MINOR PROJECT

"HEALING MANAGEMENT SYSTEM"

B-10

Submitted By:

Sakshi Dubey(13103716)
Puja Kandel (13103729)
Mohammed Rizwan Ashraf(13103739)
Anupreksha Sinha(13103745)
Pranjal Yadav(9913103674)

Table of Contents

1.Abstract

This project aims at setting a rehabilitation Management System. Set a connection while making one computer as a server & making one or more computer as clients. Today it's beyond our imagination to image a world without networks. Networking has become an integral part of our lives.

In 'Healing Management System', we'll virtually try to work as an organization which works for the distressed, depressed, addicted people . We expect curing them not only needs medicinal requirements but also a very strict check over their regular routine.

2. Introduction

In this organization, we have officials who organize a regular online test and medical treatment test. All the client users will be allotted a login id through which they can access their account & read the updates. Intense Graphical User Interface work is expected to be used here. Apart from the front add features, we've also taken care of strong backhand. Moving with the latest technology is the best thing one will ever wish to do & so are we trying to follow this legacy. Embedded features like Voice over Internet Protocol will maintain the flow of this project. A chat system is also designed with a server & multiple clients. We've proposed to give a touch of Internet of Things. When a patient develops a temperature beyond normal, the sensors put in the small wearable like watches, rings get activated and the notification is automatically send to the doctor on server to prescribe medicine. Also, the organization plans to keep a check on movement of addicted people .

3. Background Study and Findings

- *Python basics from W3 school*
- *Server & client set up study*
- *VoIP basics*
- *Arduino set up ethics*
- *Wireless Bluetooth Module basics*
- *Video streaming study*
- *Graphical User Interface Implementation*
- *File Transfer in python*
- *Designing using graphics study*
- *Database Management*
- *Pygames*

4. Requirement Analysis

4.1 Software Requirements

- *Python 2.6*

  Python Libraries:
  - ✓ *pygame set up*
  - ✓ *pip*
  - ✓ *numpy*
  - ✓ *pyaudio*
  - ✓ *open cv*
- *Sql for managing Database*
- *Aurdino for interfacing hardware implementation*
- *Tera term for hardware*

- ***Apps***
  - ✓ *Heal O India*
  - ✓ *Arduino Bluetooth Terminal*
  - ✓ *Heart Rate Sensor*

4.2.Hardware Requirements

- *Arduino*
- *Bluetooth Sensor*
- *RF Module*
- *Transmitter & Receiver*

4.3 Functional Requirements

*In Software engineering and systems engineering, a **functional requirement** defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.[1] Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.*

- *Study of patients needs*
- *Review of statitical data of hospitals*
- *Compatibility of Python with other environments*
- *Study of range of bluetooth & RF sensors*

4.4 Non - Functional Requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements.[1]

Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of "system shall do <requirement>", an

individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of "system shall be <requirement>", an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed.

- *Security issues with Login credentials*
- *Limited Active Range in VoIP*
- *Connection set up in Video Conferencing*
- *Installing of different libraries for py games*

4.5 User Requirements

4.5.1 Process Model

A Process Model tells us about how the data is processed and how the data flows from one table to another to gather the required information. This model consists of the Functional Decomposition Diagram and Data Flow Diagram.

Extreme programming in Agile process model.

Agile Model:

Agile development model is also a type of Incremental. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications.

- *Agile Methods:*
  - *Extreme Programming*
  - *SCRUM*
  - *OpenUP*
  - *LEAN Development*
  - *Crystal*
  - *Test Driven Development*

- *Extreme Programming: Extreme Programming (XP) takes an 'extreme' approach to iterative development.*
  1) *New versions may be built several times per day;*
  2) *Increments are delivered to customers every 2 weeks;*
  3) *All tests must be run for every build and the build is only if tests run successfully.*
  4) *There are some steps which are followed in extreme programming:*
     - *Incremental Planning*
     - *Simple and Effective Design*
     - *Test first development*
     - *Refactoring*
     - *Pair Programming*
     - *Continuous Integration*

➢ ***Incremental Planning:***

Requirements are recorded on Story Cards and the Stories to be included in a release are determined by the time available and their relative priority. In our project we used incremental planning. We divide our project requirements as story card.

➢ ***Simple and Effective Design:***

Enough design is carried out to meet the current requirements and no more. After the planning phase the next phase is the Design phase where application is designed according to requirements. This phase is very important phase because it will structure the layout of our application. Our project doesn't have extraneous information.

➢ ***Test first development:***

An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.

1) Writing tests before code clarifies the requirements to be implemented.

2) Tests are written as programs rather than data so that they can be executed automatically. The test includes a check that it has executed correctly.
3) All previous and new tests are automatically run when new functionality is added. Thus checking that the new functionality has not introduced errors.

- ***Refactoring:***

All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

1) Refactoring is the process of code improvement where code is re-organised and rewritten to make it more efficient, easier to understand, etc.
2) Refactoring is required because frequent releases mean that code is developed incrementally and therefore tends to become messy.
3) Refactoring should not change the functionality of the system.
4) Automated testing simplifies refactoring as you can see if the changed code still runs the tests successfully.

In our project we refactor the code at each stage of development because *harder it is to see the design in the code, the harder it is to preserve it, and the more rapidly it decays. Regular refactoring helps code retain its shape.*

We applied different Techniques that allow for more abstraction such as:

1) Replace type-checking code with State/Strategy.
2) Create more general types to allow for more code sharing.

- ***Pair Programming:***

1) In XP, programmers work in pairs, sitting together to developcode.
2) This helps develop common ownership of code and spreads knowledge across the team.

3) It serves as an informal review process as each line of code is looked at by more than 1 person.
4) It encourages refactoring as the whole team can benefit from this.
5) Measurements suggest that development productivity with pair programming is similar to that of two people working independently. In our development we apply some strategy based on pair programming:

1) *The task was something we are confident that we can complete in an hour or two.*
2) *The key to good pairing is to sync up very frequently—within seconds or a minute of noticing that we are out of sync. If we are spending five minutes (or more) out of sync, we might as well be coding solo, because it's the frequent re-sync'ing that creates the synergy of pairing.*
3) *Switch roles often—at least every half hour.*
4) *Rely on partner as well as support partner.*

- ***Continuous Integration:***

As soon as work on a task is complete it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.

4.6 Unified Modeling Language:

Describing a system at a high level of abstraction.

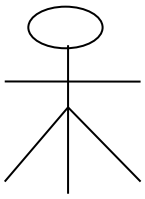
- *A model of the system*
- *Used for requirements and specifications*
- *It is an industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems*
- *The Unified Modeling Language uses mostly graphical notations to express the Object Oriented analysis and design of software projects.*
- *Simplifies the complex process of software design.*
- *Use graphical notation to communicate more clearly than natural language (imprecise) and code (too detailed).*
- *Help acquire an overall view of a system.*
- *Unified Modeling Language is not dependent on any one language or technology.*
- *Unified Modeling Language moves us from fragmentation to standardization.*
- *Unified Modeling Language includes mostly five diagrams:*
    - *Use case Diagram*
    - *Class Diagram*
    - *Sequence Diagram*
    - *State Diagram*
    - *Activity Diagram*

4.6.1 Use Case Diagram:

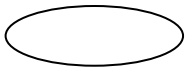
- *Use cases serve as a technique for capturing the functional requirements of a system.*
- *Describes the typical interactions between the users of a system and the system itself, providing a narrative of how a system is used.*
- *A use case consists of a set of one or more scenarios tied together by a common user goal.*
- *A scenario is a sequence of steps describing an interaction between a user and a system; some scenarios describe successful interaction; others describe failure or errors*
- *Users are referred to as actors; an actor is a role that carries out a use case*
- *An actor need not always be a person; it can also be an external system that is either automated or manual.*

- *A use case diagram is like a graphical table of contents of the use cases for a system*
  - *It shows the use cases, the actors, and the relationships between them.*
- *Use cases represent an external view of the system; consequently, they have no correlation to the classes in the system*

  -They can serve as a starting point for writing software validation test cases.
- *Use case diagram core components:*
  - ***Actor**: A role that a user plays with respect to the system, including human users and other systems. An external system that needs some information from the current system.*



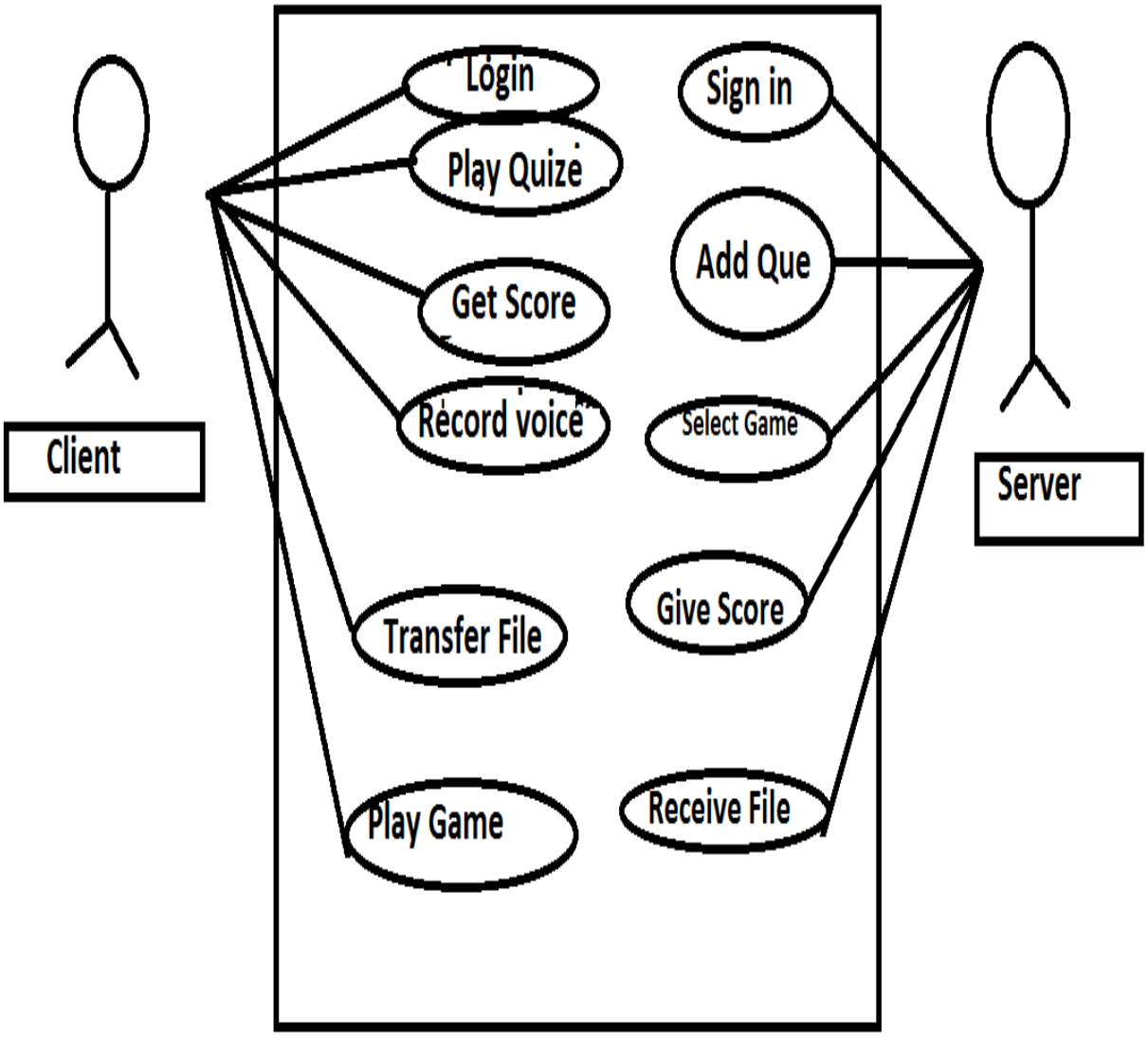
  - ***Use case:** A set of scenarios that describing an interaction between a user and a system, including alternatives.*



  - ***System boundary**: rectangle diagram representing the boundary between the actors and the system.*
  - ***Association**: communication between an actor and a use case; Represented by a solid line.*

  - ***Include**: a dotted line labeled <<include>> beginning at base use case and ending with an arrows pointing to the include use case. Include relationship occurs when a chunk of behavior is similar across more than one use case. Use "include" instead of copying the description of that behavior.*

- ***Extend***: *a dotted line labeled <<extend>> with an arrow toward the base case. The extending use case may add behavior to the base use case.*



5. Gantt Chart

*A **Gantt chart** is a type of bar chart, developed by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.*

Although now regarded as a common charting technique, Gantt charts were considered revolutionary when first introduced. This chart is also used in information technology to represent data that have been collected.

HISTORICAL DEVELOPMENT:

The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a harmonogram. Adamiecki published his chart in 1931, however, only in Polish, which limited both its adoption and recognition of his authorship. The chart is named after Henry Gantt (1861–1919), who designed his chart around the years 1910–1915.

One of the first major applications of Gantt charts was by the United States during World War I, at the instigation of General William Crozier.

In the 1980s, personal computers allowed for widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers. With the advent of the internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative groupware.

6. Detailed Design

6.1 Features

1. Multiclient Server

2. Aptitude Test

3. VoIP

4. Tkinter Chat box

5. Alarm Notification

6. Video Conferencing

7. File Sharing

8. Packet Data Loss Recognition

9. Games

10. Database Management using sql

11. Xamp

12. Temperature Measure

13. Transmitter & Receiver using RF

14. Bluetooth Operated Info

15. Android App Heal O India

7. Implementation

7.1 Server Client Connection

Server has the authority to initiate all the connections. Once the server has set up connections with clients, we can clearly observe the virtual picture of Healing Management System. Authorities to server

- *Can upload questions for Aptitude Test*
- *Can upload file of actual answers*
- *Can connect to client for Voice over Internet Protocol*
- *Can decide a particular pygame for particular client*

7.2 Tkinter Chat Box

In this module, we have provided a previlege to chat simultaneously with multiple clients.
It requires login credentials of both server & client. Once deatils are matched, we have a chat room for the same. It is one of the major feature of Graphical User Interface. It provides a platform to chat simultaneously.

7.3 Aptitude Test Conduction

Server owns the right to conduct Aptitude Test. In short, Doctor aims to conduct Aptitude Test for patients. Server can upload questions & right answers. Apart from this, server can upload a file . We can take a analogy of uploading files on fileserver.

7.4 Voice Over Internet Protocol

This is one of the interesting feature of this project. One client can exchange audio with other client. It can record voice for 20 seconds. It has a proper record in its database. We are able to accomplish this feature on python with the help of one to one client connection.

7.5 Video Conferencing

Video Conferencing has been achieved with the help of multiple clients & single server. It requires permission from server. If the login credential matches, the server possess the right to connect through one client & perform vedio conferencing. It has been achieved with the help of many inbuilt directories. It includes pynum, pip, pyaudio, open cv etc.

7.6 PyGames

Our sole motivation of this project is to present a charming rehabilitation system. Hence we have provided an entertainment portal for patients. It includes some of the games.

- *Angry Snake*
- *Moon Lander*
- *Mastermind*

7.7 Database Management System

Database manages all the backhand functioning of the program. It stores the crucial data of the project.

- *Login Username & password*
- *Score*
- *Questions uploaded by server*
- *Recorded Voice*

7.8 Graphical User Interface

One cannot imagine a project without GUI. Graphical User Interface has been implemented with utmost care. It includes many things namely:

- *Tkinter Chat Box*
- *Server client connection*
- *Games*

- *VoIP*
- *Aptitude Test*
- *Score Calculation*

7.9 File Sharing

We can share the file among several computers. It again acts like multiple client and server connection. It can exchange following things.

- *pdf*
- *text file*
- *images*
- *music*
- *video*

7.10 Packet Data Loss Recognition

Whenever we transfer a data, there is always a possibility of losing the data. Hence we tried to come up with this problem. In case the connection set up fails, we have tried to retrieve what percentage of data has already been transferred & how much is left. It will avoid repetitive exchange of data.

7.11 Arduino Based Implementation

It has been done with the concern when patient is not in the hospital. In times of sudden attacks, we felt an immediate need of transforming information. Hence I've tried to develop an alternative with the immediate information to Doctor, family, relatives & colleagues in case one faces a sudden stroke. It has been achieved with three different modules depending upon the way of sending signals.

- *Transmitter Receiver Module*

Transmitter will be kept at Patients side & Receiver is kept at Doctors side. As stated these devices are meant when patient is not in the approach of doctor. It sends 4 varieties of signals with the help of trigger button. It works on Radio Frequency.

- *Mobile Operation using Bluetooth Sensor*

Usually we just carry our cell phones while doing our daily work. With this intention in mind, I've connected sensors with cell phone using Bluetooth. Hence we just need to send a code & led will glow in Doctors cabin. Hence immediate care can be given.

- *Temperature Measurement*

I've made a set up through which we can measure the temperature of a body & its reading will be automatically sent to server which will be operated by Doctor

7.12 "Heal O India" Android App

Our project includes an Android App which is meant for good health of people. It contains following pages.

- *Healthy India: It's the home page of app*
- *Emergency Call: It has Login details. Once done, we can make an immediate call.*
- *Contact: Contains Personal Information*
- *Enquiry:*
    - ✓ *Appointment: Fix appointment with Doctor*
    - ✓ *Online Medical Help: Can Order medicines*
- *Daily Health: Motivational Website for good health*
- *Lets Chat: Chatting options on whatsapp, skype*
- *Events: Shares latest events*
- *Share: Entertainmemt poratl includes facebook, instagram, google, twitter*

8. Division of Project Work

MODULE: *SAKSHI DUBEY*

Objective: *Working with IOT*

Description: *The communication today is widely dependent on network & so on Internet of Things. The* ***Internet of Things*** *(IoT) is the network of physical objects—devices, vehicles, buildings and other items which are embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data.*

Details:

- *Working with wireless Radio Frequency Module of Transmitter & Receiver*

  00 Everything is fine
  01 Fever
  10 High Pulse Rate
  11 Stroke or Emergency Situation

- *Mobile operated signal using bluetooth sensor*

  It sends 1 or 0 via cell & led glows at home/doctor
- *Working with Temperature*

  If there is a sudden increase in temperature of patient, then notification is send to Dr.

- ***Android App: Heal O India***

  Features : *Emergency Call*
  Book Appointment
  Order Medicine
  Chat system

Additional Requirements:

- *Aurdino, Tera Term Software*
- *Bluetooth Module*
- *Temperature Sensor*
- *RF Module*
- *Arduino Bluetooth Terminal App*
- *Heart Beat App*

MODULE: *PUJA KANDEL*

Objective: *Thinker Bells*

Description: *This module deals with the online interaction of many computers at one time. It might be understood as conducting an online competition with time constraint. Not only for online quizzes but it can also be used as the entertainment portal. Also an interesting feature of voice conversation is added up to facilitate mass communication. Voice over IP (**VoIP**) is a methodology and group of technologies for the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Other terms commonly associated with**VoIP** are IP telephony, Internet telephony, broadband telephony, and broadband phone service.*

Details:

- *Conducts Online Medical Test*
    - ✓ *Upload questions*
    - ✓ *Multi chat server*
    - ✓ *Calculation of scores*
- *Works on Voice on Internet Protocol:*
    - ✓ *Records voice*
    - ✓ *Uses concept of multithreading*
- *Database Handling: Stores database of questions & answers*
    - ✓ *Work on sql*
    - ✓ *Work on Xamp*
    - ✓ *Fetches stored database*
- *Works on Multi Client Server*
    - ✓ *Creating hotspot using IP Adress in muti client*

Additional Requirements:

- *Database of online test*
- *Instant score updating*

MODULE: *MOHAMMED RIZWAN ASHRAF*

Objective: *GUI*

Description: *In computer science, a graphical user interface or* ***GUI****, is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. This deals with the designing of graphical user interface . It will act as the main front end. It will be basically based on tkinter.*

Details:

- *Works on front end using Graphical User Interface*
  - ✓ *File Sharing feature*
  - ✓ *Sign up & Log in page*
  - ✓ *Calculation of scores*
  - ✓ *Aptitude Test*
  - ✓ *Tkinter CHat*
- *Py Games*
  - ✓ *Mastermind*
  - ✓ *Moon Lander*
  - ✓ *Angry Snake*
- *SM Upload*
  - ✓ *Uploads file on server*
  - ✓ *Needs authentication from server*

Additional Requirements:

- *Wide application of Tkinter*
- *Notification when file is received by client*

MODULE: *ANUPREKSHA SINHA*

Objective: *Front end & Video Chat*

Description: *This idea is based on the fact that if a teacher is sending a message to all the students & there is a possibility of mass communication then video chat comes in picture. In this way teacher or the server operands can try to send the important message to multiple signed in users.*

Details:

- *File Sharing*
  - ✓ *Shares file with multiple clients*
  - ✓ *Needs authentication from server*
  - ✓ *Login Credentials required*

- *Packet Data Loss Recognition*
  - ✓ *Packet Data Loss Measures*
  - ✓ *Keeps a check on transformation*
  - ✓ *Checks progress of data*
  - ✓ *If lost, tells percentage of data transferred*
  - ✓ *Saves repetitive sending of data*
- *Tkinter Chat Box*
  - ✓ *GUI based*
  - ✓ *Mutiple Client*

Additional Requirements:

- *Wide application of Tkinter*
- *Buffering concept understanding*

MODULE: *PRANJAL YADAV*

Objective*: Video Conferencing*

Description: *Videoconferencing (VC) is the conduct of a videoconference (also known as a video conference or video teleconference) by a set of telecommunication technologies which allow two or more locations to communicate by simultaneous two-way video and audio transmissions. Videoconferencing software is quickly becoming standard computer equipment. For example, Microsoft's NetMeeting is included in Windows 2000 and is also available for free download from the NetMeeting homepage. For personal use, free or inexpensive videoconference software and a digital cameraafford the user easy - and cheap - live connections to distant friends and family.*

Additional Features:

- *Video Conferencing*
    - ✓ *Multiparty multimedia conferencing with audio, video file sharing*
    - ✓ *Recording the video*
    - ✓ *Engage use of flask*
    - ✓ *Uses numpy*
    - ✓ *Uses open cv*
    - ✓ *Uses pip*
    - ✓ *Images are stored in multi array to make the video*
- *Uses pre-existing third-party libraries to provide common functions*
- *Database Management System*
    - ✓ *Handles sql*
    - ✓ *stores aptitude questions*
    - ✓ *calculates scores*
    - ✓ *works on xamp*
    - ✓ *creates tables*

Additional Requirements:

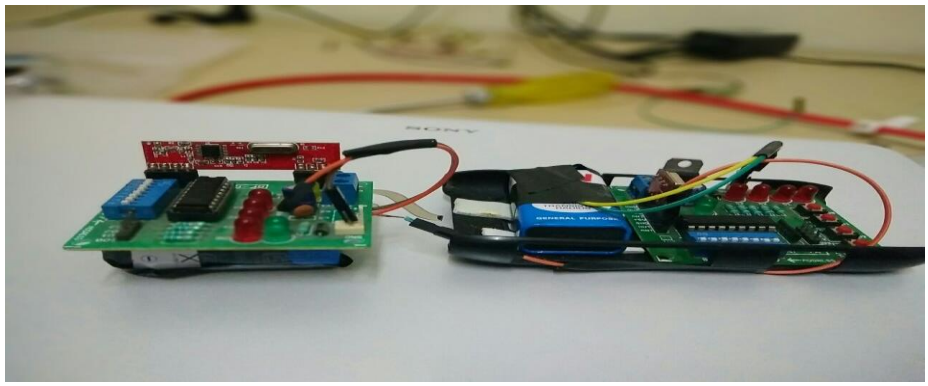
- *Flask as a micro framework of Python*
- *Buffering concept in mind*

9. Future Scope

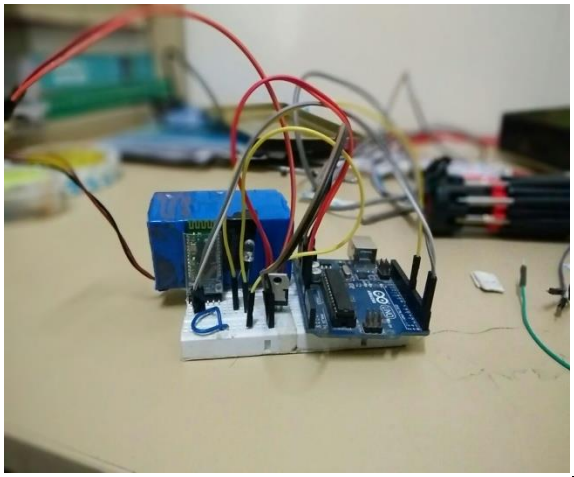
This entire project can become a trendsetter in rehabilitation organizations. It's quite user friendly, hence it can be easily deployed among the hospitals. It assures a good network between doctors and the patients. Technologies used in this project go hand in hand with today's environment. Also, special hardware features have been added to keep an eye when patients are not in direct contact with doctors. Wireless modules, mobile operated device & Android app can be of great use. In nutshell, we hope that this minor project will be able to cross academic boundaries & can practically contribute in making a healthy India.

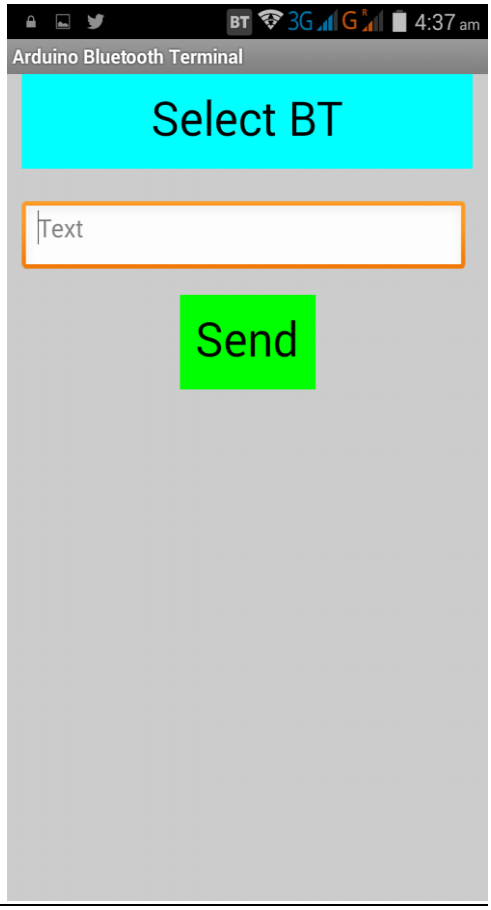
11. Testing & Snapshots

- *Transmitter & Receiver using RF Module*

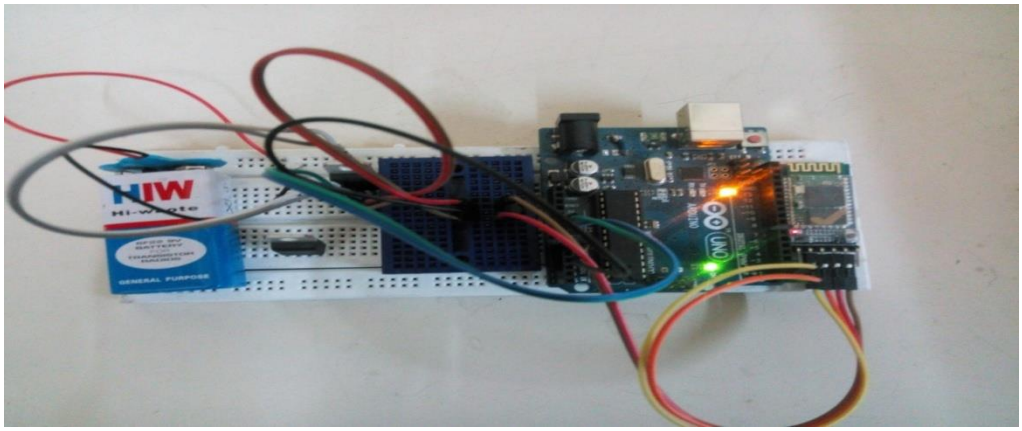


- *Mobile Operated Bluetooth Sensor*

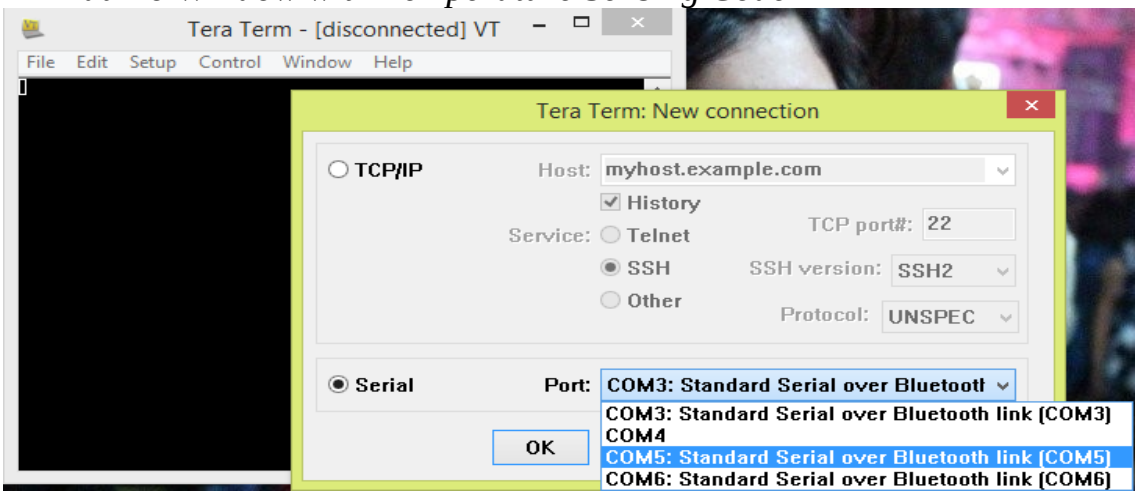




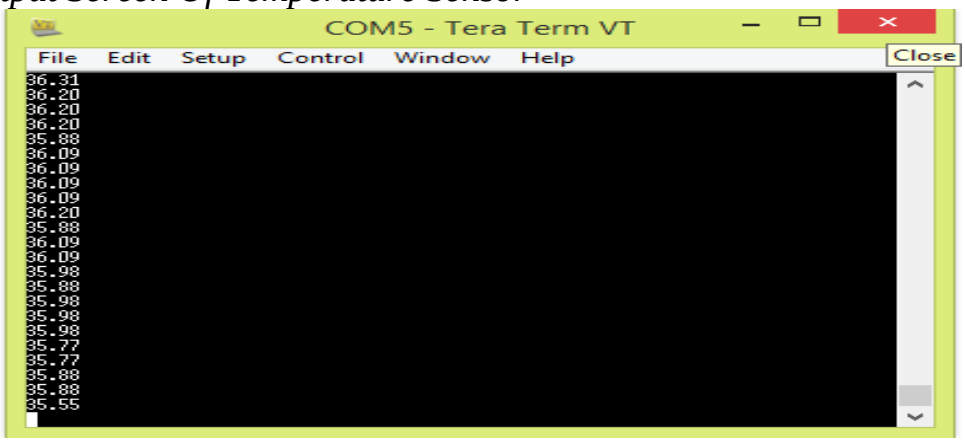
- *Arduino Set Up with Temperature Sensor & Wireless Bluetooth Module*



- *Arduino Window with Temperature Sensing Code*

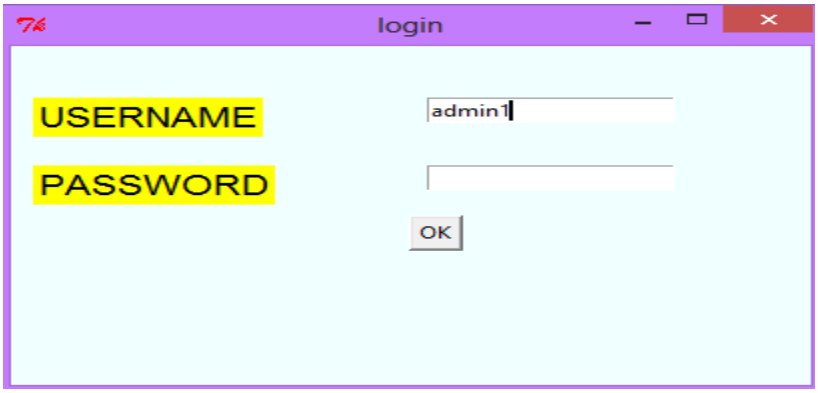


- *Output Screen Of Temperature Sensor*



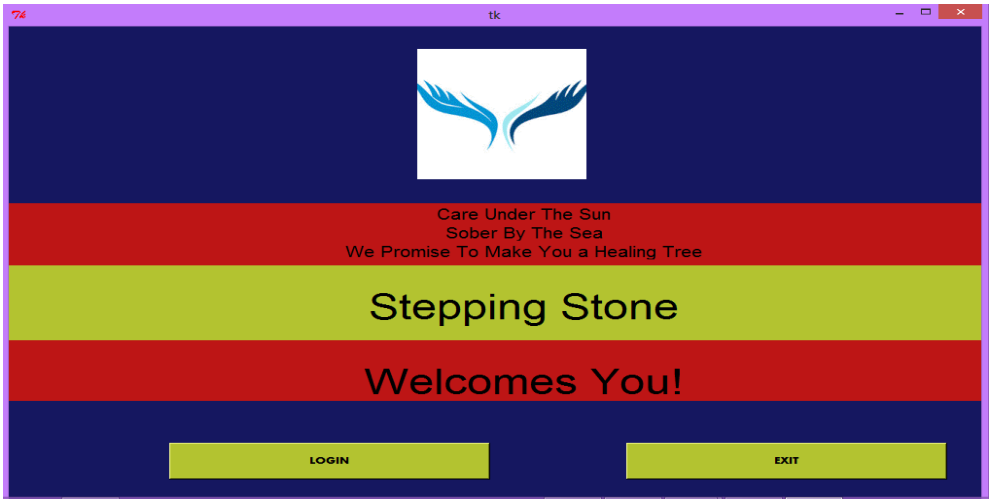
- *Welcoming Window*

- *Login Details*

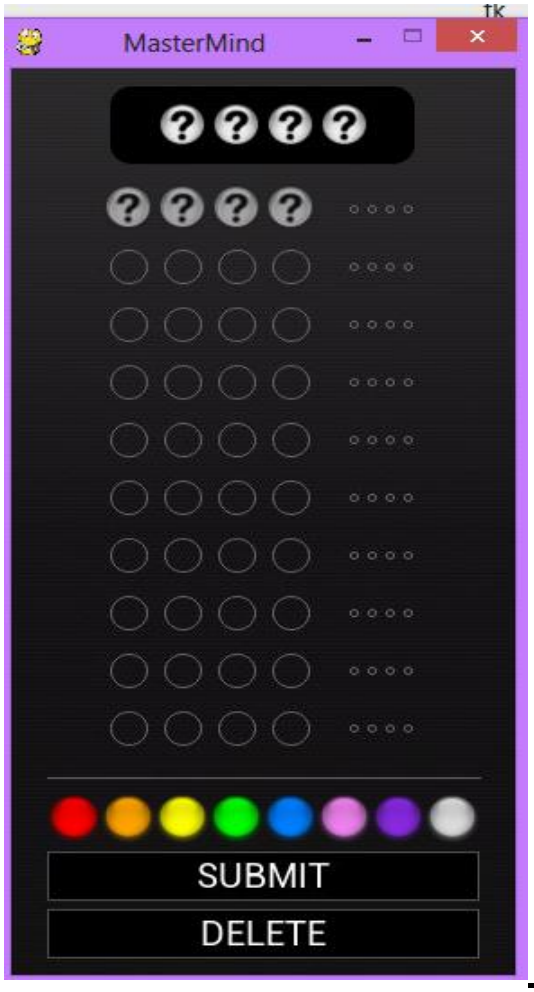


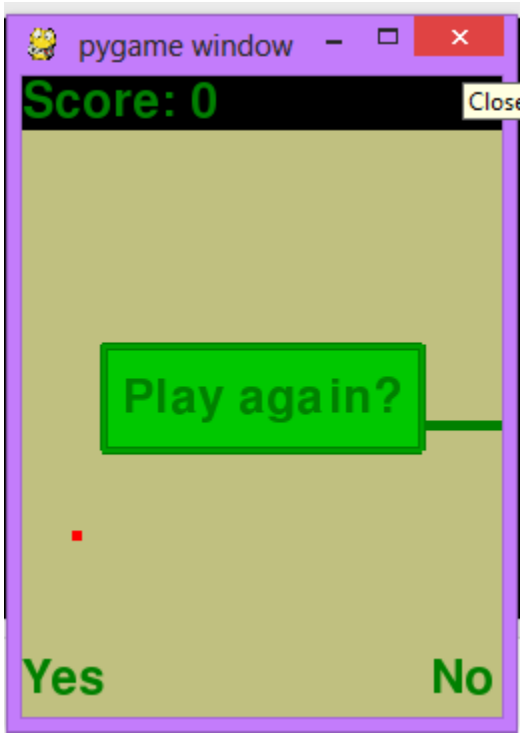
- *Pygames*

- *Game Window*

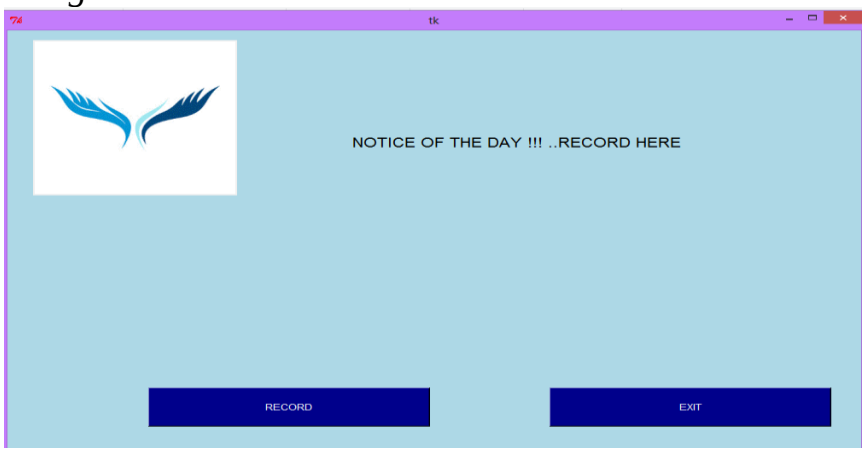


- *Pygames*

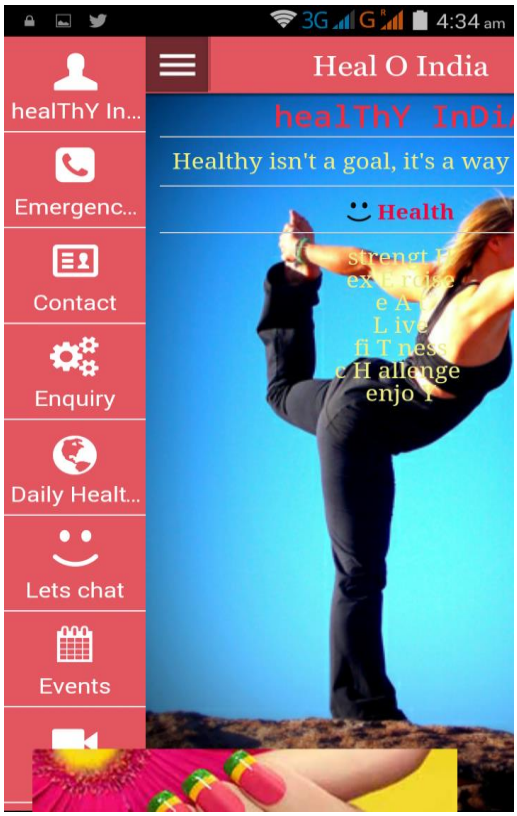


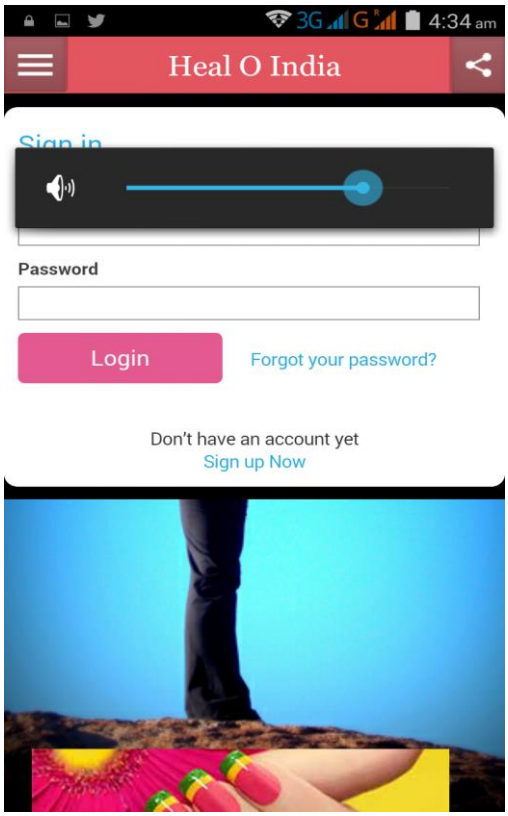


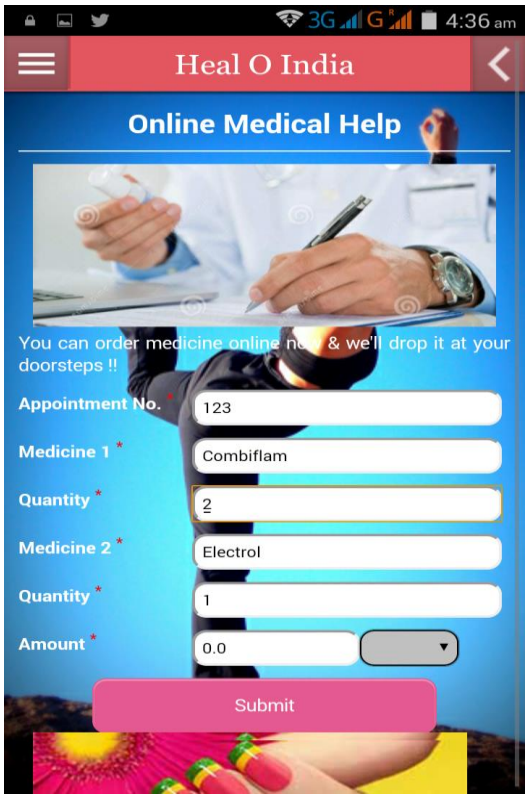
- *Record Page*

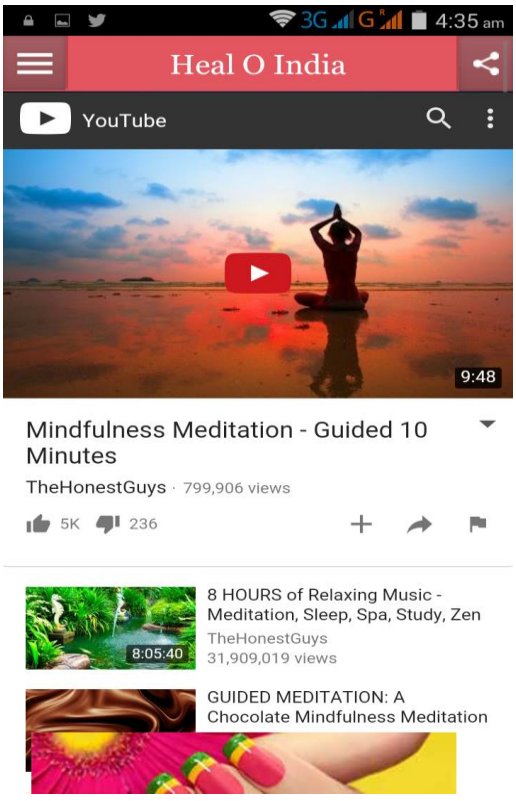


- *Android App: Heal O India*









11. References

- *http://encyclopedia.laborlawtalk.com/MYSQLfor information on MYSQL*
- *http://aspnet.4guysfromrolla.com/articles/020404-1.phpfor relationship between MYSQL and PHP.*
- *http://www.x-cart.com/articles/design_development.htmlfor online customer behavior.*
- *http://www.agilemodeling.com/artifacts/dataFlowDiagram.htmfor definition of Data Flow Diagram.*
- *http://www.informatik.uni-bremen.de/uniform/gdpa_d/methods/m- fctd.htm for definition of Functional Decomposition.*
- *http://www.python-course.eu/python_tkinter.php for basic learning of Python*
- *http://www.instrutables.com/id/ for implementation of hardware*