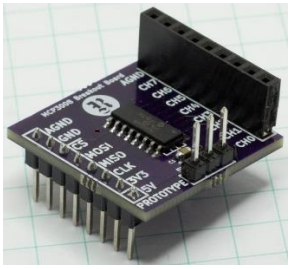
A
Project Report
On
"HEART BEAT MONITOR"



Prepared by

14CE107 Varsh Patel

14CE145 Drashti Vashi

Under the guidance of

Prof. Mohammed Bohara

Submitted to

Charotar University of Science & Technology

Degree of Bachelor of Technology

in Computer Engineering

CE415 -Software Group Project

Of 7th Semester of B.Tech

Submitted at

CHARUSAT

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

U. & P. U PATEL DEPARTMENT OF COMPUTER ENGINEERING

Faculty of Technology & Engineering, CHARUSAT

Chandubhai S. Patel Institute of Technology

At: Changa, Dist.: Anand– 388421

August-December 2017

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "Heart Beat Monitor" is a bona fide work carried out by **Mr. Varsh Patel (14CE107)** and **Ms. Drashti Vashi (14CE145)** under the guidance and supervision of Prof. Mohammed Bohara for the subject **Software Group Project (CE415)** of 7th Semester of Bachelor of Technology in **Computer Engineering** at Faculty of Technology & Engineering (C.S.P.I.T.) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate themselves, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under the supervision of,

Prof. Mohammed Bohara
Assistant Professor
U.& P. U Patel Dept. of Computer Engineering
C.S.P.I.T., CHARUSAT-Changa.

Dr. (Prof.)Amit Ganatra
Dean,
Faculty of Technology& Engineering
Head, U.& P. U Patel Department of Computer Engineering
C.S.P.I.T., CHARUSAT-Changa, Gujarat.

Chandubhai S Patel Institute of Technology (C.S.P.I.T.)

Faculty of Technology & Engineering, CHARUSAT

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

Abstract

"Heart Rate Monitor" is a personal monitoring device that senses the pulse rate of the users. It allows real time storage of data over the cloud. So the doctors can access the data from any corner of the world with the help of the web portal developed to analyze it. It is of prime benefit for the patients suffering from high blood pressure who need to regularly monitor their heart-rate. This project uses the RASPBERRY PI Microcontroller.

Acknowledgement

We take this opportunity to present our votes of thanks to all those guidepost who really acted as lightening pillars to enlighten our way throughout this project that has led to successful and satisfactory completion of this study.

We are really grateful to our **Dr. (Prof.) Amit Ganatra**, HOD - U & P U Patel Department of Computer Engineering, Dean - Faculty of Technology & Engineering for providing us with an opportunity to undertake this project in this university and providing us with all the facilities. We are highly thankful to **Prof. Mohammed Bohara** for his active support, valuable time and advice during the study and in completing the assignment of preparing the said project within the time stipulated.

Lastly, We are thankful to all those, particularly the various friends, who have been instrumental in creating proper, healthy and conductive environment and including new and fresh innovative ideas for us during the project, their help, it would have been extremely difficult for us to prepare the project in a time bound framework.

Table of contents

		Page No.
	Abstract	i
	Acknowledgement	ii
1.	**Introduction**	1
	1.1 Purpose	1
	1.2 Scope	1
	1.3 Definitions, Acronyms & Abbreviations	1
	1.4 References	1
	1.5 Overview	1
2.	**Overall Description**	2
	2.1 Product Perspective	2
	2.1.1 Hardware Interfaces	2
	2.1.2 Software Interfaces	2
	2.1.3 Operations	2
	2.2 Product Functions	2
	2.3 User Characteristics	2
	2.4 Constraints	2
3.	**System Analysis**	3
	3.1 Requirements Of New System	3
	3.1.1 Functional Requirements	3
	3.1.2 Non-Functional Requirements	4
	3.2 Feasibility Study	4
	3.3 Functions of System	5
	3.4 Selection of Hardware and Software and Justification	6
4.	**Project Estimation**	7
	4.1 Estimation Technique	7
	4.1.1 LOC (Line Of Codes) And FP (Function Point)	7
	4.1.2. COCOMO (COst COnstructive MOdel)	9
5.	**Project Scheduling**	10
	5.1 Work Breakdown Structure	10
	5.2 Task Network Representation	11
	5.3 Gantt Chart	11
	5.3 Pert Chart	12
6.	**Project Resources**	13
	6.1 People	13
	6.1.1 Team Structure	13
	6.1.2 Management Report	13
	6.2 Hardware And Software	13
	6.2.1 Hardware Requirement	13
	6.2.2 Software Requirement	13

7.	**Project Tracking**	14
	7.1 Tracking Gantt	14
8.	**System Design**	15
	8.1 System Application Design	15
	8.1.1 Method Pseudo Code	15
9.	**Implementation Planning**	17
	9.1 Implementation Environment	17
	9.2 Project/Module Specification	17
	9.3 Coding Standards	17
10.	**Testing**	18
	10.1 Testing Strategy And Test Cases	18
11.	**Limitation & Future Enhancements**	19
	11.1 Limitations	19
	11.2 Future Enhancements	19
12.	**Conclusion & Discussion**	20
	12.1 Self Analysis and Project Viabilities	20
	12.2 Problem Encountered and Possible Solutions	20
	12.3 Summary of Project Work	20

CHAPTER 1 INTRODUCTION:

1.1. PURPOSE:

The main purpose of the specification is to guide the developer about the application and the development structure, requirements and the input and output feature.

1.2. SCOPE:

This project not only senses the pulse rate of the user but also stores it over the cloud to help doctors access it with the help of developed web portal for later analysis.

BENEFITS:

- User-friendly.
- Get time sensitive data
- Efficient.

OBJECTIVES:

- Eliminate wastage of time in taking attendance
- Less involvement of human

1.3. DEFINITIONS, ACRONYMS AND ABBREVIATION:

- RP – Raspberry Pi 2B+
- SSMS – SQL Server Management Studio
- BPM – Beats Per Minute

1.4. REFERENCES:

The books and websites used for the pre development of this project include:-

- https://www.raspberrypi.org/
- http://www.raspberrypitutorials.yolasite.com/
- https://tutorials-raspberrypi.com/raspberry-pi-heartbeat-pulse-measuring/

1.5. OVERVIEW:

Section I gives a brief introduction about the document and the objectives of the software. The overall description such as system interface, software interface, hardware interface, communication interface are discussed in section II and the software product featured section III.

CHAPTER 2 OVERALL DESCRIPTION:

2.1. PRODUCT PERSPECTIVE:

The system/product is fully automated. It is only to be kept at the wrist of the end-user. All operations are performed by microcontroller on its own.

2.1.1: Hardware interfaces:

- Raspberry Pi Microcontroller
- MCP3008 ADC
- Breadboard
- Jumper Wires

2.1.2: Software Interfaces:

- Raspbian OS
- Visual Studio

2.1.2: Operations:

The list of operations are given as below:-

- Measuring the pulse rate of the user.
- Automatically store it over the cloud.
- Access the data with the use of web portal.

2.2. PRODUCT FUNCTIONS:

The major functions include:

- Sense the pulse-rate of the end user.
- Store it over the cloud.
- Easy retrieval of stored data whenever needed.

2.3 USER CHARACTERISTICS:

This product only requires to be put on the user's wrist for sensing the pulse-rate. The retrieval of data can be done easily from the web portal developed.

2.4. CONSTRAINTS

- Ultrasonic sensor may fail anytime.
- Raspberry Pi may fail anytime.

CHAPTER 3 SYSTEM ANALYSIS:

3.1 REQUIREMENTS OF NEW SYSTEM

3.1.1 Functional Requirements

R.1 Heartrate Sensing -

Description: User's heartbeat will be sensed and computed in BPM by the heartrate sensor.
Input: On physical touch.
Output: Read heartrate and compute the following step.

R.2 Store-

Description: Store the BPM reading in real time.
Input: In accordance with previous event.
Output: Store the BPM readings in a text file.

R.3 Transferring text file over cloud

Description: Transfers the file that was created by R.2 over cloud.
Input: In accordance with previous event.
Output: File successfully stored over cloud.

R.4 Retrieval of text file from cloud

Description: User/Doctor can easily retrieve the text file that has been stored over the cloud, anytime, anywhere.
Input: On click event
Output: User/Doctor can view the text file.

3.1.2 Non Functional Requirements

- Accuracy: System should display accurate heart rate of the users.
- Consistency: Outcome of system should remain consistent after years also.
- Speed: System should process the data as fast as possible for sake of time integrity constrain.
- Reliable- There is less probability of failures.

3.2 FEASIBILITY STUDY

Questionnaire with intern nurses in hospitals:

Q: How frequent you find patients consulting for heart rate monitoring?
A: 55% of the patients that visit our hospital needs to be check and examined for heart rate first before any treatment can be provided.

Q: Is heart rate monitoring process by Sphygmomanometer time consuming?
A: It almost takes up to 4 to 6 minutes per patient.

Q: Would u expect a system who perform the same task in an efficient way?
A: Yes, unless and until it is less time consuming and accurate.

Q: What do you think about an electronic heart rate monitoring device which can read and store BPM of a patient in real time?
A: Yes, it would be of a great help in reducing the time consumption and accuracy provided by electronic device in comparison to present available traditional process.

3.3 FUNCTIONS OF SYSTEM

- Display the data fetched by the sensor.
- Store the data of a particular event.
- Continuous monitoring the data.

3.4 SELECTION OF HARDWARE AND SOFTWARE AND JUSTIFICATION

We have selected MCP3008 ADC for the interaction between Raspberry Pi and the blood rate sensor. Raspberry Pi is powered up by 2.4V adapter via micro USB cable. HDMI cable for connecting the RP to the monitor or any other output source.

CHAPTER 4 PROJECT ESTIMATION:

4.1 ESTIMATION TECHNIQUE USED:

The project size is a measure of the problem complexity in terms of the effort and time required to develop the product. Currently, two metrics are popularly being used to estimate size: Line of code (LOC) and function point (FP).

4.2 EFFORT RESOURCE, PROJECT DURATION ESTIMATION:

Parameters	**Count**		**Simple**	**Average**	**Complex**		**Total**
No. of user Input	1	X	3	**4**	6	=	4
No. of user Output	1	X	4	**5**	7	=	5
No. of Inquires	0	X	3	**4**	6	=	0
No. of Files	2	X	7	**5**	15	=	10
external Interface	2	X	5	**7**	10	=	14

Complexity Weight Factor:

Sr. No.	Factors	Weights
1.	Does the system require reliable backup and recovery?	**4**
2.	Are data communication required?	**3**
3.	Are there distributed processing functions?	**3**
4.	Is performance critical?	**3**
5.	Will the system run in an existing, heavily utilized operational environment?	**3**
6.	Does the system require online data entry?	**4**
7.	Does the on-line data entry require the input transactions to be built over multiple screens or operation?	**2**
8.	Are the master file updated on-line	**2**
9.	Are the inputs, outputs, files, or inquiries complex?	**2**
10.	Is the internal processing complex?	**2**
11.	Is the code designed to be reusable?	**4**
12.	Are conversion and installation included in the design?	**3**
13.	Is the system designed for multiple installations in differentorganizations?	**4**
14.	Is the application designed to facilitate change and ease of use by the user?	**4**

Weight	Degree Of Influence
0	No Influence
1	Incidental
2	Moderate
3	Average
4	Significant
5	Essential

FP Count:

FP = count total*[0.65+0.01 *∑(Di)]

FP = 20* [0.65+0.01*53]

FP = 46.01

Function Point is: 46.01

Line of code (LOC) = FP*30 = 46.01*30 = 1380.3

KLOC = 1.3803

Software Project Type

Type	**a_b**	**b_b**	**c_b**	**d_b**
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	**3.6**	**1.20**	**2.5**	**0.32**

Effort = a_b*(KLOC) ^b_b

= 3.6*(1.3803) ^1.20

= **5.29 PM**

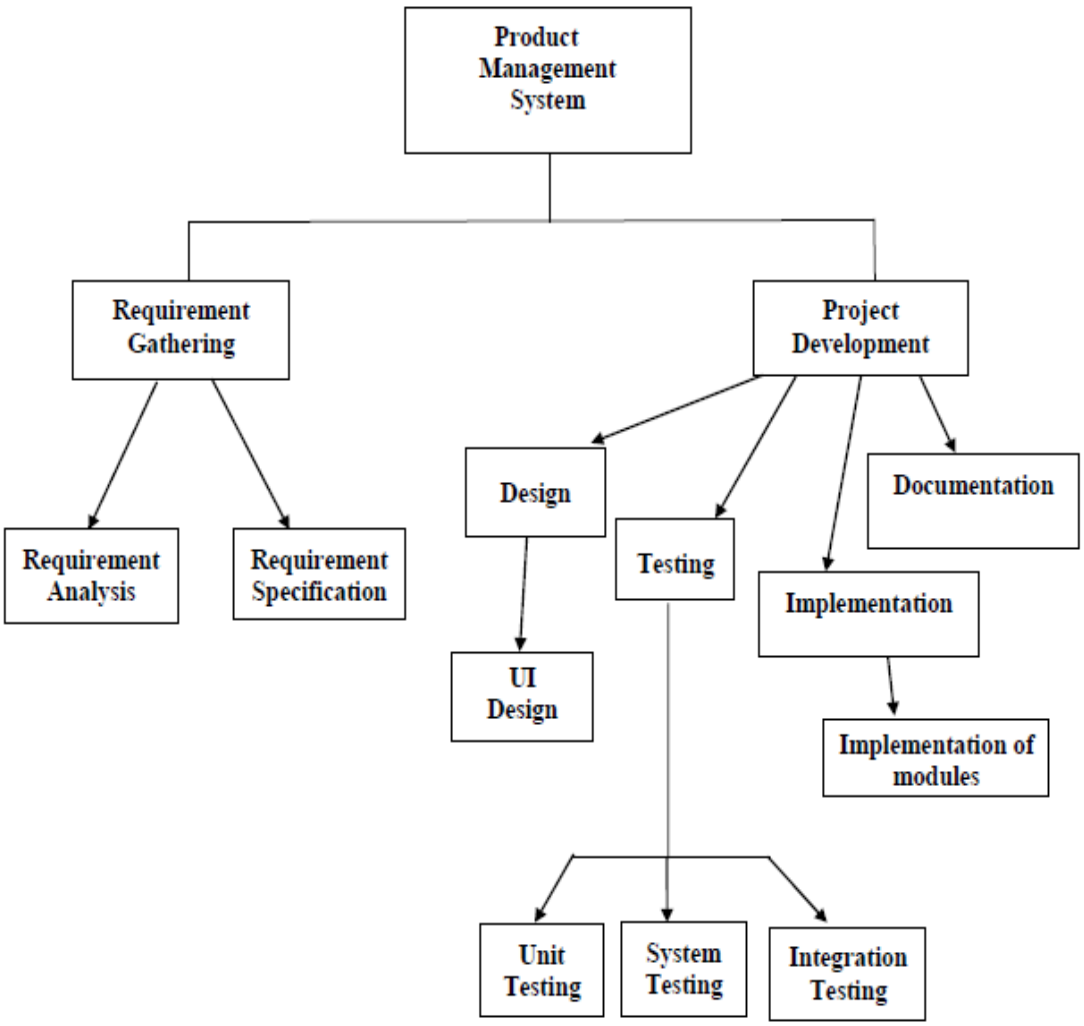
Tdev = c_b * (Effort) ^d_b

= 2.5*(5.29) ^0.32

= **4.26 Months**

CHAPTER 5 SCHEDULE IN WHICH

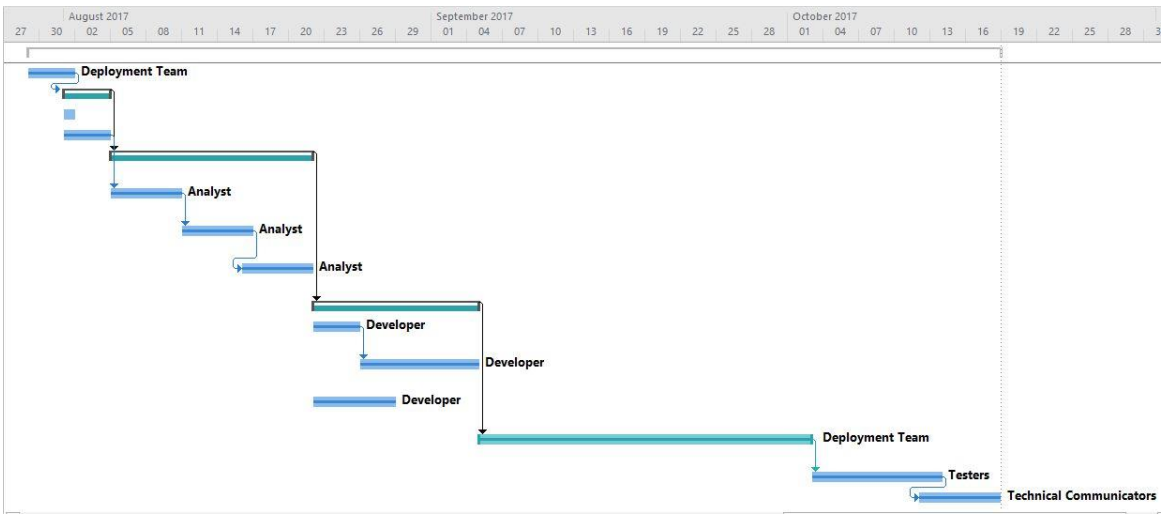
5.1 BREAKDOWN STRUCTURE



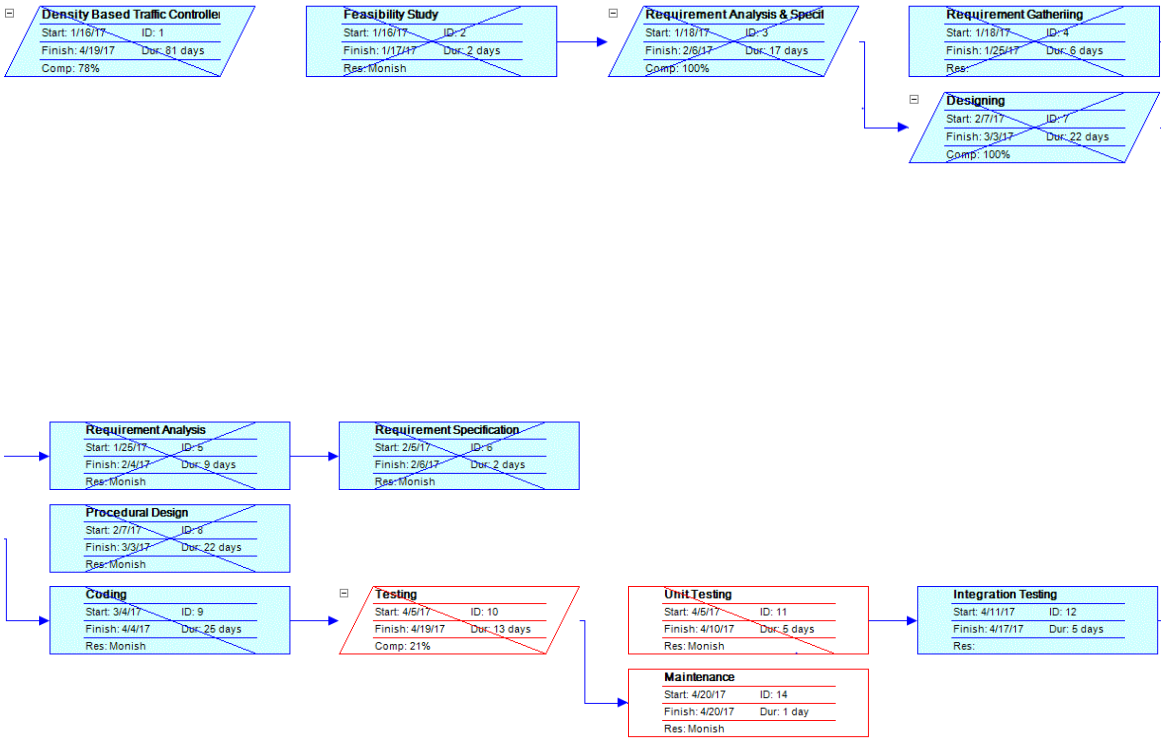
5.2TASK NETWORK REPRESENTATION:-

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
0		**Heart Beat Monitor**	**58 days**	**Sat 29-07-17**	**Wed 18-10-17**		
1		**Feasibility Study**	2 days	Sat 29-07-17	Tue 01-08-17		**Deployment Team**
2		**Requirement Gathering**	**4 days**	**Tue 01-08-17**	**Fri 04-08-17**	**1**	**Analyst**
3			1 day	Tue 01-08-17	Tue 01-08-17		
4		Survey	4 days	Tue 01-08-17	Fri 04-08-17		
5		**Planning and Analysis**	**12 days**	**Sat 05-08-17**	**Mon 21-08-17**	**2**	**Deployment Team**
6		Hrs. Distribution	4 days	Sat 05-08-17	Thu 10-08-17	4	Analyst
7		Work Distribution	4 days	Fri 11-08-17	Wed 16-08-17	6	Analyst
8		Resource Allocation	4 days	Wed 16-08-17	Mon 21-08-17	7	Analyst
9		**Designing**	**10 days**	**Tue 22-08-17**	**Mon 04-09-17**	**5**	**Deployment Team**
10		Models	4 days	Tue 22-08-17	Fri 25-08-17		Developer
11		Web Services	6 days	Sat 26-08-17	Mon 04-09-17	10	Developer
12		SQL Server Management	5 days	Tue 22-08-17	Mon 28-08-17		Developer
13		**Implementation and Coding**	20 days	Tue 05-09-17	Mon 02-10-17	9	**Deployment Team**
14		**Testing**	9 days	Tue 03-10-17	Fri 13-10-17	13	**Testers**
15		**Documentation**	5 days	Thu 12-10-17	Wed 18-10-17	14	**Technical Communicators**

5.3 GANTT CHART REPRESENTATION:-



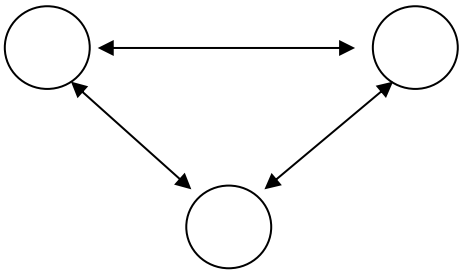
5.4 PERT CHART REPRESENTATION:-



CHAPTER 6: PROJECT RESOURCES

6.1 PEOPLE

- The team structure is Democratic structure. The team members are able to interact with each other and share their ideas about the development of the project.
- Three members are involved in the project.

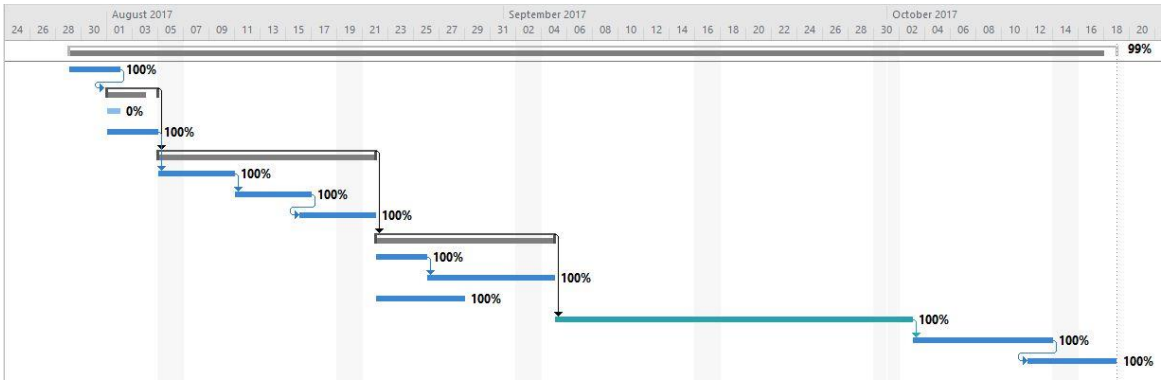


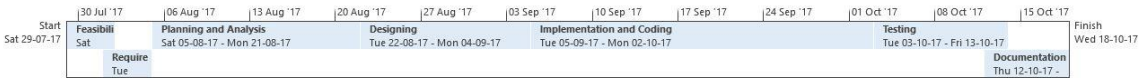
Members	**Responsibility**
Varsh Patel (14CE107)	Feasibility Study Requirement Analysis Designing Coding Cloud Management Backend
Drashti Vashi (14CE145)	Requirement Gathering Coding Testing Cloud Management .NET Documentation

6.2 HARDWARE AND SOFTWARE

Hardware	**Software**
- Raspberry-Pi - MCP3008 ADC - Bread Board - Jumper Wires	-Development tool: Raspbian OS, ASP.NET

CHAPTER 7: PROJECT TRACKING AND CONTROL PLAN





CHAPTER 8: SYSTEM DESIGN

8.1 SYSTEM APPLICATION DESIGN

8.1.1 Method Pseudo code

Coding module of project:

```
import time
import Adafruit_ADS1x15

if __name__ == '__main__':
adc = Adafruit_ADS1x15.ADS1015()
# initialization
GAIN = 2/3
curState = 0
thresh = 525 # mid point in the waveform
P = 512
T = 512
stateChanged = 0
sampleCounter = 0
lastBeatTime = 0
firstBeat = True
secondBeat = False
Pulse = False
IBI = 600
rate = [0]*10
amp = 100

lastTime = int(time.time()*1000)
while True:
Signal = adc.read_adc(0, gain=GAIN)
curTime = int(time.time()*1000)
sampleCounter += curTime - lastTime;
lastTime = curTime
N = sampleCounter - lastBeatTime;
```

```
if Signal < thresh and N > (IBI/5.0)*3.0 :
if Signal < T :
T = Signal;
if Signal > thresh and Signal > P:
P = Signal;
if N > 250 :
if (Signal > thresh) and (Pulse == False) and (N >
(IBI/5.0)*3.0) :
Pulse = True;
IBI = sampleCounter - lastBeatTime;

lastBeatTime = sampleCounter

if Signal < thresh and Pulse == True :
Pulse = False;
amp = P - T;
thresh = amp/2 + T;
P = thresh;
T = thresh;

if N > 2500 :
thresh = 512;
P = 512;
T = 512;
lastBeatTime = sampleCounter;
firstBeat = True;
secondBeat = False;
print "no beats found"

time.sleep(0.005)
```

CHAPTER 9: IMPLEMENTATION PLANNING

9.1 IMPLEMENTATION ENVIRONMENT (Single vs Multiuser, GUI vs Non GUI)

The implementation environment is multiuser as involvement of data over cloud is present in our project. It has user friendly interactive GUI for doctors to keep track of the patients' BPM reading.

9.2 PROGRAM /MODULES SPECIFICATION

This project is divided into modules which has different functionalities and altogether they process the data that is collected and stored in a text file from the sensor and stores the same onto cloud.

9.3 CODING STANDARD

- Class names have first letter in capital case.
  Example: Login
- Variable names are in lower case.
  Example: wpm
- Method names are in mixed case with first letter in lower case.
  Example: void check();
- Name of packages are in lower case.
  Example: package cashfree1test;
- Textarea and Textpane names are of the form name_textarea.
  Example: type_textarea
- Buttons names are of the form name_button. For example pause_button.
- Buttons used to make the keyboard are named by letter name.
  Example: a

CHAPTER 10: TESTING

10.1 TESTING PLAN

We have used Black box testing for the project

10.2 LEARN MORE ABOUT CTS

10.2 TEST SUITE DESIGN

10.2.1Test Cases

Homepage Test Suite and Test Case Design

Test Suite No: 1
Test Suite Detail: All functionality of Heart rate monitoring.

Test Case ID	Function Name	Test Case (condition)	Expected Results	Actual Result	Pass/Fail
1	Heart rate sensing.	Proper finger placed onto the sensor.	Reads the BPM accurately.	Reads the BPM accurately.	Pass
2	Converting sensed reading to BPM range.	Proper reading measured.	Shows BPM in the predefined range for humans.	Shows BPM in the predefined range for humans.	Pass
3	Storing BPM reading in a text file.	No predefined condition needs to me met.	File created and data entered successfully.	File created and data entered successfully.	Pass
4	Transferring the created text file onto cloud.	When active internet connection found, text file gets automatically synced onto cloud.	File successfully stored onto the cloud.	File successfully stored onto the cloud.	Pass

CHAPTER 11: LIMITATION AND FUTURE ENHANCEMENT

11.1 LIMITATION

- It is hardware based system so it requires maintenance after regular interval of period.
- Without internet the text file might not get synced to the cloud.
- Sensor is without outer casing which sometimes leads to inaccurate BPM reading.

11.2 FUTURE ENHANCEMENT

- We can also provide support for smartphone/wearable devices by developing android/iOS application.
- By replacing hardware sensor to new upgraded technology, with BPM the blood pressure and blood platelets count can also be accurately measured.

CHAPTER 12: CONCLUSION AND DISCUSSION

12.1 SELF ANALYSIS OF PROJECT VIABILITIES

- When time is the major concerned for hospitals in emergencies this device can play a vital role in reducing the time compared to what a traditional device that reads the BPM does.
- This project is capable of storing the BPM reading in a text file over the cloud which can be accessed anytime, anywhere by doctors.

12.2 PROBLEM ENCOUNTERED AND POSSIBLE SOLUTIONS

- While implementing this project we faced problems like interfacing it with Raspberry Pi board and other components.
- Another problem was storing the sensed data over the cloud and retrieving it via web portal.

12.3 SUMMARY OF PROJECT WORK

- We have developed a system which helps the patient and doctor to monitor and read the BPM and store it over the cloud. The reading of BPM is pretty accurate. It can help doctor to keep track of BPM reading for all patient that he/she may be consulting to.