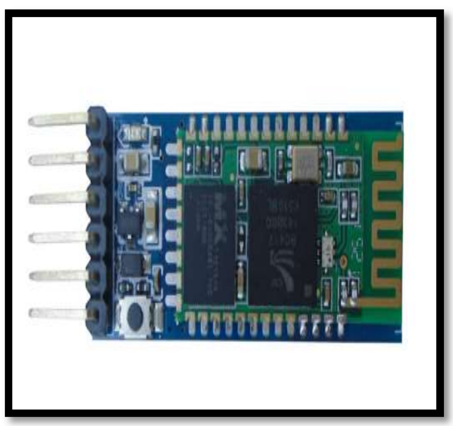

Figure 7.4 HC05 Bluetooth Module

- **NAND flash board** is needed to flash the Embedded C code into the Renesas MCU. This board is shown in Figure 7.5. NAND flash is a type of flash memory which does not need power to retain the data.

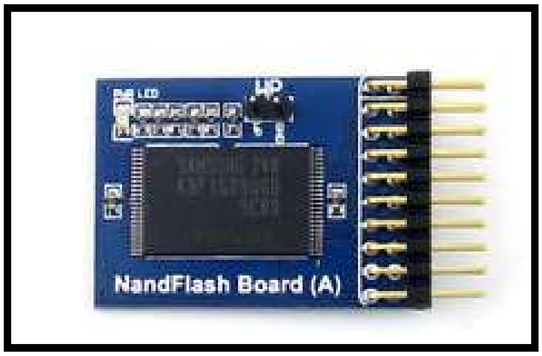

Figure 7.5 NAND flash

- In addition to these main components, one 16*2 LCD, buzzer, an android phone and a knee cap is used. These are shown in Figure 7.6. Also a 5v power cable, flash cable and few jumper wires are needed to implement the prototype. The sensors are placed on the knee cap so that the prototype is wearable. The LCD is used to demonstrate the working of the prototype.

Figure 7.6 Additional Components

7.2 Technology Introduction

- Bluetooth (SPP also known as serial port profile) is the technology used for communicating between the Android application and the Renesas board.
- The accelerometer sensors used are capacitive in nature.
- The embedded C program is written using an IDE called Cube Suite+. The purpose of this code is to read the analog voltage output from the sensors and to convert them into digital values and to send them for display to the LCD and also to the send them to the android application with the help of HC05 Bluetooth module.
- The android application is written in Java using Eclipse (Kepler) with android plugin (ADT) and SDK's.
- Software called Renesas Flash programmer is used along with the NAND flash to flash the Embedded C code into the MCU.
- Flash Magic is software that is used for testing the HC05 Bluetooth module.
- Internet is used for sending Email alerts.
- GPS (Global positioning system) is used for determining the location of patient.

7.3 Overall view of the project in terms of implementation

Embedded C code is written using Cube Suite+ IDE to perform the following functions

- Read the analog voltage output of the sensors and convert them to digital values.
- Determine the posture or activity of the patient from these values and assign a unique code for it.
- Display these digital values along with the activity on the LCD.
- Send the unique code to the Android application via Bluetooth with the help of HC05.
- Trigger the buzzer when the patient does a knee movement which is prohibited.
- Display Emergency Alert on the LCD when the emergency button is pressed.

The Android application is written using Eclipse IDE to perform the following functions

- Connect and pair to HC05 module by turning on Bluetooth.
- It has a login page to authorize the user.
- Also a field called Emergency number is present to send SMS alert when the emergency button has been pressed.

- To generate a pie chart.
- To clear the existing data.
- To send an Email alert along with the location of the user using internet and GPS.

Figure 7.7 gives the Overall view of the project in terms of Implementation.

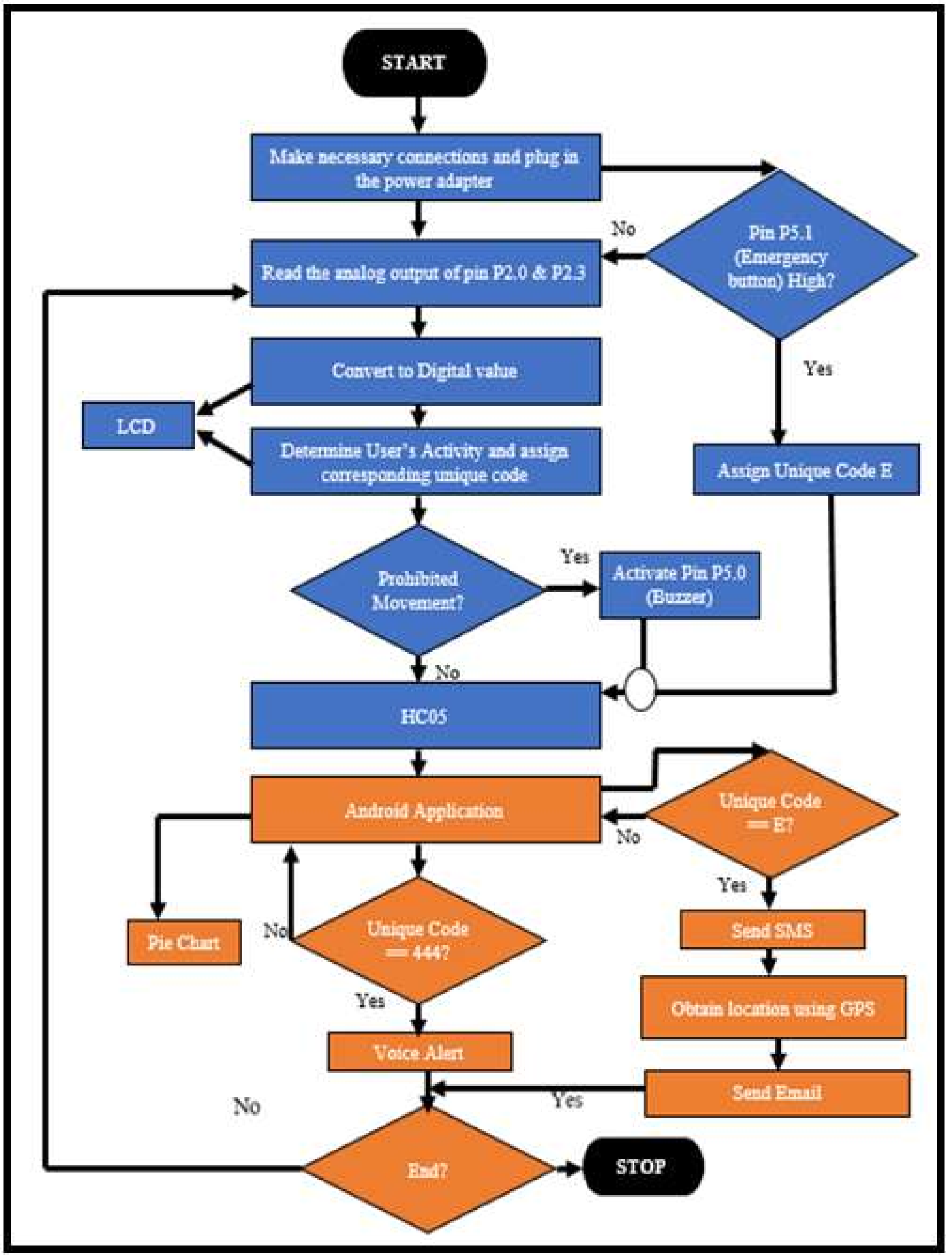

Figure 7.7 Overall View in terms of Implementation

In the above Figure, all the blue colored boxes correspond to the tasks performed by the embedded C code and manually whereas the orange colored boxes correspond to the tasks performed by the Android Application.

7.4 Algorithm

This algorithm deals with the hardware aspects of the project.

Input: Renesas MCU with all necessary connections and prototype worn by user.

Output: Determining user's knee activity and sending data.

1. Avolt ← Sensor Value due to knee motion. // Avolt= analog voltage output
2. Dvolt ← ADC(Avolt) // Dvolt= Digital Voltage
3. DispLCD(Dvolt) // Function to display data on LCD
4. If (Dvolt >= Ut && Dvolt<= Lt) // Ut= Upper Threshold; Lt=Lower threshold
    I. DispLCD("Walking")
    II. UART_Send("Unique_Activity_code")
5. If (Dvolt >= Ut1 && Dvolt<= Lt1)
    I. DispLCD("Running")
    II. UART_Send("Unique_Activity_code")
6. If (Dvolt >= Ut2 && Dvolt<= Lt2)
    I. DispLCD("Sitting")
    II. UART_Send("Unique_Activity_code")
7. If (Dvolt >= Ut3 && Dvolt<= Lt3)
    I. DispLCD("Standing")
    II. UART_Send("Unique_Activity_code")
8. If (Dvolt >= Ut4 && Dvolt<= Lt4)
    I. DispLCD ("Wrong Movement")
    II. UART_Send("Unique_Activity_code")
    III. Buzzer = On
    IV. Delay(5s)
    V. Buzzer = Off
9. If (Ebutton == High)
    I. UART_Send("Unique_Emergency_Code")
10. Loop to Step 1.

The above algorithm deals with the hardware code which is flashed into the Renesas MCU and is written using Cube Suite+ IDE in embedded C. The ADC is a successive approximation ADC which produces a digital output by using the equation 7.1

$$D\_val = ((A\_val * Vref) / (2^n - 1)) * 100 \quad \ldots\ldots (7.1)$$

From D_val, each digit is extracted and converted to its ASCII value for displaying on the LCD.

In the Equation 7.1, Vref= 5v and 100 is a constant and A_val is the Analog value and D_val represents the Digital value of the voltage. Table 7.1 gives the details of the activity name and the corresponding unique code that is sent as we are using serial communication and it would be better to use small codes rather than sending the activity names.

Sl. No.	Activity Name	Unique Code
1	Walking	111
2	Running	222
3	Sitting	333
4	Standing	555
5	Emergency	E
6	Wrong Move	444

Table 7.1 Unique Codes for each activity

Table 7.2 gives the details of the commands for implementing the LCD.

Sl. No.	LCD Function	Command
1	2lines and 5*7 matrix	0x38
2	Display ON, Cursor blinking	0x0E
3	Clear display Screen	0x01
4	Shift cursor right	0x06
5	Data Instruction	RS=1
6	Command Instruction	RS=0

Table 7.2 LCD Details

The algorithm mentioned below deals with the Android application.

Input: Data from HC05 module

Output: Knee Joint Monitoring Application

1. Enter username and password.
2. If (uname == “admin” && pwd == “*****”)

    [1] Logged in successfully.

    [2] Bluetooth = On

    [3] Connect to Knee cap

    [4] If (Bluetooth Key =” ****”)

        Connected successfully.

        Code ← Received data

        If (code == 111)

            Wc++ // walking count

        Else if (code == 222)

            Rc++ // running count

        Else if (code == 333)

            Sitc++ // sitting count

        Else if (code == 555)

            Stdc++ // standing count

        Else if (code == 444)

            Wmc++ // wrong movement count

            Play Voice alert

        Else if (code == E)

            Send SMS

            Fetch Location using GPS

            Send Email Alert

    [5] Else

        Wrong Bluetooth Key

3. Else

    Wrong Username or Password

7.5 Information about the implementation of modules

With the help of the algorithms written in the previous section the hardware code was implemented using Cube Suite+ in embedded C. There are three classes namely the main class, Knee_ADC and Knee_LCD. Knee_ADC performs the task of converting the analog voltage output of the accelerometer sensors into digital value using equation 7.1 and then they are converted into their respective ASCII characters for displaying on the LCD module. Knee_LCD performs the task of initializing the LCD and sending data as well as command instructions to display the data using the information given in Table 7.2. The main class detects the various activities of the patient and sends corresponding unique codes mentioned in Table 7.2 and also is responsible for triggering the buzzer in case of wrong movements. It also makes calls to the functions present in Knee_ADC and Knee_LCD. In addition to these certain libraries have been imported to support certain features of LCD.

The android application has been implemented using Eclipse (Kepler) with Android ADT and SDK plugins. It has 9 classes namely BluetoothSPPConnection, BluetoothSPPConnectionListener, Child_Movement_Graph_activity, Constants, DatabaseHelper, DeviceListActivity, Login_activity, Mail and MainActivity. BluetoothSPPConnection is responsible for connecting the android application to the HC05 Bluetooth module so that data can be transferred. Child_Movement_Graph_activity performs the function of constructing the pie chart with the values of the following arrays: colors, ar_values and ar_namelist. Constant is used for defining and initializing the counter. DatabaseHelper inserts the activity name and count into a SQL database so that it can be fetched for drawing the pie chart. However, if the data is cleared then the database is truncated and the pie chart will be drawn using the new values. DeviceListActivity in conjunction with BluetoothSPPConnection is responsible for connecting and pairing to HC05. Login_activity is responsible for handling the events pertaining to the login page of the android application. Mail is used for sending email alerts. MainActivity is the core class which calls the functions from the other classes. Also its computes the location of the patient by using GPS and sends SMS alerts in case of unforeseen situations

7.6 Conclusion

The main aim during implementation was to keep the code size as small as possible. Testing was done initially before implementation as we are using extreme programming software development model which encourages test driven development. So initially we had written test cases using tools like Junit and Robotium for the different modules of our prototype and ran it on an Android Virtual Device. Later the actual code was written and integration testing was performed. The next chapter deals with testing.

Chapter 8

TESTING

8.1 Introduction

Software testing is defined as "A process of executing a program or application with the intent of finding the software bugs". It can also be stated as "The process of validating and verifying that a software program or application or product meets the business and technical requirements that guided its design and development, works as expected and can be implemented with the same characteristic".

In our project we have used two different programming languages namely Embedded C for programming the microcontroller (Renesas RL78/G12) and Java for developing the Android application. Thus we used Junit framework for testing the various components of the application and a tool called Flash Magic was used for testing the HC05 Bluetooth module. The testing of hardware components was manually done with the help of Cube Suite+ whereas for the android application, it was automated using Junit framework. JUnit is a "unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit that originated with Junit".

Also the Android Virtual Device Manager (AVD) was used to perform integration testing on the unit tested components of the Android application. An Android Virtual Device (AVD) is "an emulator configuration that lets you model an actual device by defining hardware and software options to be emulated by the Android Emulator". Apart from performing functional testing AVD also aids in nonfunctional testing (compatibility or portability testing) as it can simulate different android phones from brands like Nexus, Samsung, and Nokia etc.

A framework called Robotium has been used for performing automated testing on the Android Application in conjunction with Junit. Robotium is "An open-source test framework for writing automatic gray box testing cases for Android applications. With the support of Robotium, test case developers can write function, system and acceptance test scenarios, spanning multiple Android activities". When running the test case on the emulator or an actual device, you will see the values being entered into the UI components or see them being clicked as if you were doing it yourself. After the tests are completed, the JUnit view in Eclipse will show which tests have failed and passed.

8.2 Testing tools and environment

The different testing tools and environment used are as follows:

- Flash Magic: This tool is used to unit test the HC05 Bluetooth module to ensure that the process of sending and receiving messages via HC05 works correctly.
- Junit: It's a unit testing framework used for testing different components of the Android Application. Various test cases can be run and assertions can be made to indicate if the tests failed or succeeded.
- Robotium: This framework is used for automating test cases. When running the test case on the emulator or an actual device, you will see the values being entered into the UI components or see them being clicked as if you were doing it yourself. After the tests are completed, the JUnit view in Eclipse will show which tests have failed and passed.
- Cube Suite+: This is an IDE for developing embedded C programs for Renesas microcontrollers. However, this IDE has some options to enable pins and thus helps in emulating the Renesas MCU RL78.
- Android Virtual Device Manager: This simulates the environment for running an android application and has been used to test SMS module and login module.

8.3 Test Cases

1. Pairing to HC05 Bluetooth Module and Transfer data (Using Android Phone)

Sl. No.	Test Case Objective	Execution steps	Test Input	Expected Output
1	Enter invalid password	NA	Password:***	Failed to connect
2	Enter correct password	NA	Password:*****	Allow user to proceed
3	Connect to device	Click Connect	NA	Device Paired

Table 8.1 Test case1

A Software called Flash Magic can be used to test the HC05 Bluetooth module. By shorting the Rx and Tx pins i.e. the receiver and the transmitter pins of HC05, the message that is sent is delivered to HC05 itself.

Once the devices are paired, i.e. HC05 and the android phone, messages can be sent depending on the accelerometer sensor values which correspond to various activities such as walking, running, sitting, standing etc. Refer Figure A.1 to A.3 for this test case.

2. Login to Android Application (Using Android Virtual Device)

For this test case refer Figure A.4 to A.9.

Sl. No.	Test Case Objective	Execution steps	Test Input	Expected Output
1	Enter invalid password and/or invalid username	Click Login	Password:*** Username: adman	Login Failed
2	Enter correct password & correct username	Click Login	Password:***** Username: admin	Login Successfully, Allow user to proceed
3	Clear Data in the fields	Click Cancel	NA	Data Cleared

Table 8.2 Test case2

3. Android Application's Portability/Compatibility (Using Android Virtual Device)

For this test case refer Figure A.10 and A.11.

Sl. No.	Test Case Objective	Execution steps	Test Input	Expected Output
1	Check if application runs on various phones	Run in AVD with different Android versions and phones	NA	All features of the application are working

Table 8.3 Test case3

4. Sending SMS alerts (Emergency Based using Android Virtual Device)

For this test case refer Figure A.12 and A.13.

Sl. No.	Test Case Objective	Execution steps	Test Input	Expected Output
1	Check if SMS is sent to the intended recipient	Enter the emergency number using the application and press the emergency button to send the alert	Text Message: Emergency alert	SMS delivered to the recipient

Table 8.4 Test case4

5. Sending Email and Voice Alert (Emergency Based)

For this test case refer Figure A.14 and A.15.

Sl. No.	Test Case Objective	Execution steps	Test Input	Expected Output
1	Check if email is sent to the intended recipients	Press the emergency button to send the alert.	Email Message: Emergency alert + GPS location (latitude ,longitude)	Email delivered to the recipients
2	Voice alert should be played	Perform Wrong movement	NA	Voice message played on phone

Table 8.5 Test case5

6. Testing Interfaced Hardware components

The LCD is tested using the Embedded C program written in Cube Suite+ which is flashed into the MCU using NAND flash. The purpose of this test program is to display a welcome message on the LCD to ensure that it's been interfaced properly. Later this program is extended to read the values from the accelerometer sensors and to display these values on the LCD.For this test case refer Figure A.16 and A.17.

8.4 Module wise Defect Distribution

There are five defects which were identified and are listed below with respect to the module that they belong in the system.

- Android Application

  Login page: Invalid Username or password.

  Whenever the patient enters a wrong username or password, the Android application doesn't show an invalid prompt. There is no response sent from the Android application when this happens.

- SMS Module

  Messages are sent after a long delay after the patient presses the emergency button.

- Email Module

  Email is not sent but it was overcome by turning on access for less secure applications. But the GPS location is being received as null as being

- Voice Alert

  It's played when the user presses the emergency button and it seems trivial and unnecessary to play a voice message.

- Wrong Movement

  The application doesn't show these wrong movements in the pie chart.

The task of resolving the defect in the Email module is given in Figure A.18 in the appendix. It can be seen that Emails cannot be sent from devices like Raspberry pi, custom made android apps, etc. as they are considered to be less secure according to the Gmail's policy. In the next section a table is given which highlights the above mentioned defects along with their status and severity. Some of the defects have been resolved and the aim is to resolve the remaining ones as soon as possible. Finally, this chapter ends by mentioning the different types of testing that was performed by using the various testing tools and environment mentioned in 8.2.

8.5 Number of Defects Identified and their status and severity

Table 8.6 gives the number of defects identified along with their status and severity.

Sl. No.	Defect Name	Defect Description	Severity	Status
1	Invalid Username or password (Android Application)	When the user enters an invalid username or password the application doesn’t prompts the user to enter it again	Low	Active
2	SMS module	SMS alerts sent after a long delay	Medium	Active
3	GPS Location	GPS attached with the email message was null	High	Resolved
4	Voice Alert	Voice alert played when user presses the emergency button	Low	Resolved
5	Wrong Movement	The pie chart in the android application doesn’t show the wrong movements	High	Resolved

Table 8.6 Details of Defects

8.6 Types of Testing

- **Gray box automated testing:** Robotium is open source gray box automated testing tool that has been used for running 4 test cases on the android application. Automated testing tools "Are capable of executing tests, reporting outcomes and comparing results with earlier test runs. Tests carried out with these tools can be run repeatedly, at any time of day". Gray box testing, also called gray box analysis, is a strategy for software debugging in which the tester has limited knowledge of the internal details of the program. Thus Gray-box testing is a combination of white-box testing and black-box testing. The aim of this testing is to search for the defects if any due to improper structure or improper usage of applications.
- **Manual Testing:** In order to test the hardware components such as HC05 Bluetooth module, LCD, Buzzer etc. In manual testing the tests aren't executed by machines. Instead they need to be carried out physically.
- **Unit Testing:** Junit is an open source unit testing framework which has been used for testing the SMS module, running Robotium tests etc. In unit testing the individual modules are tested for their functionalities and are later integrated.
- **Nonfunctional Testing (Portability Testing):** By using the android virtual device manager various android virtual devices can be created with different versions of the Android OS and hardware specifications. Thus it resembles portability testing as the Android application is tested for running on various phones and it's expected to run with all its functionalities intact.
- **Integration Testing:** Once the important modules have been unit tested, it's followed by integration testing where multiple modules are combined together and are then tested using an Android virtual device.
- **System Testing:** The entire system with all its modules together has been tested to ensure that all the features of the system work properly. This is done by flashing the Embedded C code into the Renesas MCU and by installing the APK file of the android application on a phone and then testing the entire system.
- **Usability Testing:** The user friendliness of the android application and its GUI has been tested by using an AVD and also by installing and running the APK file on the phone.
- **Beta Testing:** We let few our friends test this application so that it could achieve black box testing as they won't know the internal specifications of the project.

8.7 Conclusion

In this chapter the different test cases were discussed along with the screenshots and various defects were identified and classified module wise and also various types of testing such as gray box automated testing (Robotium), Manual testing, Unit testing (Junit), Portability testing, Integration testing, System testing, Usability and Beta testing were demonstrated.

Chapter 8

CONCLUSION AND SCOPE FOR FUTURE WORK

The system developed is a prototype in which the microcontroller needs constant power supply for it to run. Also the prototype has been highly customized for a single patient. This project helps athletes, old people who have knee problems and people who have undergone knee surgery by constantly monitoring the different activities performed by the patient which affects their knee. Also only certain postures such as walking, running, sitting and standing have been used to demonstrate the concept. The two major sections are the hardware code which is implemented using Cube Suite+ in embedded C and the Android application which is implemented using Eclipse Kepler with Android ADT and SDK plugins.

The most important aspect to be considered in the future is to make the entire prototype wearable and to ensure that the fabrication costs involved due to it is low so that the product can be released in the market. This is because the cost of fabrication is quite high as it needs to ensure that the device is small, efficient and can be worn without causing any discomfort. Further the system must extend its variety in identifying the different postures and activities performed by the user. This system can be integrated with smart devices such as watches etc. and also the IOS version of the application is to be developed.

REFERENCES

[1] Masdar, A.; Ibrahim, B.S.K.K; Hanafi, D.; Jamil, M.M.A.; Rahman, K.A.A, "Knee Joint Angle Measurement System Using Gyroscope and Flex-sensors for rehabilitation", Biomedical Engineering International Conference (BMEiCON), 2013 6^{th}, vol., no., pp.1,4, 23-25 Oct. 2013.

[2] Lee, Y.; Kim, J.; Son, M.; Lee, M.; "Implementation of accelerometer sensor module and fall detection monitoring system based on wireless sensor network", Proc. 29th IEEE EMBS Annual International Conference, pp. 2315–2318, Aug 23–26, 2007, Lyon, France.

[3] Jeethendria, A.; Senthilmuguran, S.; "Monitoring Human Joint Movement Using Wearable Computing", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 4, April 2014

[4] Amin Ahmadi; Edmond Mitchell; Chris Richter; FrancoisDestelle; Marc Gowing; Noel E. O'Connor; Kieran Moran; "Toward Automatic Activity Classification and Movement Assessment During a Sports Training Session", IEEE Internet of Things Journal, Vol. 2, No. 1, February 2015.

[5] http://www.renesas.com/products/mpumcu/rl78/index.jsp

[6] Aszkler; Craig.; "Acceleration and Shock and Vibration Sensors, in Sensor Technology Handbook", edited by Jon S. Wilson, 137-159. Burlington: Elsevier, 2005.

[7] http://www.sensorwiki.org/doku.php/sensors/accelerometer

[8] Alexander G. Dean; James M. Conrad; "Creating Fast, Responsive and Energy-Efficient Embedded Systems using the Renesas RL78 Microcontroller", Renesas Books, 2012.

[9] Alexander G. Dean; "Embedded Systems Design, Analysis and Optimization using the Renesas RL78 Microcontroller", Renesas Books, 2012.

APPENDIX

1 Screen Snapshots

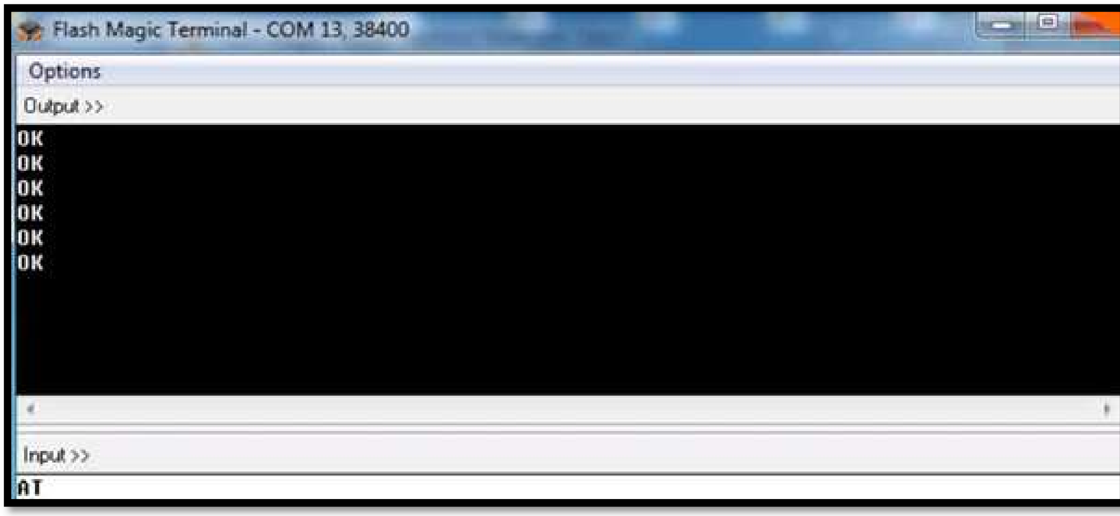

Figure A.1 Testing HC05 Using Flash Magic

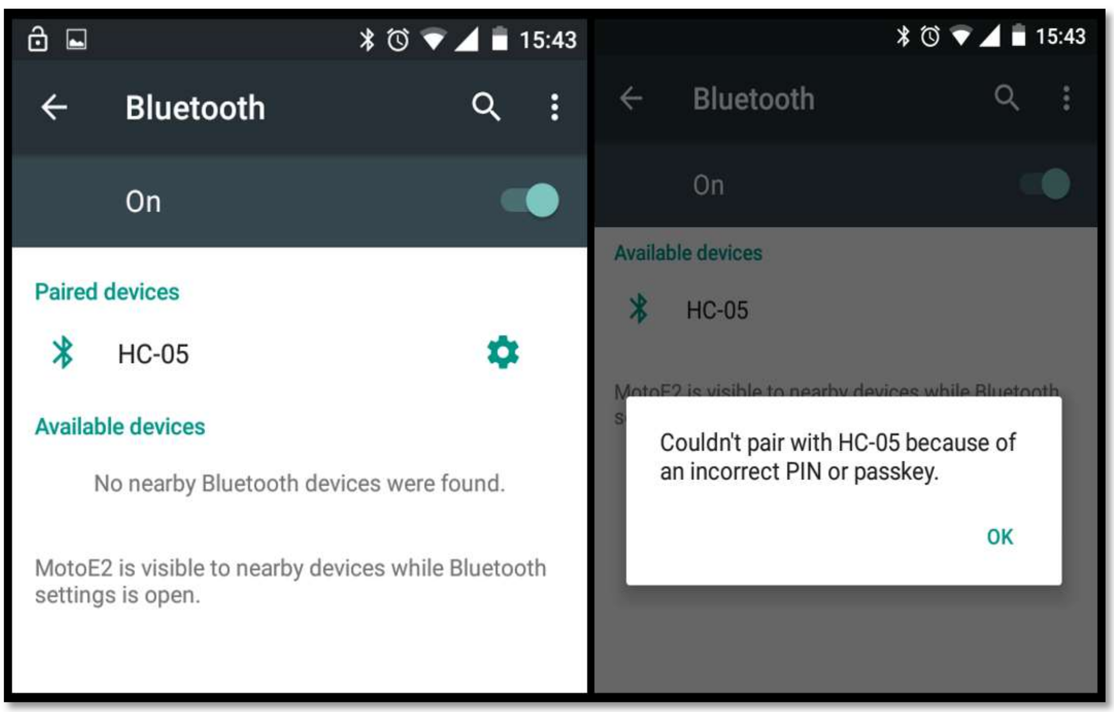

Figure A.2 Pairing HC05 Using APK

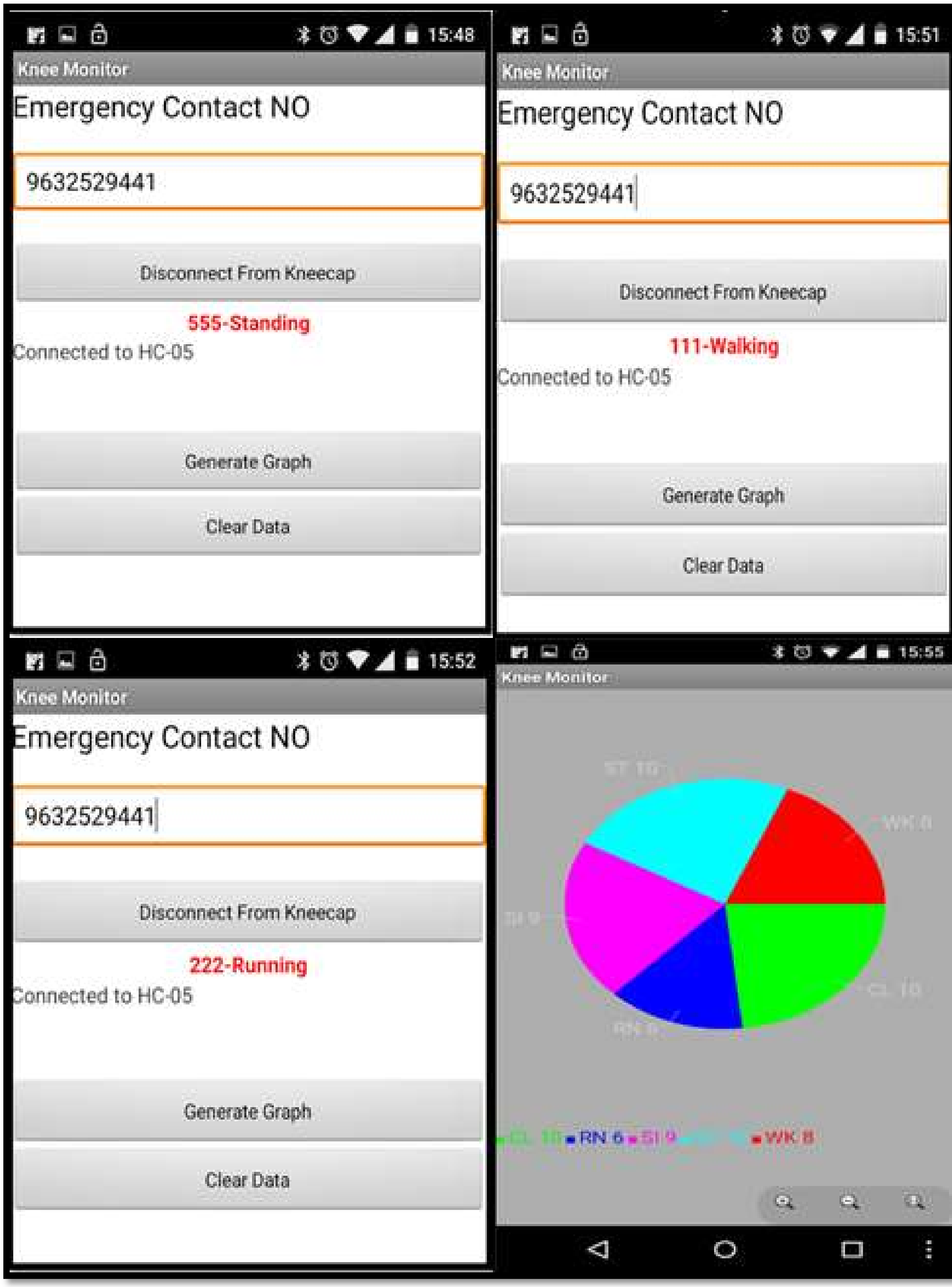

Figure A.3 Data Transfer Testing Using APK

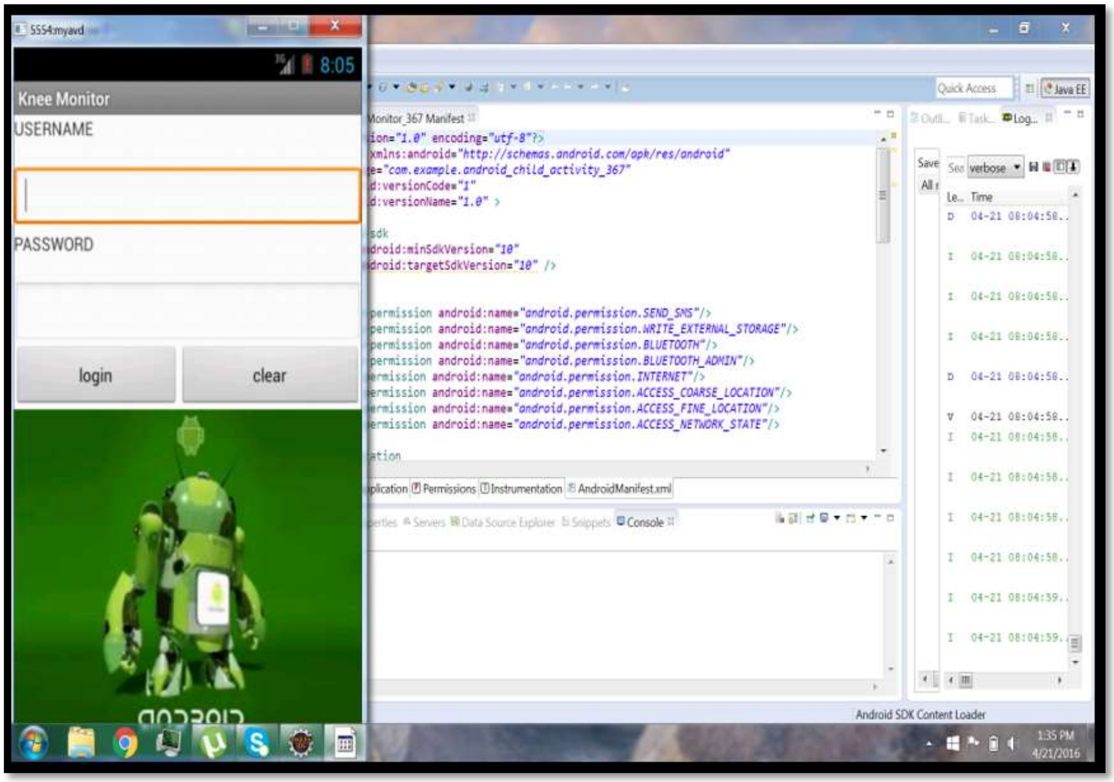

Figure A.4 Testing using AVD

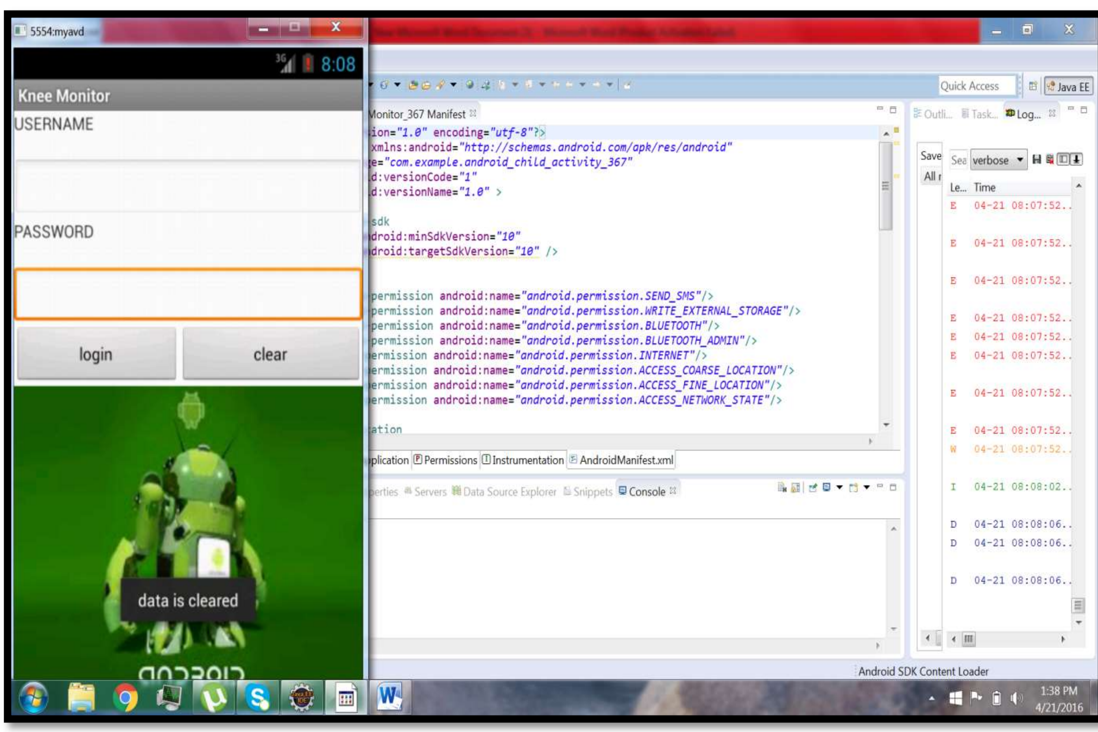

Figure A.5 Testing using AVD (Data Clear)

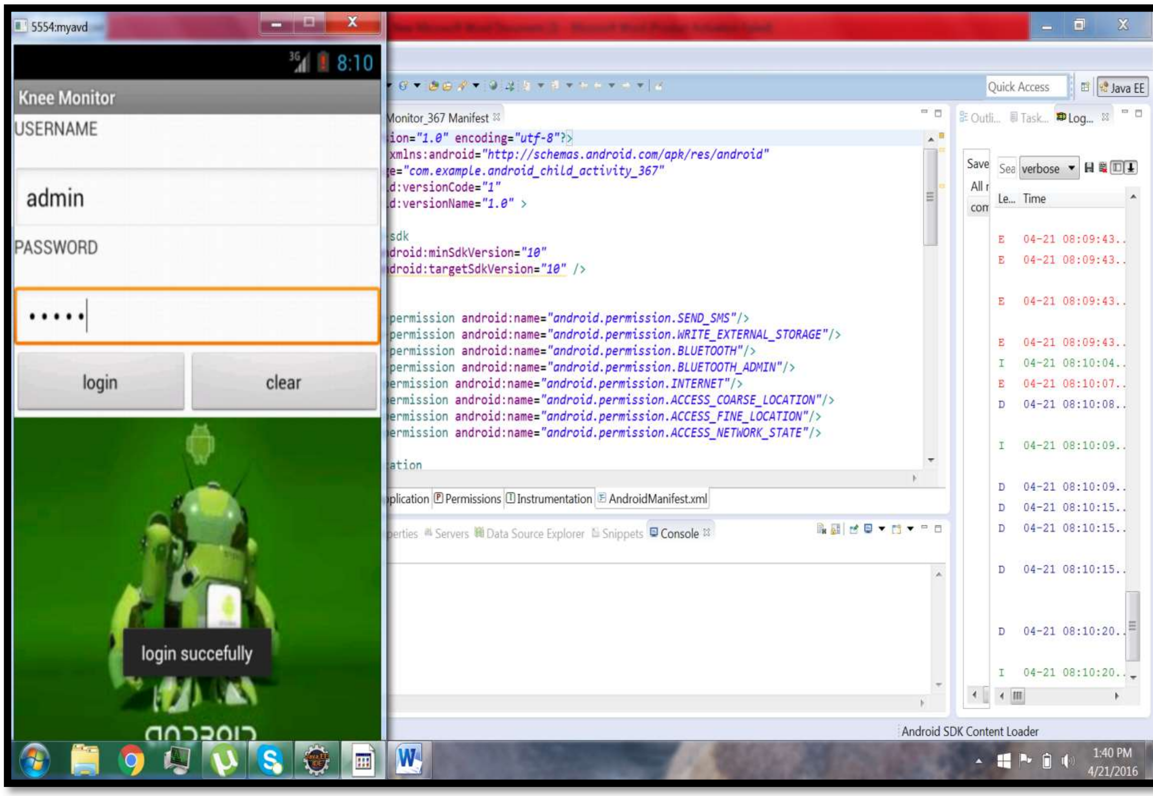

Figure A.6 Testing using AVD (Login)

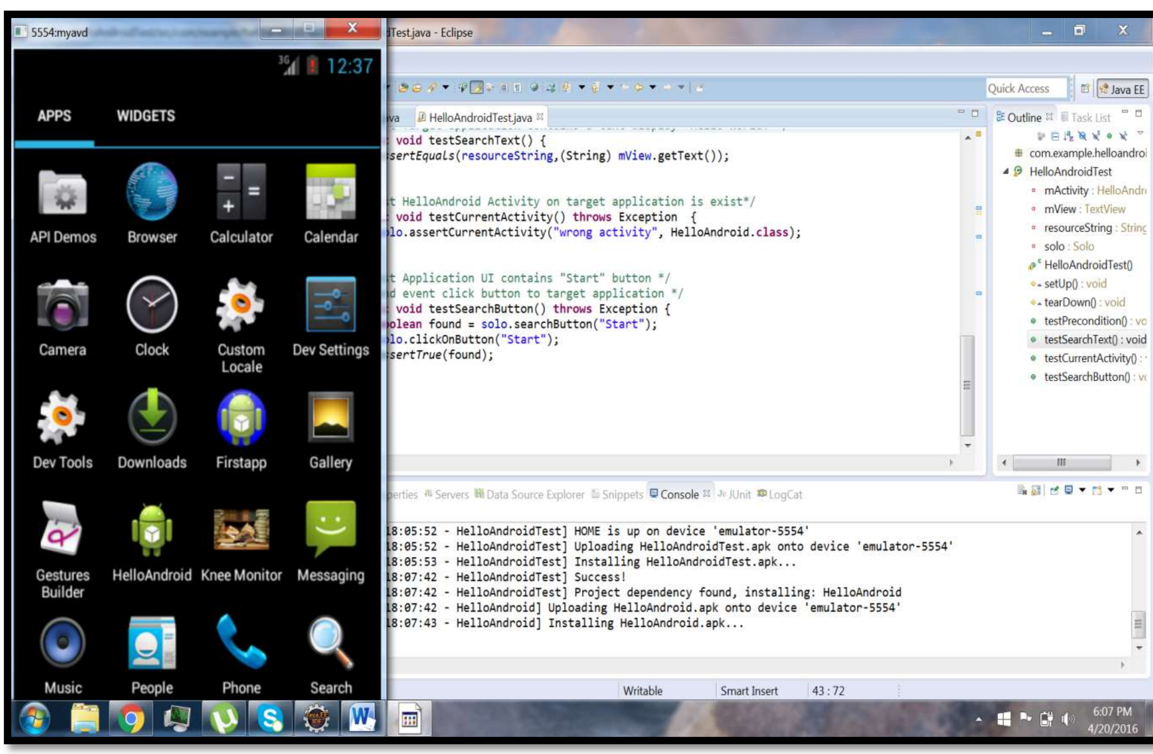

Figure A.7 Junit Testing

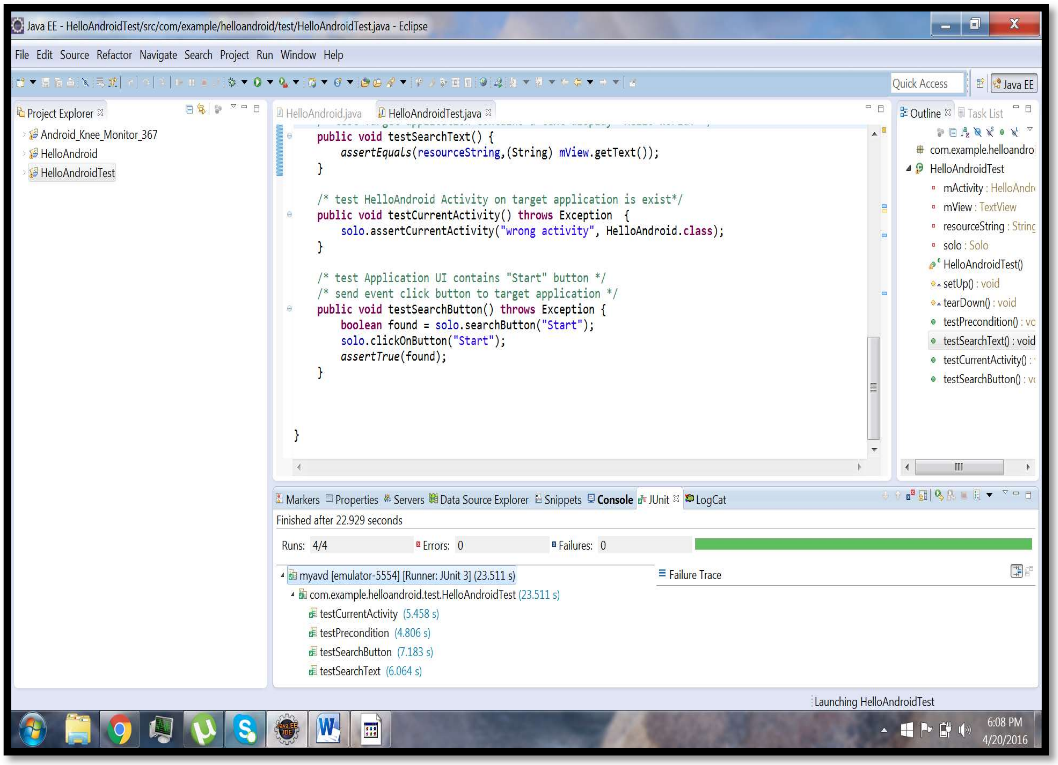

Figure A.8 Junit Testing (Success)

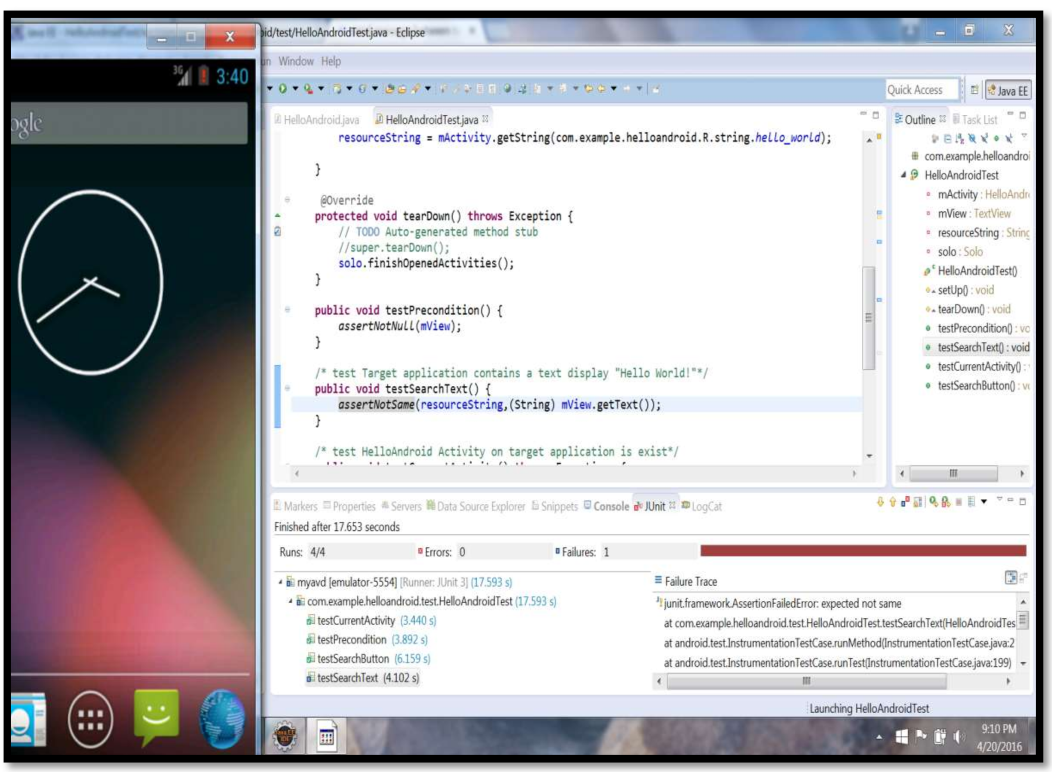

Figure A.9 Junit Testing (Fail)

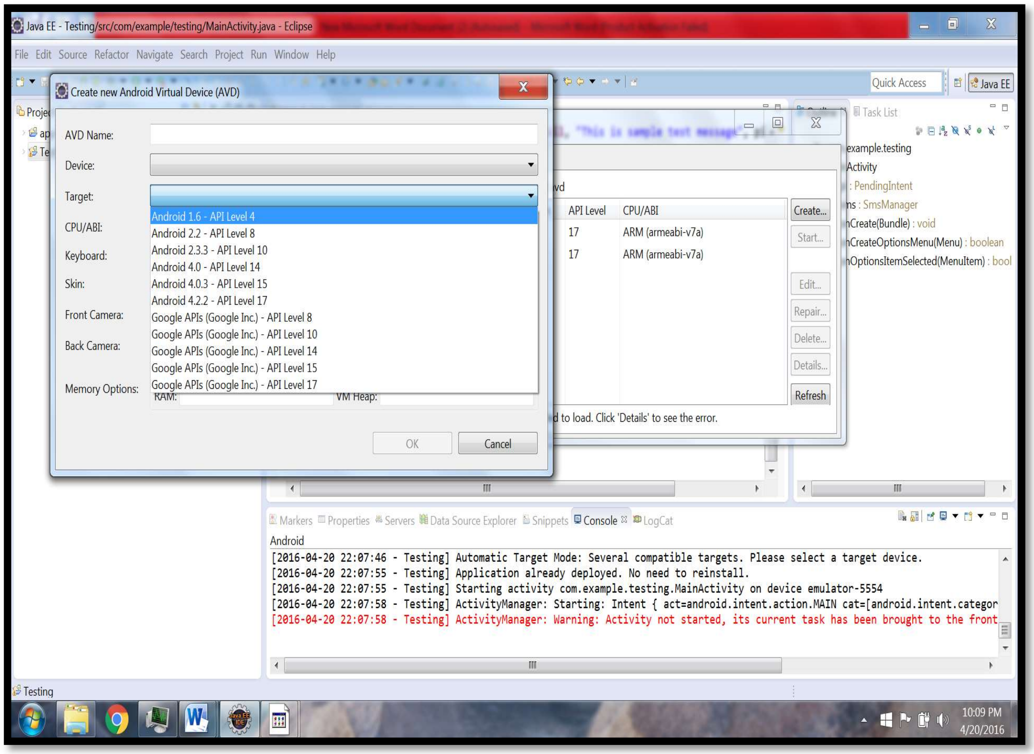

Figure A.10 Different Phone Models (AVD)

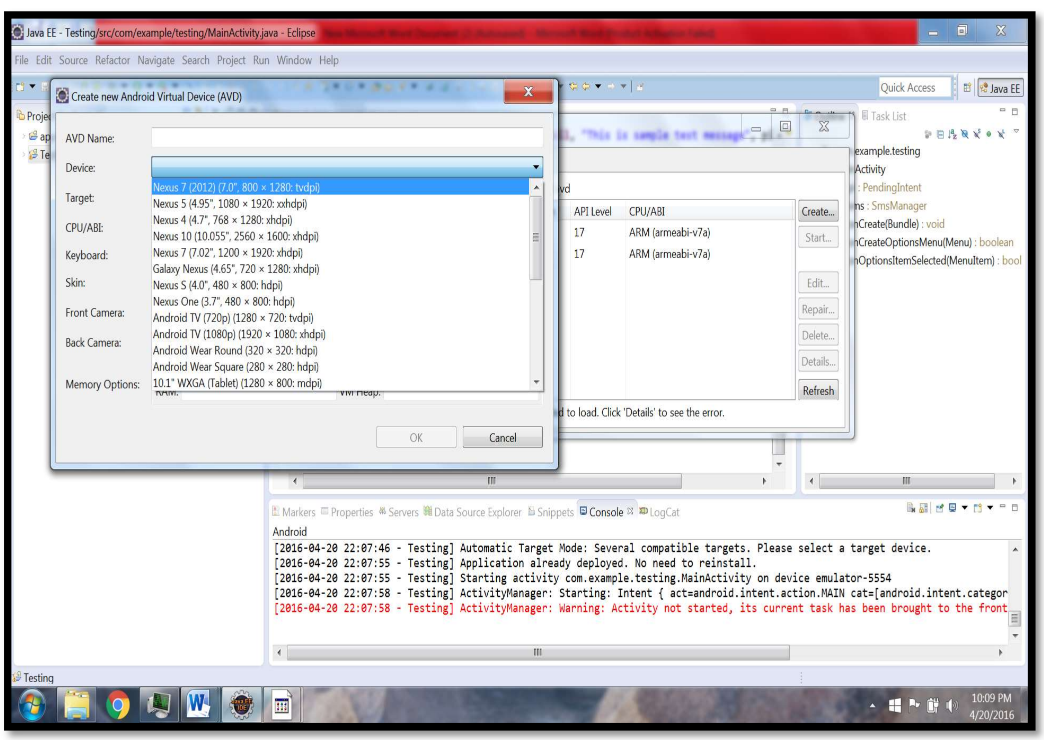

Figure A.11 Different OS versions (AVD)

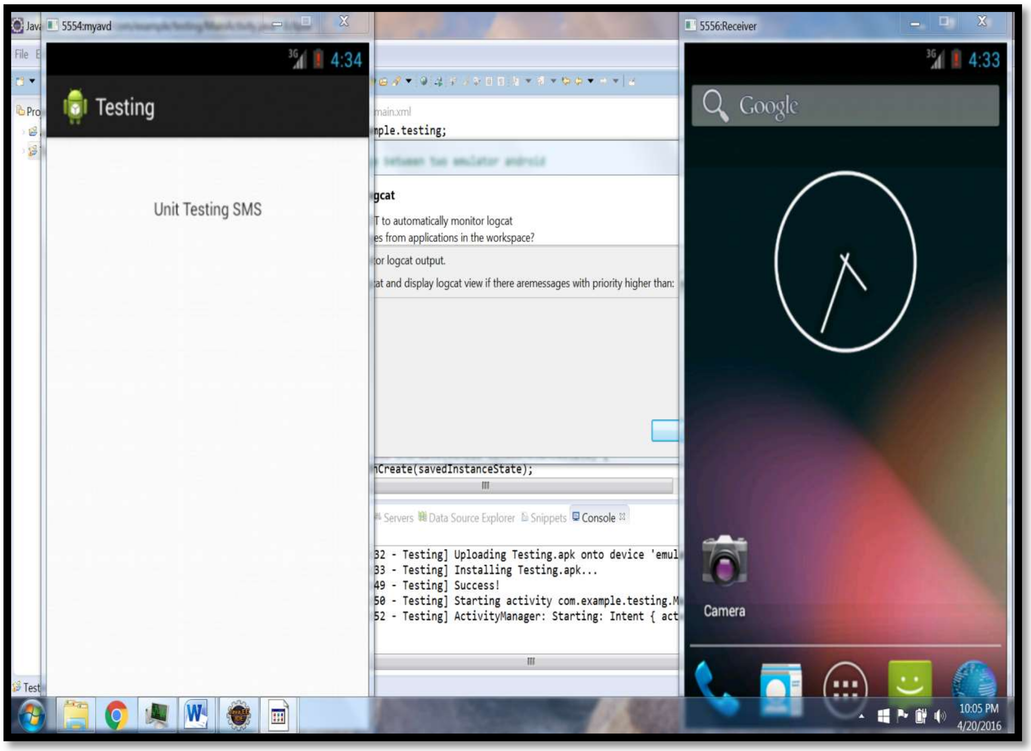

Figure A.12 Testing SMS module (AVD)

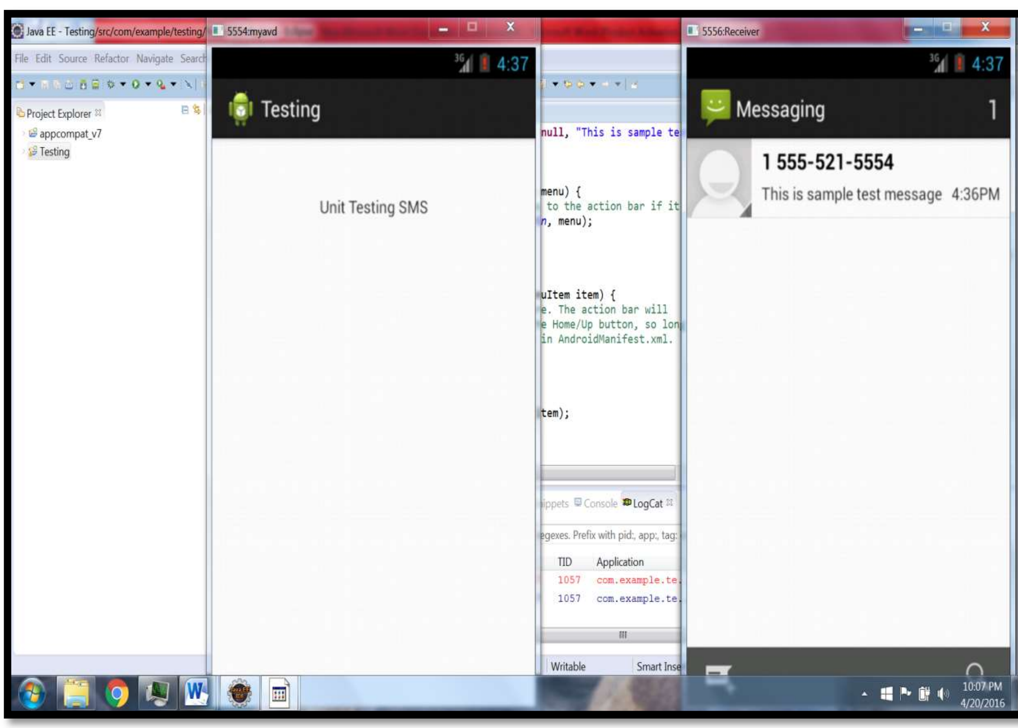

Figure A.13 SMS Received Successfully (AVD)

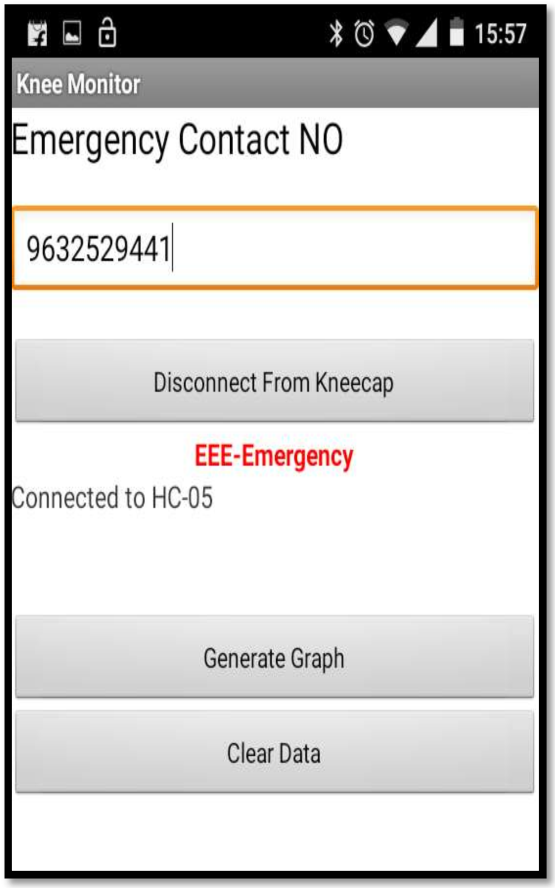

Figure A.14 Emergency Alert Testing (APK)

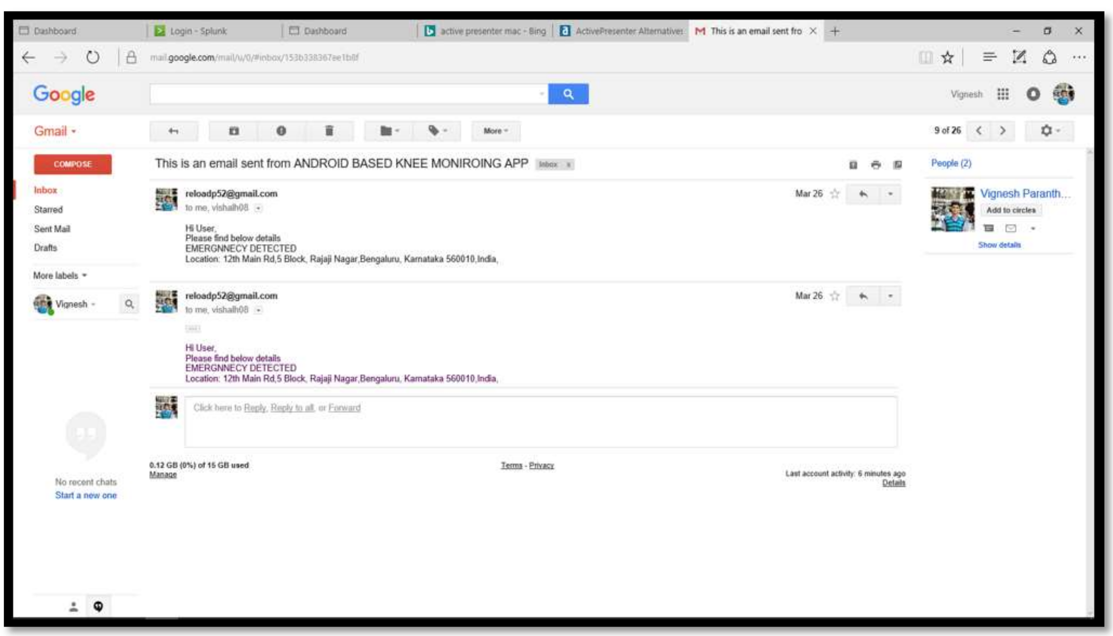

Figure A.15 Email Received Successfully

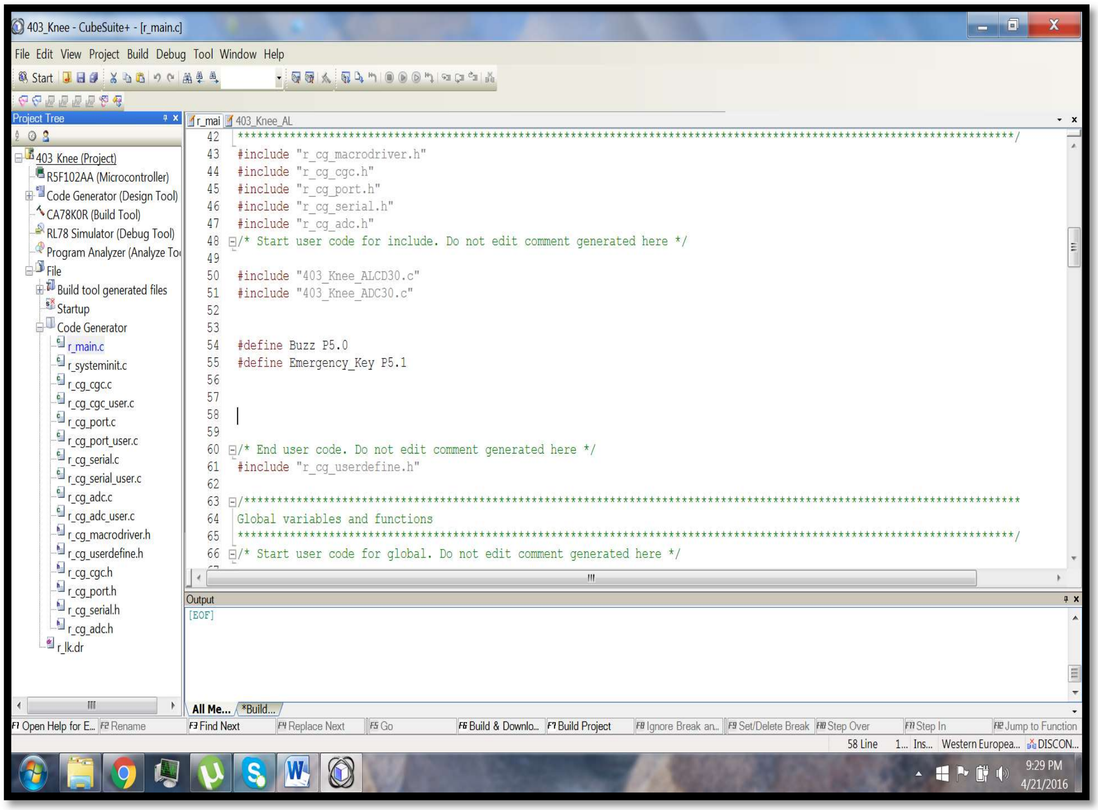

Figure A.16 Hardware Testing (CS+)

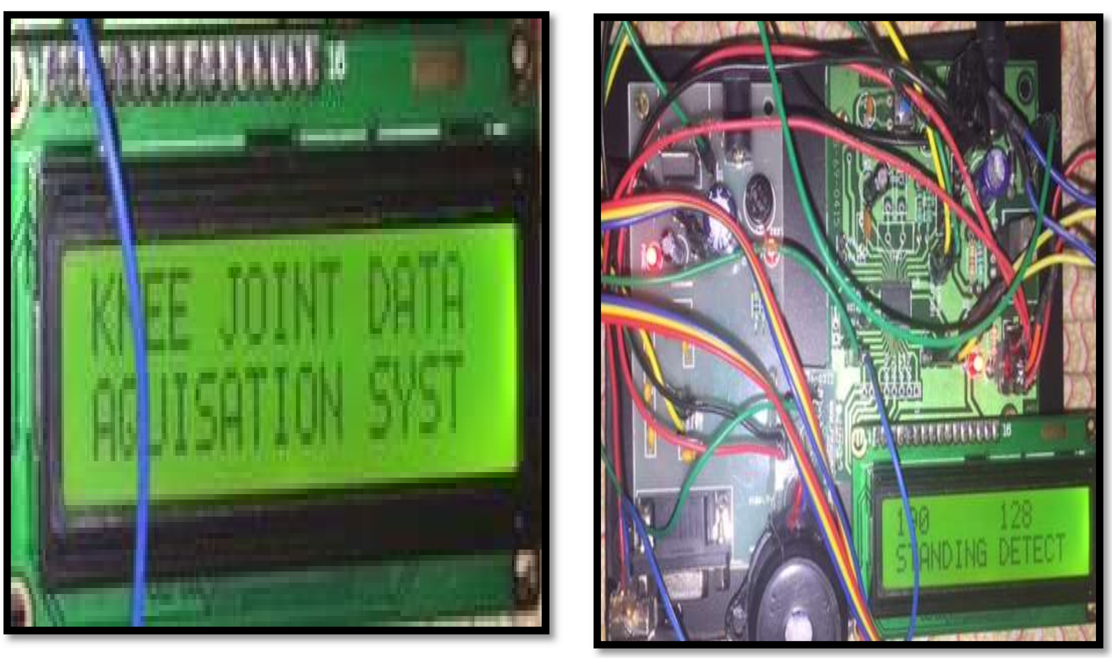

Figure A.17 LCD Testing

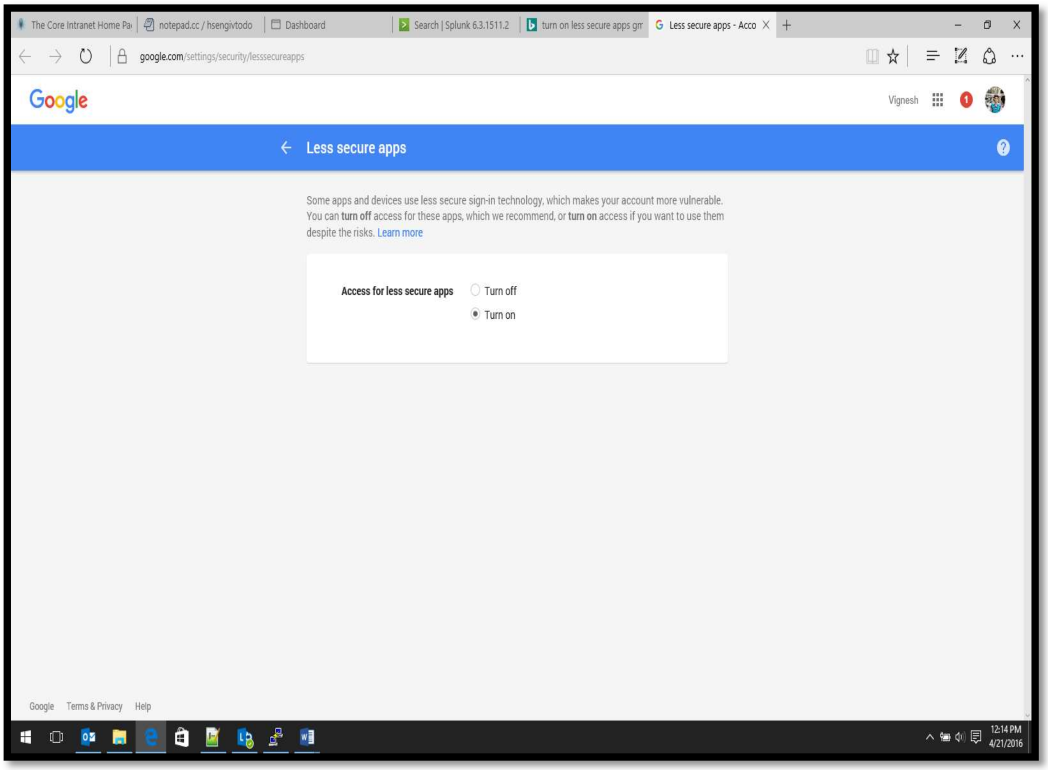

Figure A.18 Resolving Defect in Email Module