

1.0 Business Case

AUTOMATED HEART RATE MONITORING SYSTEM

THE PROJECT

- This project is designed to monitor heart patients or any people who need to measure the heart rate.
- It automatically sends an emergency call/message in case of an abnormal heart reading.



BENEFITS

- Detects abnormal heart readings
- Has high accuracy
- Fast
- Safe and reliable
- Age specific

HISTORY

- Nowadays smart watches have many features including heart rate monitors but none of them have emergency alerts in case of abnormal readings.
- Many smart watches also have a low accuracy PPG sensor.

APPROACH

Software Requirements:

- Windows 8+ or higher version of OS
- Any latest wearable OS
- Any programming language code editor

Hardware Requirements:

- Smart Watch
- PPG sensor
- Laptop or PC

LIMITATIONS

- High cost of accurate sensors
- Limited Features
- Developing Stage
- Cannot operate in no network area
- Emergency calls or messages may not deliver in time

DATE

16/03/2022

SUBMITTED BY

RAAVI VENKATESH
LAGAN MEHTA
AKASH KANDASWAMY

2 Stakeholder and User Description

| PROJECT NAME | AUTOMATED HEART RATE MONITORING SYSTEM | | | |
|----------------------------------|---|--|--|--|
| PREPARED BY | RAAVI VENKATESH, LAGAN MEHTA, AKASH.K | | | |
| DATE | 23/03/2022 | | | |
| PROJECT STAKE HOLDER NAME | SPECIFIC INFORMATION NEEDS | PROJECT INTERESTS | IMPACT ON PROJECT | ROLE |
| | Types & Frequency of Communication | Specific areas of interest and participation | Positive, Negative, Influencer, Supporter, Readblock | Decision Maker, Collaborator, Consultant, Information, Recipient |
| PROJECT SPONSOR | Allocating resources and financial support | Financial | Supporter | Budget Management |
| PROJECT MANAGER | Managing progress, solving problem | Technical and business project managers | Supporter | Decision making |
| DEVELOPER | Design Program, programming | Skilled and experience programmer | Positive | Participant |
| EMPLOYEE HEAD | Assigning the work, Forming the Workflow | Group Head & Team Leader | Supporter | Consultant |
| EMPLOYEE | Work on assigned task | Co-operating with colleagues | Positive | Collaborator |
| COLLEAGUES | Performing the Task | Co-operating with other members | Positive | Collaborator |
| CLIENTS AND CUSTOMERS | Checking the Demand and Complaints | Record arrangement and review | Negative | Participant |

2.1 Project Model



Justification of using Agile Development Methodology:

1. Reduces Technical Debt:

Technical debt refers to the maintenance tasks required to support the existing product. Those tasks include defect resolution, refactoring, and testing.

2. Easily and Quickly Adapt to Change:

Teams not only adapt to change in Agile, they are encouraged to embrace the practice. Agile acknowledges that customer needs change and that teams must be able to adapt.

3. Using Agile for Mobile Application Development and Testing Creates Total Alignment and Transparency:

An Agile software development process requires a level of collaboration and involvement that one would not find in a traditional waterfall project.

4. Agile Software Development and Test Minimize Risk:

Although teams do their best to plan the phases of a waterfall project, there is often a level of uncertainty that isn't typically found in Agile software development.

5. Higher Quality Product:

Waterfall methodology can negatively impact the quality of the product. In a waterfall methodology, project phases may be so full of features that developers must rush to complete them and little time is left for testing.

6. Predictable Delivery Dates:

Waterfall projects revolve around lengthy project cycles that make it difficult for teams to predict a release date accurately.

3.1 REQUIREMENTS

FUNCTIONAL REQUIREMENT:

- Registration Page
- Syncing Data
- Logic Module
- Emergency Module
- Contact Module

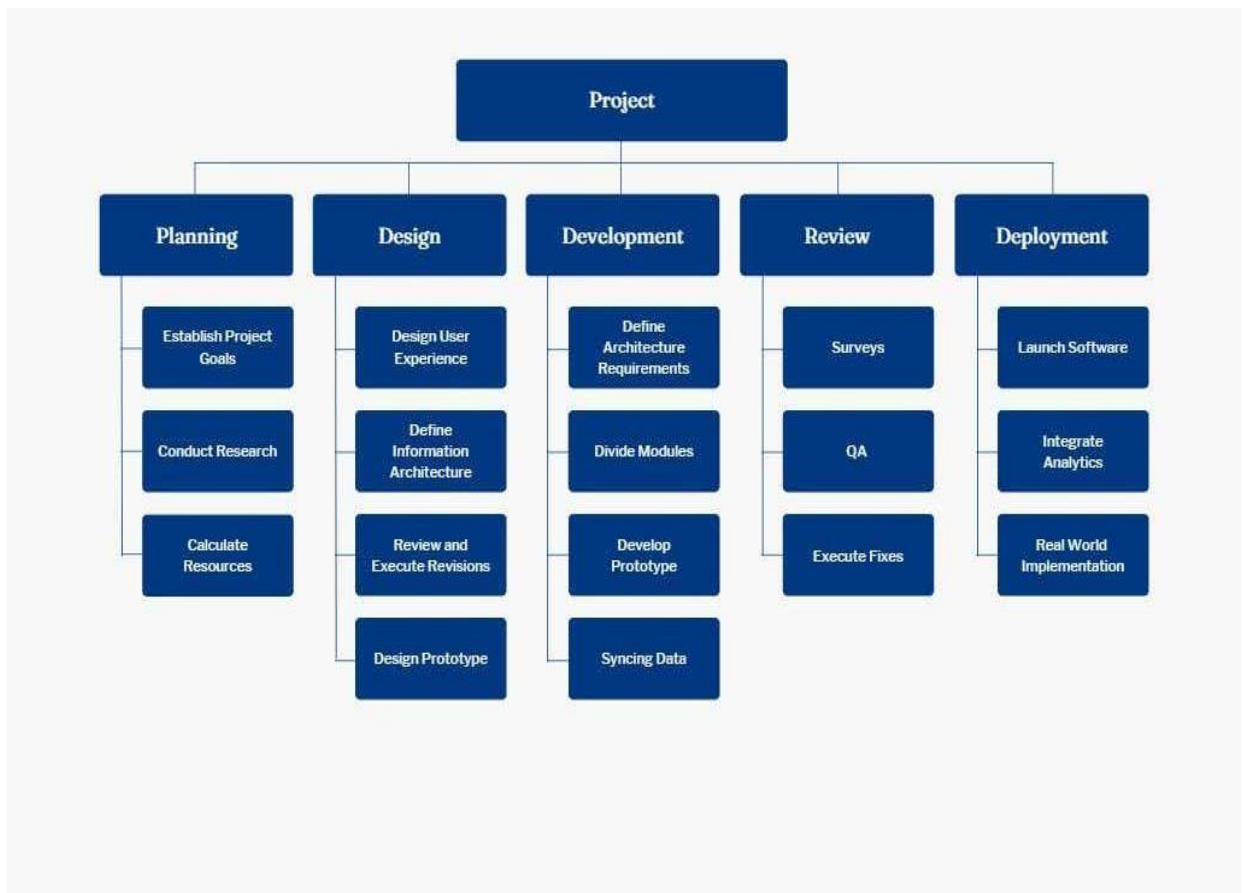
NON-FUNCTIONAL REQUIREMENT:

- Authentication
- Data Security
- Performance
- Maintenance

SURVEY:

- High Representativeness
- Good Statistical Significance
- Precise Results

4.1 PROJECT PLAN



Software and Hardware Requirements for Development:

Core i7 8th Generation Processor or above

8 Gb of RAM or higher

Windows 10 Single Language or Pro 27-inch Monitor or above

Reliable Internet Connection

Web installed and Environment variable should be done

PROJECT PLAN

COST ESTIMATION

In order to achieve efficient and effective management of software projects, it is important to estimate the size and cost of the project. For the size and cost estimation of our Chatbot project, we will use Function Point Analysis (FPA) and COCOMO model to predict the development effort of the statistical analysis package. Function point analysis and the COCOMO model are an accepted standard for the measurement of software size in software engineering.

Function Points Analysis

The FPA is a reliable method for measuring the size of computer software. It essentially measures functionality that the user requests and receives. It also measures the software development and maintenance cost and size independently of the technology used for implementation

The general approach that FPA follows is

Count the number of inputs, outputs, inquiries, master files, and interfaces required, then calculate the Unadjusted Function Points (UFP)

- Calculate the adjusted function point (AFP) by multiplying these counts by an adjustment factor; the UFP and the product complexity adjustment.

Calculate the Source Lines of Code (SLOC) with the help of the AFP and the Language Factor (LF)

1. Calculation of the unadjusted function points (UFP):

The FPA measures functionality that the user requires. The specific user functionality is a measurement of the functionality delivered by the application as for user request. The 5 function types identified are

- external input which receives information from outside the application boundary,
- external output which presents information of the information system,
- external enquiries which is special kind of an external output. An external inquiry presents information of the information system based on a uniquely identifying search criterion, without applying additional processing (such as calculations).
- internal logical files contains permanent data that is relevant other user The information system references and maintains the data and
- external interface files also contains permanent data that is relevant to the user. The information system references the data, but the data is maintained by another information system

For each function identified above the function is further classified as Simple, average or complex and a weight is given to each. The sum of the weights quantifies the size of information processing and is referred to as the Unadjusted Function points.

The table below shows the function types and the weighting factors for the varying complexities.

| Function type | Simple | Average | Comple |
|-------------------------|--------|---------|--------|
| Internal Logical File | 7 | 10 | 15 |
| External Interface File | 5 | 7 | 10 |
| External Input | 3 | 4 | 6 |
| External Output | 4 | 5 | 7 |
| External Inquiry | 3 | 4 | 6 |

Using these definitions above, the files types in my project can be counted as follows:

| | | Weighting Factor | | | Count |
|-------------|-------------------------------------|------------------|---------|---------|-------|
| | | Simple | Average | Complex | |
| Inputs | Member Login | 3 | | | 15 |
| | Member Registration | | 4 | | |
| | Select question for data analysis | | 4 | | |
| | Select answer for option analysis | | 4 | | |
| Outputs | Member login confirmation | 3 | | | 12 |
| | Member Registration confirmation | 3 | | | |
| | List of possible enquiry | | | | |
| | Graph/Table of user choice analysis | 3 | | | |
| | Live Customer Support enquiry | 3 | | | |
| Inquiries | Validate member information | | 4 | | 8 |
| | View accepted documentation list | | 4 | | |
| Files | User Input Database | | 10 | | 20 |
| | User Response Database | | 10 | | |
| Interface s | Application server to database | | | 10 | 20 |
| | User to application server | | | 10 | |
| Total UFP | | | | | 75 |

2. Calculate Adjusted Function Point

To calculate the Complexity adjustment value, several factors have to be considered, such as Backup and recovery, code design for reuse, etc. All the factors and their estimated values in this project are shown in the following table. The adjusted function point denoted by FP is given by the formula:

FP = total UFP * (Complexity adjustment factor)

Total complexity adjustment value is counted based on responses to questions called complexity weighting factors in the table below. Each complexity weighting factor is assigned a value (complexity adjustment value) that ranges between 0 (not important) to 5 (absolutely essential).

Table Adjusted Function Points

| Number | Complexity Weighting Factor | Value |
|--------|--|-----------|
| 1 | Backup and recovery | 1 |
| 2 | Data communications | 2 |
| 3 | Distributed processing | 2 |
| 4 | Performance | 5 |
| 5 | Existing operating environment | 3 |
| 6 | On-line data entry | 3 |
| 7 | Input transaction over multiple screens | 1 |
| 8 | Information domain values complex | 5 |
| 9 | Application designed for change | 4 |
| | Total complexity adjustment value | 26 |

3. Calculate the Source Lines of Code (SLOC) and the formulas used

€

- Total Unadjusted Function Points (UFP) = 75
- Product Complexity Adjustment (PC) = $0.65 + (0.01 * 26) = 0.91$
- Total Adjusted Function Points (FP) = $UFP * PC = 68.25$
- Language Factor (LF) for C++ assumed as = 50
- Source Lines of Code (SLOC) = $FP * LF = 3413$

Cost Drivers:

| | Very Low | Nominal | High | VeryHigh |
|-------------------------------|----------|---------|------|----------|
| Product Attributes | | | | |
| Required Software Reliability | 0.75 | | | |
| Size of Application Database | | 1.00 | | |
| Complexity of The Product | 0.70 | | | |

| | | |
|-------------------------------|------|------|
| Hardware Attributes | | |
| Memory Constraints | | 1.00 |
| Personnel attributes | | |
| Applications experience | | 0.91 |
| Programming language | | |
| experience | 1.14 | |
| Project Attributes | | |
| Application of software | | |
| engineering methods | | 1.00 |
| Use of software tools | 1.24 | |
| Required development schedule | | 1.08 |

COCOMO Model

The COCOMO model is a good measure for estimating the number of person- months required to develop software. Our project, Muzi : Smart Enquiry Campus Chatbot is an Utility program. The table below presents the COCOMO formulae for different types of programs:

| TDEV | Programmer Productivity | Development Time (Month) |
|---|--|--|
| Application Programs Utility Programs System Programs | $PM = 2.4 * (KDSI)^{1.05}$ $PM = 3.0 * (KDSI)^{1.12}$ $PM = 3.6 * (KDSI)^{1.20}$ | $PM = 2.5 * (PM)^{0.38}$ $PM = 2.5 * (PM)^{0.35}$ $PM = 2.5 * (PM)^{0.32}$ |

Using the above formula for the application programs,

The programmer productivity and the development time are as follows: $KDSI = 2.5$
KLO

$$PM = 2.4 * (2.5)^{1.05} = 6 \text{ person-month}$$

$$TDEV = 2.5 * (6.28)^{0.38} = 5.02 \text{ month}$$

4.1.2 Job Description and Responsibilities

| NAME | ROLE | RESPONSIBILITIES |
|-------|-------------------|---------------------------------------|
| RAAVI | PROJECT MANAGER | MANAGE THE PROJECT |
| AKASH | BUSINESS ANALYST | DISCUSS AND DOCUMENT REQUIRMENTS |
| LAGAN | TECHNICAL LEAD | DESIGN THE END-END ARCHITECTURE |
| RAAVI | UX DESIGNER | DESIGN THE USER INTERFACE |
| LAGAN | BACKEND DEVELOPER | DESIGN, DEVELOP AND UNIT TEST SERVICE |
| AKASH | DATA ANALYST | DESIGN AND ANALYSIS OF DATA |
| ALL | TESTER | DEFINE TEST CASES AND PERFORM TESTS |

5.1 Risk Management

Risk from Outer World :

Presence of Similar kind of service provides.

Competitive prices from Different kind of Websites.

High cost of sensors.

Low Website speed.

Server End :

Disk failure of the database system.Slow data iteration.

Improper connection to the internet.

Solution :

Installation of the server across the globe.

Using of better disk drive and time to time server maintenance.

Using of better algorithm for the iteration data.

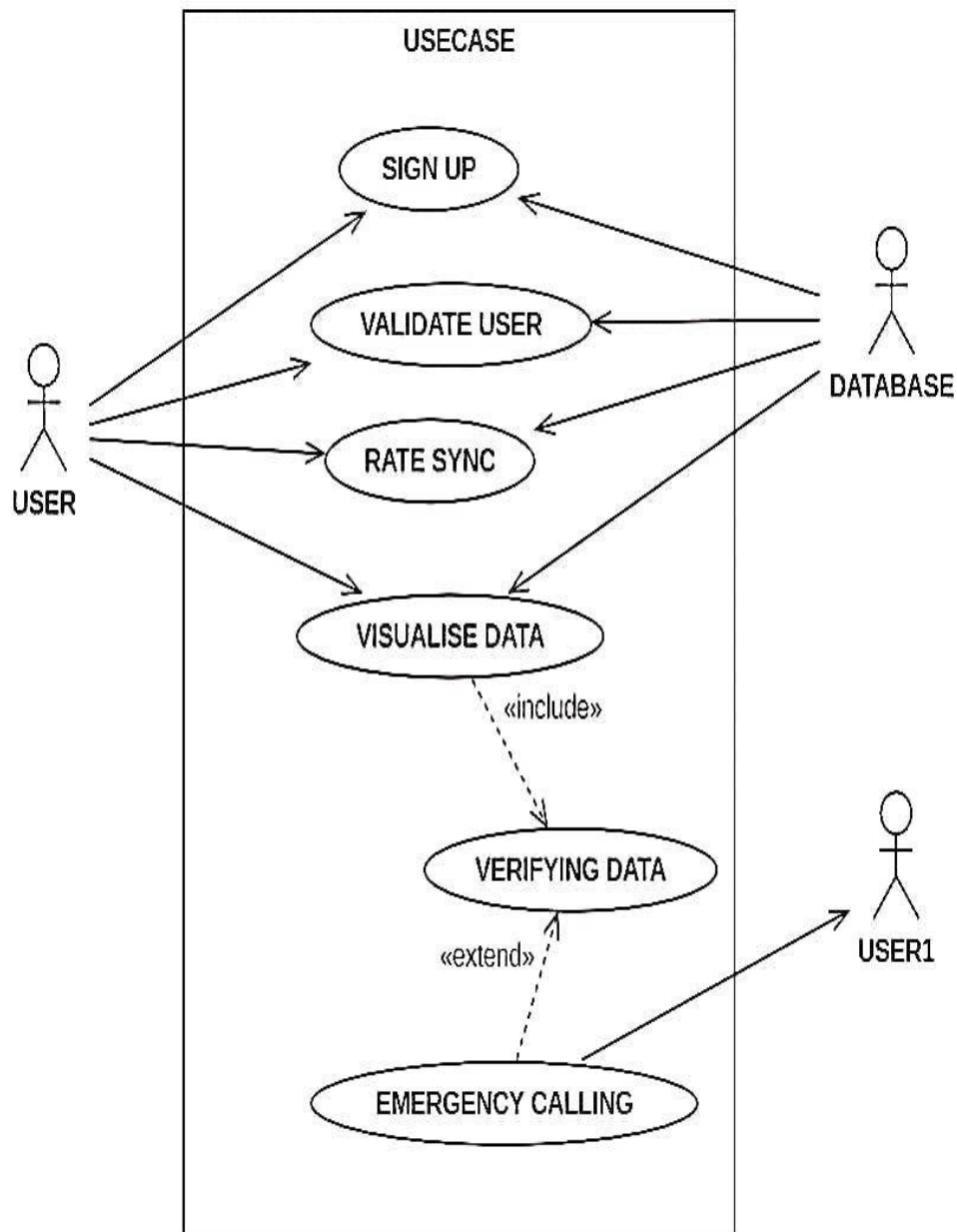
RISK MATRIX



| PROBABILITY | IMPACT CRITERIA | | | |
|-------------|-----------------|----------------------|-------------------------|----------------------|
| | NEGLIGIBLE | MARGINAL | CRITICAL | CATASTROPHIC |
| CERTAIN | SENSOR | | | |
| LIKELY | | INCORRECT DETAILS | | |
| POSSIBLE | | INCORRECT MODE | NETWORK INTERRUPTION | |
| UNLIKELY | | | SYNCING ERROR | |
| RARE | | | WATCH MALFUNCTION | INCORRECT READING |

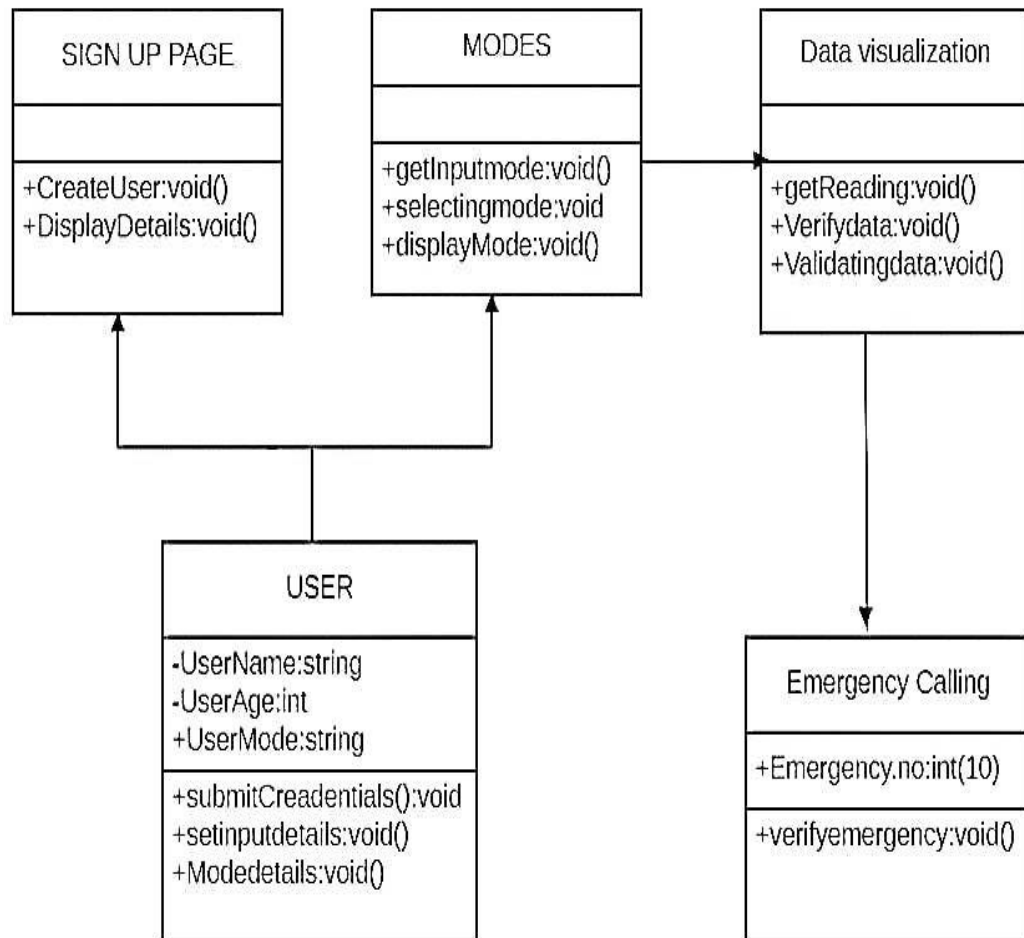
6.1 UML Diagrams

6.1.1 Use case Diagram :



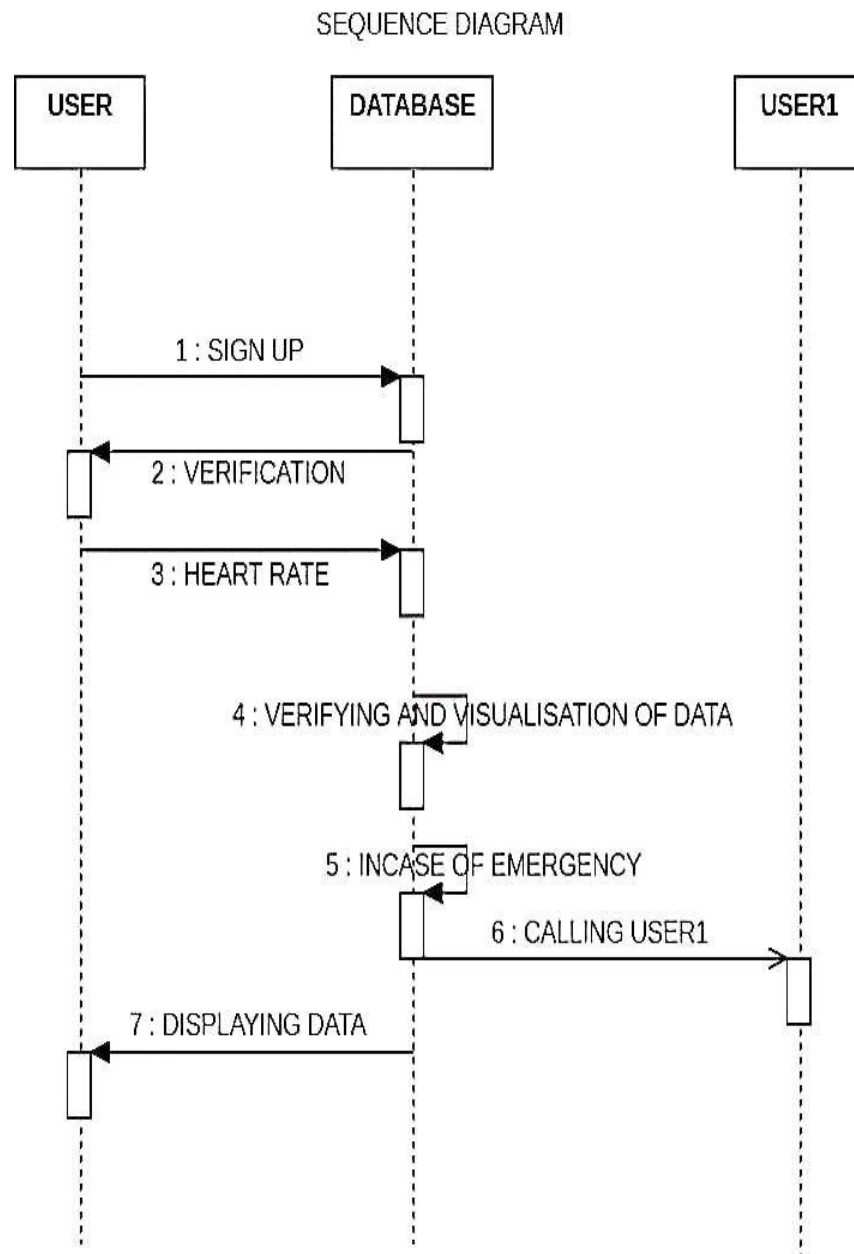
7.1.1 Class Diagram :

CLASS DIAGRAM



8.1 Structural UML Diagrams :

8.1.1 Sequence Diagram :



Sequence Diagram Description

1. LIFE LINE:

- A) USER
- B) DATABASE
- C) USER1

2. MESSAGE AND TYPES:

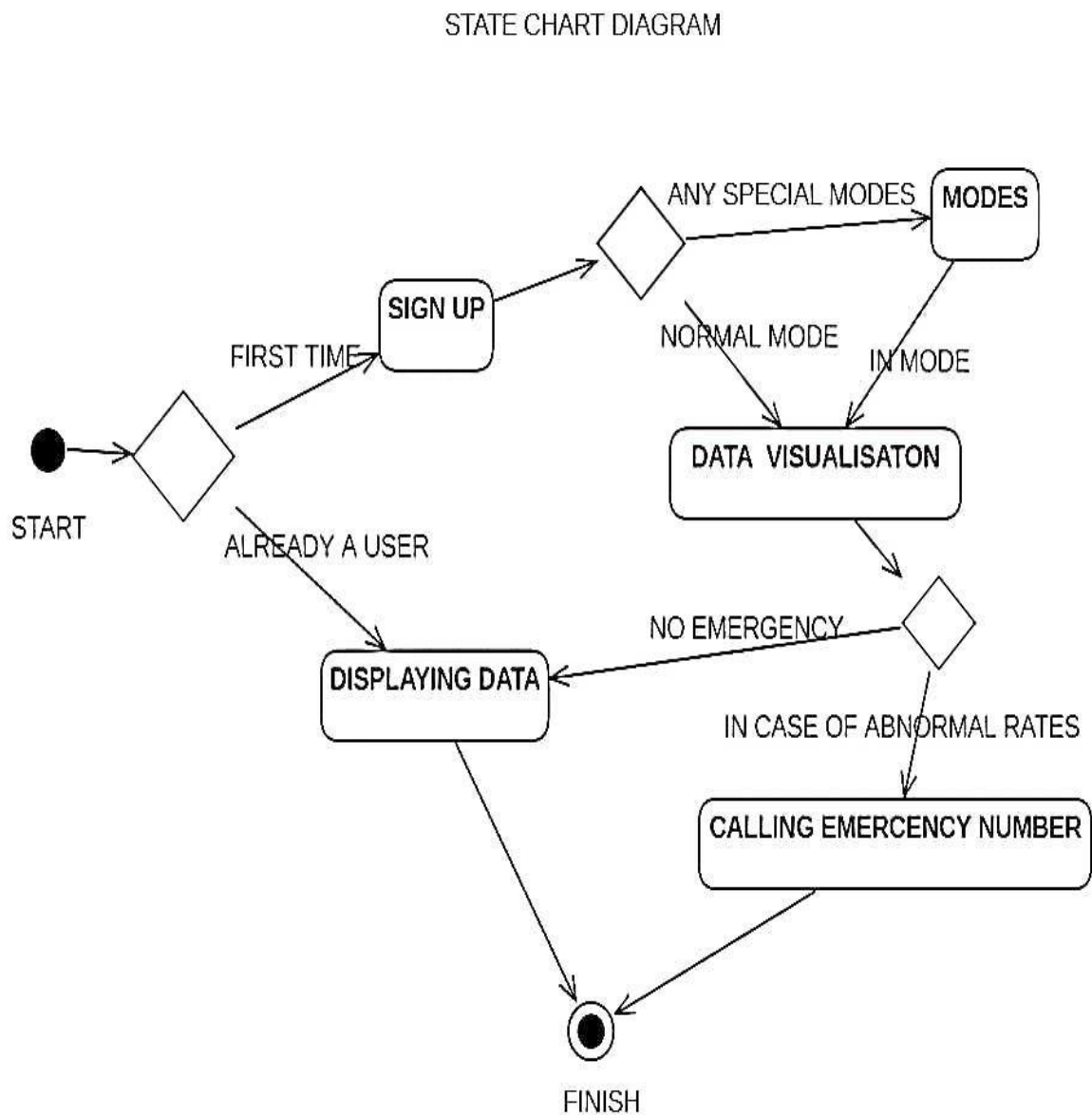
A) Recursive Message: Recursive message is a kind of message that represents the invocation of message of the same lifeline. Its target points to an activation on top of the activation where the message was invoked from.

B) Create Message: A message defines a particular communication between Lifelines of an Interaction and is a kind of message that represents the instantiation of (target) lifeline.

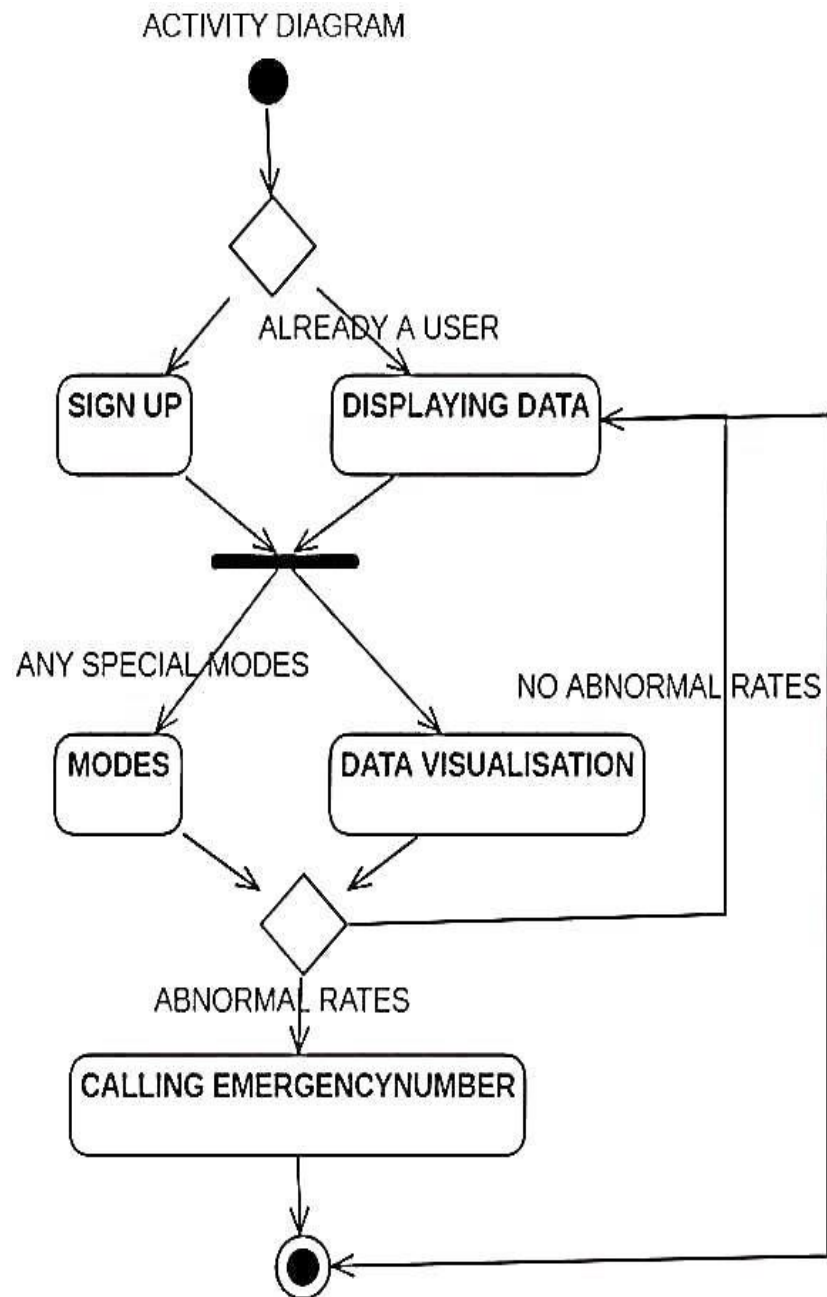
3. DESCRIPTION:

In this after the registration part the database collects the heart rates and analyse according to the results of the analysis it will call for Emergency number and Help the user.

8.1.3 State chart Diagram :



Activity Diagram



STATE CHART DESCRIPTION

1) STATES:

A) Sign up:

This state goes when user is new to this application.

This contains the name, number, gender and age which helps in analysis of the user health.

B) Displaying data:

This option is for who is already a user or the signup once . which show the data analysis of the user

C) Data visualization:

This is the basic analysis of data and showing using Graphic user interface . which makes a user to observe and follow the tip.

D) Modes:

The heart rates changes according to task we do so according to the user work and stats we can change the mode which increases the accuracy of the analysis. which reduces the errors occurs due to bad rate reading.

E) Calling Emergency number:

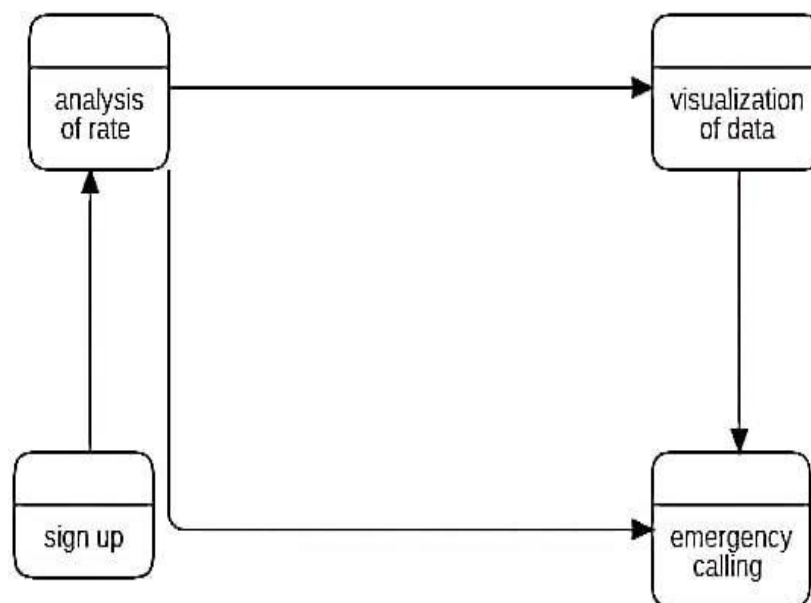
This will enable itself which makes the user not to suffer during emergency which will automatically call the number which can send the help immediately.

2) Transition:

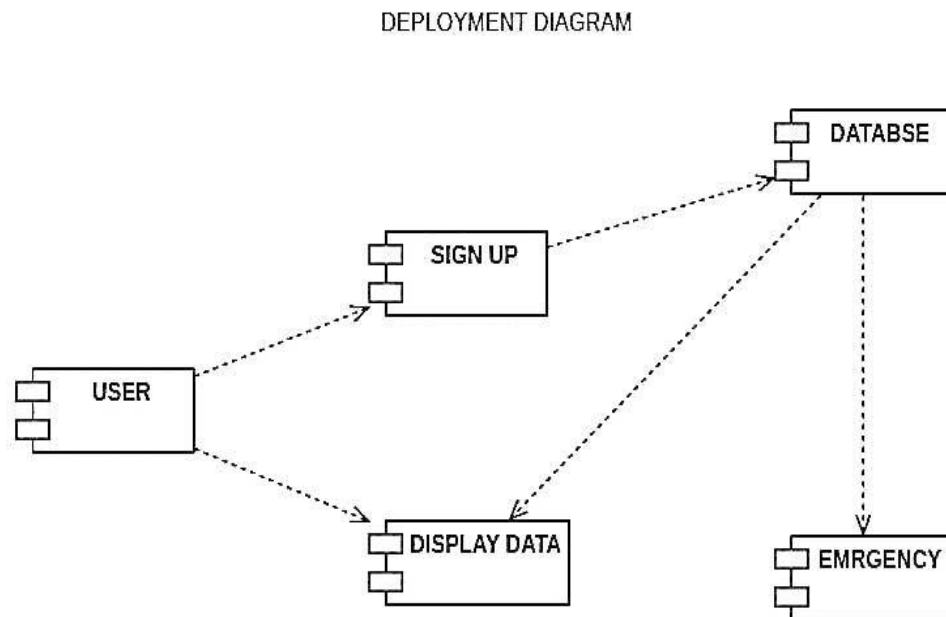
There are only two transition in this application which from start to the sign up and direct data page and other is from the Database to the calling server.

9.1.1 Dataflow Diagram :

DATAFLOW DIAGRAM



10.1 Deployment Diagram :

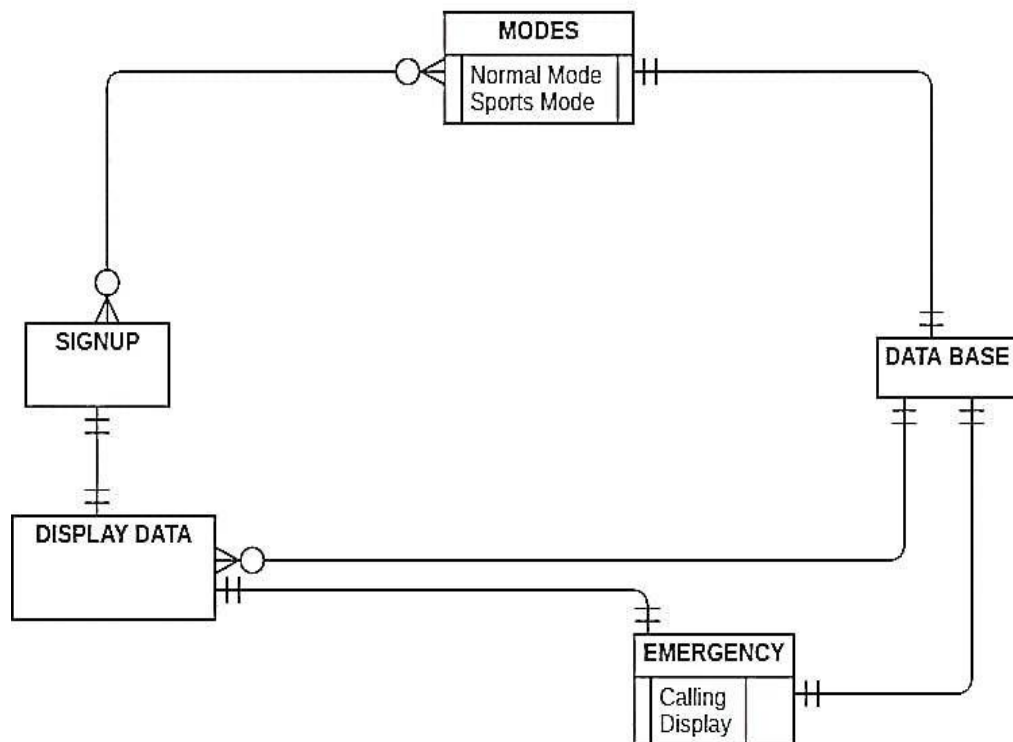


Deployment Diagram:

This Diagram shows how the clients/servers are connected to the Database. Hardware or Software Object shown by three-dimensional box called Node. Some examples are Maintenance Server, Web Server, Agents, Application Server etc.

11.1.1 Entity- Relationship Diagram :

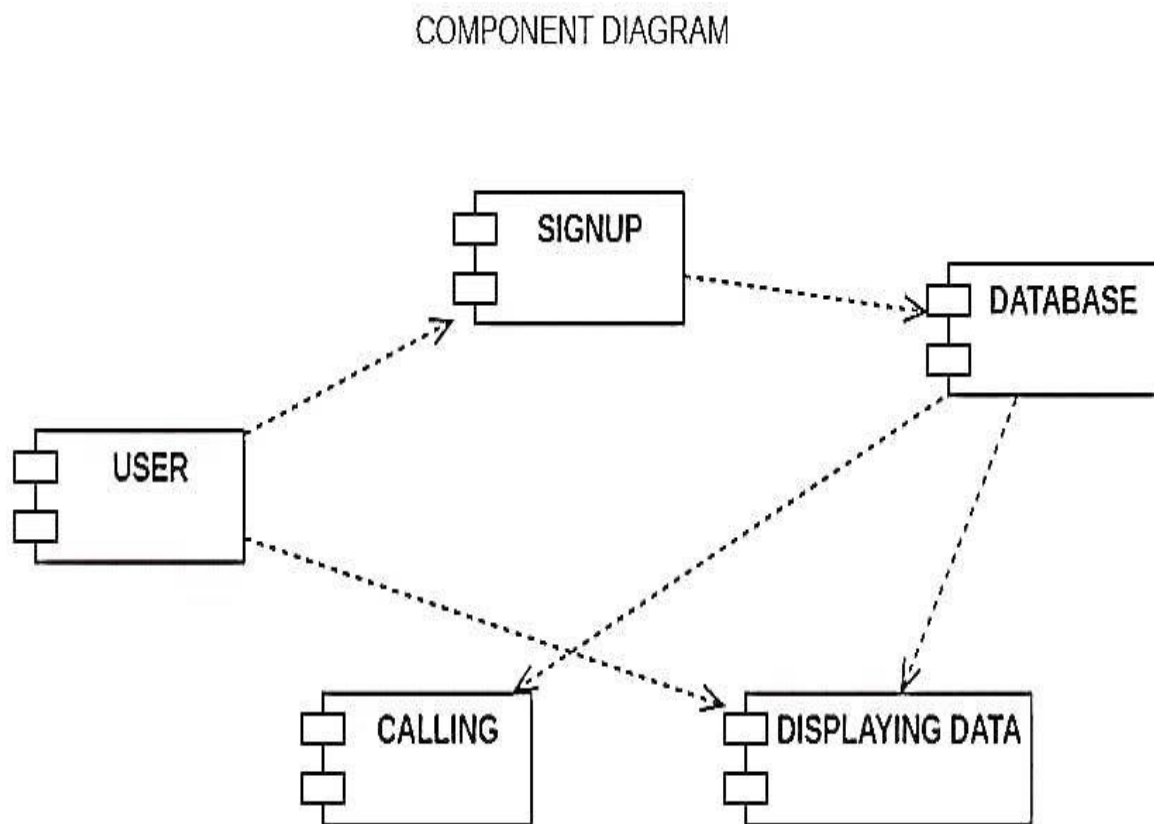
ER DIAGRAM



ER Diagram:

ER diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

12.1.1 Component Diagram :



COMPONENT DIAGRAM :

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a Railway Reservation System. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

MODULES

13.1.1 Module & Module Description :

USER MODULE :

- *The Login Module is a portal module that allows users to type a user name and password to log in.*
- *You can add this module on any module tab to allow users to log in to the system.*
- *If user hasn't yet registered for this site then the user can Register and Log-in.*

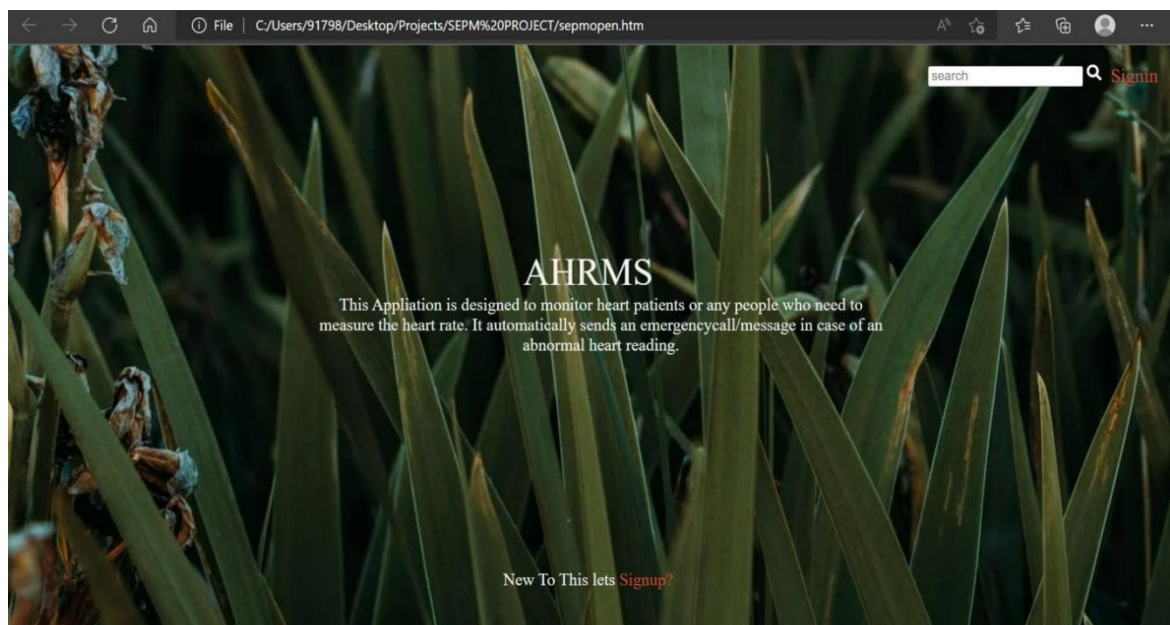
DATA MODULE :

- *This module contains the data analysis and gives the user tips and the basic output for it*
- *This module take the data from the sync option from the smart watches and store it to analysis*

14.1 IMPLEMENTATION

14.1.1 Module Implementation:

HOME PAGE:



Sign-up module:

A screenshot of a web browser displaying a registration form titled "REGISTRATION". The form is centered on a background image of green grass. The form fields include: Name (with placeholder "Enter your name"), Email (with placeholder "Enter your Email"), Gender (with placeholder "Enter your Gender(M/F/O)"), Age (optional, with placeholder "Age"), and City (with placeholder "No.1 ABC Street, Delhi, India"). Below these fields are three radio buttons for "Would you recommend this watch to a friend?" with options "Definitely", "Maybe", and "Not sure". There is also a text area for "Any comments or suggestions?" with placeholder "Enter your comment here...". A "Submit" button is at the bottom of the form.

Data module:

A screenshot of a web browser displaying a data module. The background is the same green grass image. The form shows the following data: Age (23), Heart Rate 1 (123), Heart Rate 2 (123), Heart Rate 3 (123), Heart Rate 4 (123), and Emergency Number (123). A red button labeled "GET THE REPORT" is positioned below the Emergency Number field. Below the button, the following text is displayed: "Target Heart Rate (HR) Zone (60-85%): ** 120 – 170", "Predicted Maximum HR: 200", "your average heart rate:123", and "Keep moving on. ...".

15.1 DESIGNING TEST SUITS

15.1.1 Web Application Testing:

WEB TESTING, or website testing is checking your web application or website for potential bugs before it's made live and is accessible to general public. Web Testing checks for functionality, usability, security, performance of the web application or website.

During this stage issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked.

Functionality Testing:

This is used to check if your product as per the specifications you ended for as well as the functional requirements you charred out for in your developmental documentation. Web based Testing Activities includes:

Test all links in your webpages are working correctly and make sure there are no broken links. Links to be checked will include:

Usability testing:

Usability Testing has now become a vital part of any web based project can be carried out by testers like you or a small focus group sort the target audience of the web application

Test the site Navigation:

Menus, buttons or Links to different pages on your site should be easily visible and consistent on all webpages Test the Comments

Content should be legible with no spelling or grammatical mistakes if present should contain an "alt" text

Interface Testing:

Three areas to be tested here are-Application, Web and Database Server

Application Test requests are sent correctly the Database and output at the client side is displayed correctly. If any must be caught by application and must be only shown to the administrator and not the end Web Server: Test Web server is handling all application requests without any service denial

Database Server: Make sure queries sent to the database give expected results.

Test system response when connection between the three layers (Application, Web and Database) cannot be shed and agent

Database Testing:

Database is one critical component of your web application and stress must be laid to test it thoroughly. Testing activities will include

Performance Testing:

This will ensure your site works under all loads. Software Testing activities will include but not limited to

Website application response times at different connection speeds Load test your web application to determine its behaviour under normal and peak

loads Stress test your web site to determine its break point when pushed to beyond

Test if any errors are shown while executing queries

Data Integrity is maintained while creating, updating or deleting data in database.

Check response time of queries and fine tune them if necessary. Test data retrieved from your database is shown accurately in your web application

normal loads at peak time. Test if a crash occurs due to peak load, how does the site recover from such an event

Make sure optimization techniques like gzip compression, browser and server side cache enabled to reduce load times

Compatibility

Compatibility tests ensure that your web application displays correctly across different devices. This would include

Security testing:

Security Testing is vital for e-commerce website that store sensitive customer information like credit cards. Testing Activities will include

Test unauthorized access to secure pages should not be permitted

Browser Compatibility Test: Same website in different browsers will display differently. You need to test if your web application is being displayed correctly across browsers, JavaScript, AJAX and authentication is working fine. You may also check for Mobile Browser Compatibility.

The rendering of web elements like buttons, text fields etc. changes with change in Operating System. Make sure your website works fine for various combination of Operating systems such as Windows, Linux, Mac and

Browsers such as Firefox, Internet Explorer, Safari etc.

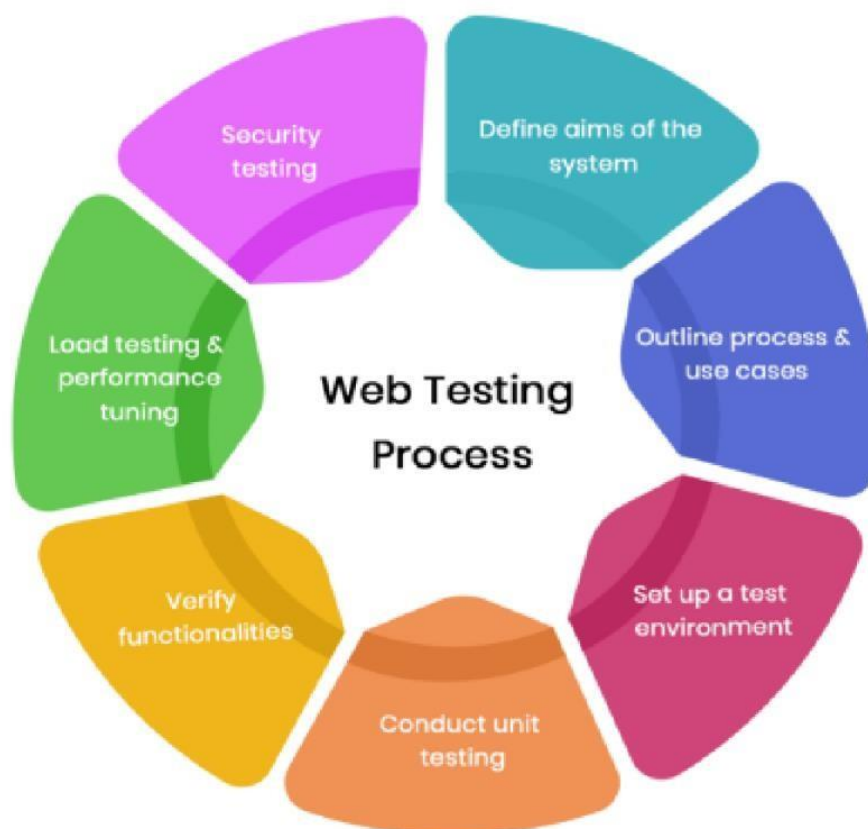
Restricted files should not be downloadable without appropriate access

Check sessions are automatically killed after prolonged user inactivity on use of SSL certificates, website should re-direct to encrypted SSL pages.

Crowd Testing:

You will select a large number of people (crowd) to execute tests which otherwise would have been executed by a select group of people in the company. Crowdsourced testing is an interesting and upcoming concept and helps unravel many unnoticed defects

15.1.1 Master Test Plan Diagram :



15.1.3 Master Test Plan

Testing Objective:

**Focusing on load management,
security and web site loading time.**

Testing Objective

Functionality Testing

Usability Testing

Test Items:

Interface Testing

Database Testing

Compatibility Testing

Performance Testing

Security TestingCrowded

Testing

Features To be Tested

**Login, Sign/up Process and website
faster loading time.**

**Features Not To be
Tested**

Usability Testing,

Method

Manual Testing

17.1.1 Manual Testing

| Testing | Input | Description | Result |
|----------------------------|------------------|---|--------|
| Login And Sign Up | specified ids | Implementing bot and specified ids to Login and Sign up | PASS |
| Load Up time | Website on Local | Testing website with opening and host performing simultaneously on system | PASS |
| Database test | Test cases | Implementing test and verifying output | PASS |
| All the data Specification | Test cases | Implementing data in java Script Checking the speed of it | PASS |
| Attributes used | Filling form | By filling the registration Form in sign up page | PASS |

Test case1:

Sample Input:

Age: 23
Heart Rate 1:120
Heart Rate 2:132
Heart Rate 3:50
Heart Rate 4:30
Emergency Number:108

Expected output:

Predicted Maximum HR: 200
Predicted Lowest HR:60
calling emergency numbers:108

Output:

The screenshot shows a mobile application interface with a green leaf background. At the top, there is a label 'Age' with a text input field containing '23'. Below this, there are four labels: 'Heart Rate 1', 'Heart Rate 2', 'Heart Rate 3', and 'Heart Rate 4', each with a corresponding text input field containing '120', '132', '50', and '30' respectively. Below these, there is a label 'Emergency Number' with a text input field containing '108'. A red button with the text 'GET THE REPORT' is positioned below the input fields. At the bottom of the screen, the following text is displayed: 'Predicted Maximum HR: 200', 'Predicted Lowest HR:60', and 'calling emergency numbers:108'.

Test case2:

Sample Input:

Age: 14
Heart Rate 1:120
Heart Rate 2:132
Heart Rate 3:120
Heart Rate 4:120
Emergency Number:108

Expected output:

your average heart rate:123
Check your heart rate time to time and maintain good diet

Output:

The screenshot shows a mobile application interface with a green leaf background. At the top, there is a label 'Age' with a text input field containing '14'. Below this, there are four labels: 'Heart Rate 1', 'Heart Rate 2', 'Heart Rate 3', and 'Heart Rate 4', each with a corresponding text input field containing '120', '132', '120', and '120' respectively. Below these, there is a label 'Emergency Number' with a text input field containing '108'. A red button with the text 'GET THE REPORT' is positioned below the input fields. At the bottom of the screen, the following text is displayed: 'your average heart rate:123' and 'Check your heart rate time to time and maintain good diet'.

Test case3:

Sample Input:

Age: 45
Heart Rate 1:100
Heart Rate 2:120
Heart Rate 3:120
Heart Rate 4:100
Emergency Number:108

Expected output:

Target Heart Rate (HR) Zone (60-85%): 102-145
Predicted Maximum HR: 170
your average heart rate:110
Keep moving on. . .

Output:

The screenshot shows a mobile app interface with a green leaf background. At the top, there are four input fields for heart rates: 'Heart Rate 1' (100), 'Heart Rate 2' (120), 'Heart Rate 3' (120), and 'Heart Rate 4' (100). Above these is an 'Age' field with the value 45. Below the heart rate fields is an 'Emergency Number' field with the value 108. A red button labeled 'GET THE REPORT' is centered below the emergency number. At the bottom, the app displays the following text: 'Target Heart Rate (HR) Zone (60-85%): 102-145', 'Predicted Maximum HR: 170', 'your average heart rate:110', and 'Keep moving on. . .'

Test case4:

Sample Input:

Age: 32
Heart Rate 1:100
Heart Rate 2:120
Heart Rate 3:120
Heart Rate 4:130
Emergency Number:108

Expected output:

Target Heart Rate (HR) Zone (60-85%): 108-153
Predicted Maximum HR: 180
your average heart rate:117.5
Keep moving on. . .

Output:

The screenshot shows a mobile app interface with a green leaf background. At the top, there are four input fields for heart rates: 'Heart Rate 1' (100), 'Heart Rate 2' (120), 'Heart Rate 3' (120), and 'Heart Rate 4' (130). Above these is an 'Age' field with the value 32. Below the heart rate fields is an 'Emergency Number' field with the value 108. A red button labeled 'GET THE REPORT' is centered below the emergency number. At the bottom, the app displays the following text: 'Target Heart Rate (HR) Zone (60-85%): 108-153', 'Predicted Maximum HR: 180', 'your average heart rate:117.5', and 'Keep moving on. . .'

19.1 EFFORTS AND RESOURCES

19.1.1 Website development process

Conclusion:

To overcome these issues we have developed this website.

The development process for this website is characterized by the efforts made by the whole team and it also requires lots of hardware and software infrastructures. The hardware may include a PC with a core i7 processor, 16 GB RAM and a graphics card at least MX150, and software like HTML, CSS, JAVA-SCRIPT etc.

The integration effort comprises not only the design and realization of interfaces, but also the test of those interfaces. The more complex the subsystems are, the more effort is required for the interface test since the necessary test drivers/stubs should be equally complex.

19.1.2 RESOURCES

Wikipedia: For several references on various topics.

Geeks For Geeks : For learning technical concepts.

W3Schools For diagrams and related things.

Engineering for Change