**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR nagar, MSRIT post, Bangalore-54

A Design Document on

**Android based Monitoring Human Knee Joint Movement Using Wearable Computing**

Under the guidance of

Mr. M. Mallegowda

Submitted by

Sanjana K. S 1MS12CS098

Vidit Jain 1MS12CS127

Vignesh P 1MS12CS128

Vishal H 1MS12CS132

*In partial fulfillment for the award of the degree of*

# *Bachelor of Engineering in Computer Science & Engineering*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S. RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[www.msrit.edu](http://www.msrit.edu), **May 2015**

**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.

**Modular design**, or "modularity in design", is a design approach that subdivides a system into smaller parts called modules or skids, which can be independently created and then used in different systems. A modular system can be characterized by functional partitioning into discrete scalable, reusable modules, rigorous use of well-defined modular interfaces, and making use of industry standards for interfaces.

Thus this document initially describes the different modules present in the project. It is then followed by the various algorithms that are used to implement the modules. 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. Finally, the document ends with the list of references used in developing this document.

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

1. **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.

1. **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 he/she is 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.

1. **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.
2. **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/doctor needs to monitor the user’s physical activities which affect his/her knee.
3. **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.
4. **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.

**Algorithm Design**

Input: Renesas RL78/G13 microcontroller with all the necessary connections.

Output: Updating Accelerometer Sensor values to Android Application Module, sending Email Alert, voice alert via Bluetooth and triggering a buzzer if emergency button is pressed.

1. R 🡨 Raw data from the two accelerometer sensors.
2. Ra 🡨 P (R) // parsing the raw data to obtain the actual analog voltage values.
3. Rd 🡨 ADC(Ra) //Converting to digital voltage value.
4. LCD(Rd) /\* Displaying appropriate messages and digital voltage value on LCD. \*/
5. App 🡨 Bluetooth 🡨 Rd // Send data to Android Application via Bluetooth.
6. Loop (Ebutton == ON) //Ebutton refers to emergency button

Send Voice Alert.

Send Email Alert using phone’s Internet with location as attachment.

Buzzer 🡨 ON

If (Reset == ON) // If reset button is pressed

Buzzer 🡨 OFF

Break out of loop

1. If (Rd1x == Rd2x)

Pos 🡨 standing /\* If the two sensors are parallel to each other then it indicates standing \*/

Wc++ //Increment count for walking.

1. Else IF (Rd1x < Rd2x/2)

Pos 🡨 sitting /\* If the two sensors are perpendicular to each other then it indicates sitting \*/

Sc++ //Increment count for sitting

1. Else

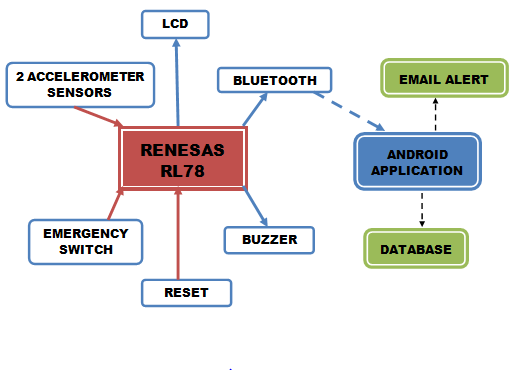
Pos 🡨 Running/walking // Indicates that user is running/walking.

1. Pie Chart 🡨 App /\* Android App generates a pie chart to indicate the different activities such as walking, running, sitting etc. depending on the values of the sensors.\*/
2. Loop back to step 1.

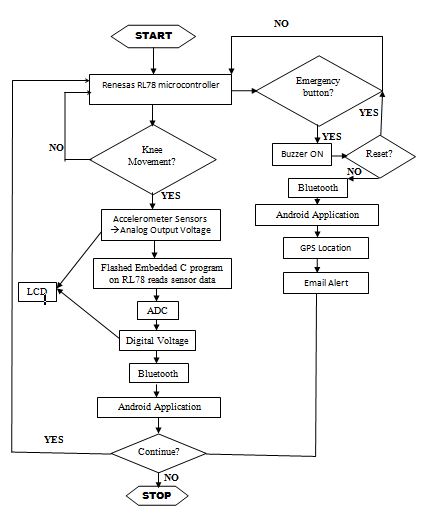
**ARCHITECTURE DESIGN**

A **system architecture** or **systems architecture** is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behaviour) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. The different modules present in this project has been described in the introduction. The system architecture is given in Figure1.



**Figure1. System Architecture**

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 app and Email Alert indicates the usage of Internet connection to send the alert. Figure2. Depicts the flow diagram for the project.****

**Figure2. Flow Diagram**

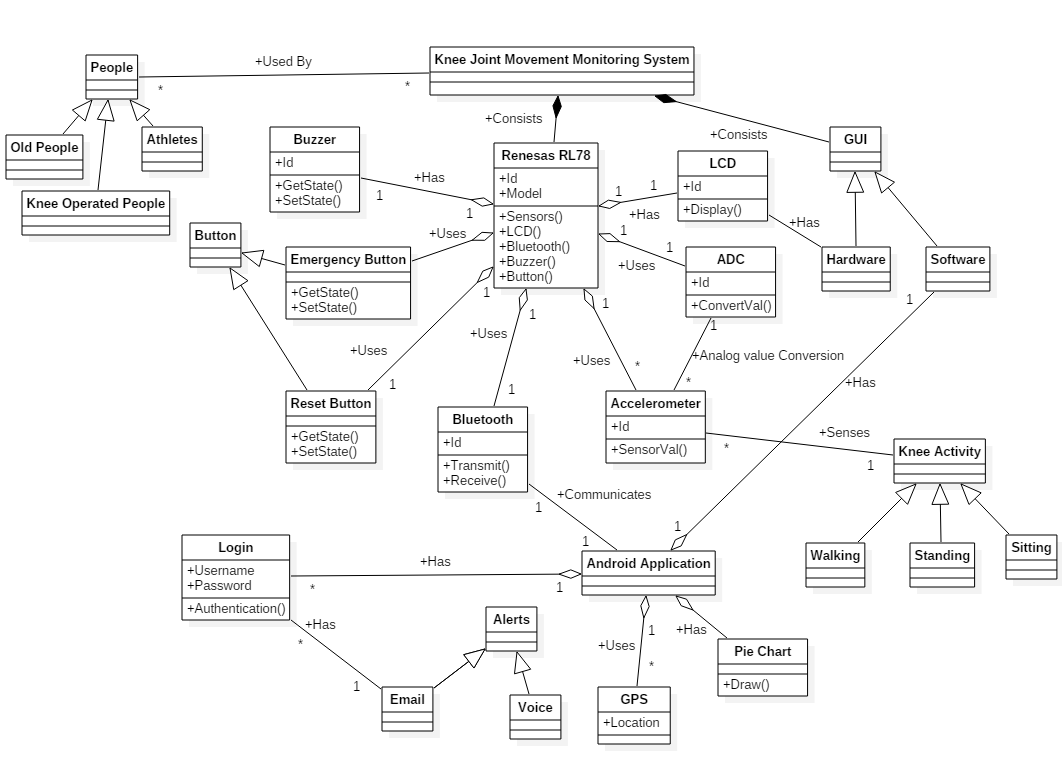
**GRAPHICAL USER INTERFACE**

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 where a pie chart is displayed. 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 a pie chart is generated to indicate the count of each activity performed by the user. Further the Android application also has a button to view the activities performed by the user in the past to provide persistent storage. The GUI has been planned to designed very efficiently so that it’s 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/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.

**CLASS DIAGRAM**

* In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
* The figure in the next slide represents the class diagram for wireless temperature monitoring system which uses association, aggregation, composition and generalization and many more features of the class model.
* The relationship among various classes and their interdependencies are effectively modeled using the class diagram.
* The class diagram for this project is shown in Figure3.

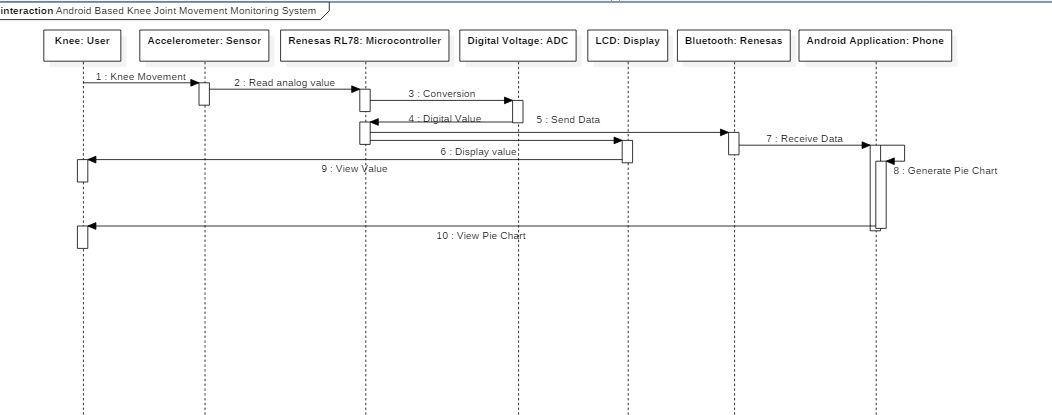


**Figure3. Class Diagram**

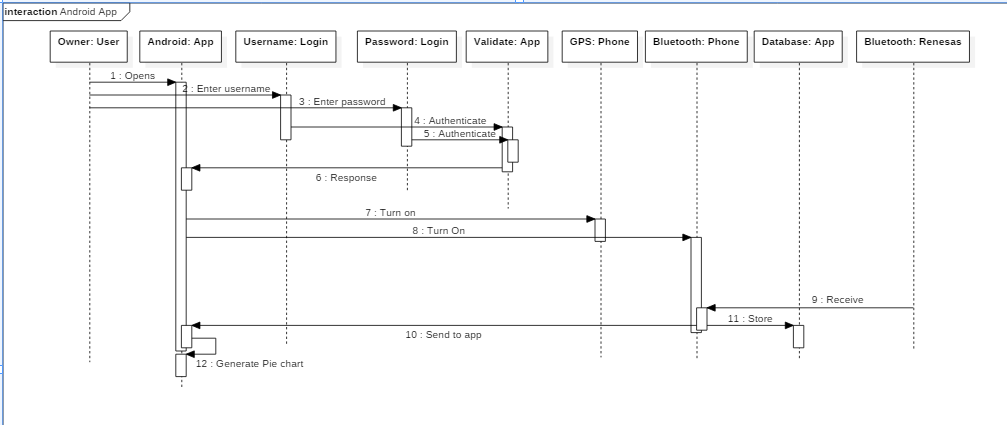
**SEQUENCE DIAGRAM**

**Sequence diagram** is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

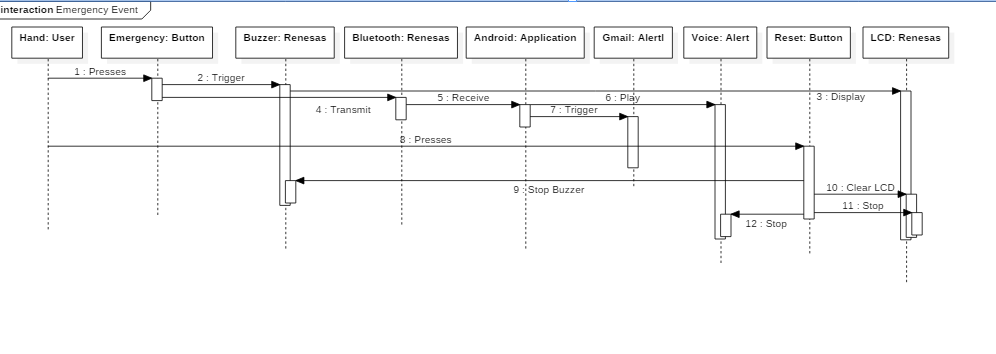
A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. The sequence diagrams for this project are as follows.

****

**Figure4. Sequence Diagram for Knee Joint Monitoring**

****

**Figure5. Sequence Diagram for Android Application**

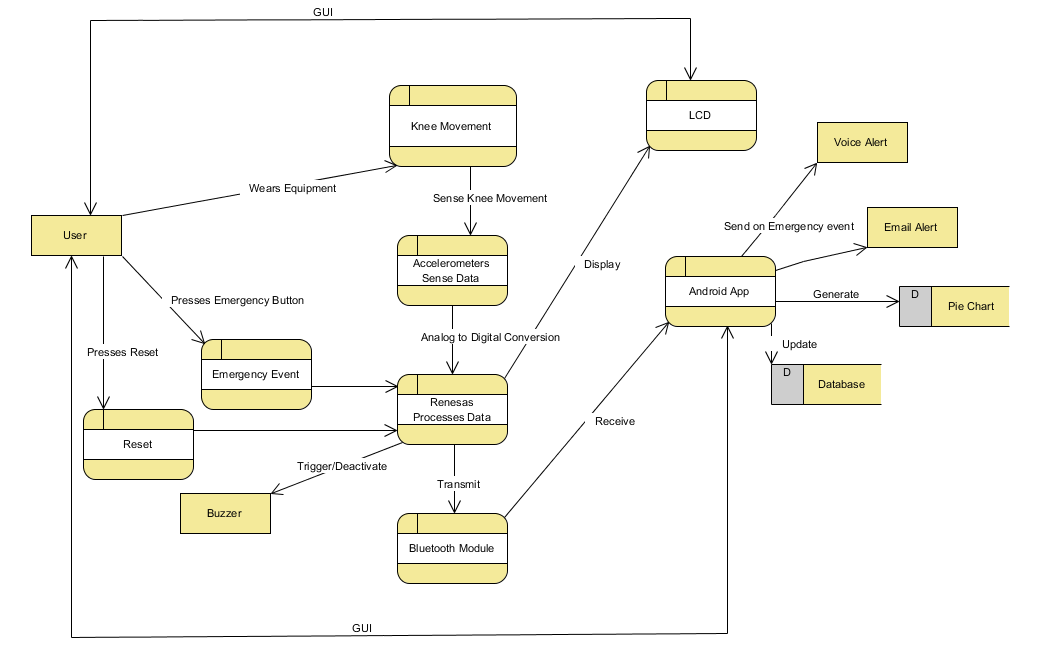
****

**Figure6. Sequence Diagram for Emergency Event**

**DATA FLOW DIAGRAM**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its processaspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

****

**Figure7. Data Flow Diagram**

**REFERENCES**

* <https://en.wikipedia.org/wiki/Accelerometer>
* <http://www.renesas.com/products/mpumcu/rl78/index.jsp>
* <http://documentation.renesas.com/doc/products/tool/doc/r20ut2888eg0100_lcdappv2_lcd.pdf>
* <https://elearning.renesas.com/file.php/1/CoursePDFs/DevCon_2012/Connectivity/2L03I_Harris_RL78BluetoothConnectivity.pdf>
* <http://europepmc.org/articles/PMC2752888>
* <https://www.deepdyve.com/lp/institute-of-electrical-and-electronics-engineers/wearable-joint-kinematic-monitoring-system-using-inertial-and-magnetic-SRrlXSmQyX>
* <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4462047>
* <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2752888/>
* <http://staruml.io/download>
* <https://www.smartdraw.com/uml-diagram/>
* <http://www.uml-diagrams.org/>
* <https://en.wikipedia.org/wiki/Unified_Modeling_Language>
* <http://am.renesas.com/press/news/2015/news20150716.jsp>
* <http://ieeexplore.ieee.org/xpl/abstractKeywords.jsp?reload=true&arnumber=7034178>
* <https://www.youtube.com/watch?v=E-mqG7dIGtQ>
* <https://www.youtube.com/watch?v=ztZsEI6C-mI>
* <https://www.youtube.com/watch?v=kApq-E2mtn0>
* <https://www.youtube.com/watch?v=QCEmzlB6CEs>