**Abstract**

Over the last few decades, heart-related diseases are the main reason for a huge amount of death in the world and has emerged as the most life-threatening disease, not only in India but in the whole world. There's a need for a reliable and accurate system to diagnose and treat such diseases in time for proper treatment. To automate the process of diagnosis and treatment Machine Learning algorithms and techniques is applied to the medical dataset. As technology is advancing, researchers have been using Machine Learning techniques in the diagnosis of heart-related diseases to help the healthcare industry.

Nowadays, health disease is increasing day by day due to lifestyle, hereditary. Especially, heart disease has become more common these days, i.e. life of people is at risk. Each individual has different values for Blood pressure, cholesterol and pulse rate. But according to medically proven results the normal values of Blood pressure is 120/90 (systolic/diastolic), cholesterol is and pulse rate is 72. The data set is taken from Data Mining Repository of the University of California, Irvine (UCI) (Newman et al., 1998). The system is validated using data sets from Cleveland. In those datasets, totally, 14 attributes such as Age, sex, chest pain type, resting blood pressure, serum cholesterol in mg/dl, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, ST depression, slope of the peak exercise ST segment, number of major vessels, thal and diagnosis of heart disease is presented. This project gives prediction using classification techniques like Decision trees for determining the risk level and also provides recommendation for the same.

**TABLE OF CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Title** | | **Page No.** | |
|  | List of Abbreviations | | iii | |
|  | List of Figures | | iv | |
|  | List of Tables | | v | |
| 1. | INTRODUCTION | |  | |
|  | 1.1 | Overview ………………………………………………. | | 1 |
|  | 1.2 | Problem Statement………………………………………... | | 2 |
|  | 1.3 | Aims and Objectives ……………………………………... | | 3 |
|  | 1.4 | Organization of Report …………………………………... | | 4 |
| 2. | PROJECT ANALYSIS AND DESIGN | |  | |
|  | 2.1 | Project Timeline and Distribution………………………... | | 8 |
|  | 2.2 | Development Methodology………………………………. | | 12 |
|  | 2.3 | Requirement Analysis……………………………………. | | 13 |
|  | 2.4 | Design Details……………………………………………. | | 14 |
|  | 2.5 | Proposed System…………………………………………. | | 18 |
| 3. | SYSTEM REQUIREMENTS | |  | |
|  | 3.1 | Hardware Requirement.…………………………………... | | 21 |
|  | 3.2 | Software Requirement……………………………………. | | 21 |
| 4. | IMPLEMENTATION AND SOLUTION | |  | |
|  | 4.1 | Implementation details…………………………………… | | 23 |

i

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5. | TESTING | |  | |
|  | 5.1 | Testing Approach………………………………………… | | 28 |
|  | 5.2 | Test Cases………………………………………………… | | 29 |
| 6. | RESULTS ANALYSIS | |  | |
|  | 6.1 | Results and Discussion………………………………………. | | 31 |
| 7. | CONCLUSION | |  | |
|  | 7.1 | Conclusion………………………………………………... | | 32 |
|  | 7.2 | Future Scope……………………………………………… | | 32 |
| 8 | LITERATURE REVIEW | | |  |
|  | 8.1 | Literature Survey…………………………………………. | |  |

ii

**List of Abbreviations**

|  |  |
| --- | --- |
| Cp | Chest pain type |
| Trestbp (mmHg) | Resting blood pressure |
| Chol (mg/dl) | Serum cholesterol |
| Fbs | Fasting blood sugar |
| Restecg | Resting electrocardiographic results |
| Thalach | Maximum heart rate achieved |
| Exang | Exercised induced angina |
| Oldpeak | ST depression induced by exercise relative to rest |
| Slope | The slope of the peak exercise ST segment |
| Ca | Number of major vessels (0-3) coloured by fluoroscopy |
| Num | Diagnosis of heart disease |

iii

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Title** | **Page no.** |
| 3.1.1 | Project Timeline -I | 9 |
| 3.1.2 | Project Timeline -II | 9 |
| 3.1.3 | Project Timeline -III | 10 |
| 3.1.4 | Project Timeline -IV | 10 |
| 3.2.1 | Waterfall Model | 12 |
| 3.4.1.1 | Use case diagram | 14 |
| 3.4.2.1 | DFD - 0 | 15 |
| 3.4.2.2 | DFD - 1 | 16 |
| 3.4.3.1 | Sequence Diagram | 17 |
| 3.5.1 | System – flow diagram | 18 |
| 5.1.1 | Registration and Login | 23 |
| 5.1.2 | Explaining Parameters | 24 |
| 5.1.3 | Accepting User Parameters | 25 |
| 5.1.4 | Recommendation provided by the system | 26 |
| 5.2.1 | User detail’s received at server | 27 |

iv

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Title** | **Page no.** |
| 2.1.1 | Comparing Existing and Proposed System | 7 |
| 3.1.1 | Task Distribution | 11 |
| 6.2.1 | Test Cases | 29 |

v

**CHAPTER 1**

**INTRODUCTION**

**Introduction**

This chapter gives an introduction to the project so that the idea about the overall project is understood well, it also contains the details about the problem statement and the aims and objectives of the given project.

* 1. **Overview**

One of the biggest causes of death nowadays is heart disease. Abnormal variations in blood pressure, cholesterol, pulse rate etc. are the major reasons for heart disease. The heart is one of the important functional part of the body and if it gets affected, the functions of whole body gets disturbed. In medical field, heart disease is one of the major challenges; because a lot of parameters and technicality is involved for accurately predicting this disease. According to latest survey conducted by WHO, the medical professional were able to correctly predict only 67% of heart disease, so there is a vast scope of research in area of predicating heart disease in humans. Huge amount of clinical data is available, where vital information is hidden is seldom visited and remains untapped, researchers use data mining techniques to help health care professionals in the diagnosis of heart disease. In order to predict the risk level of heart disease, Decision Tree algorithm is being used. The tree takes in 14 attributes as it’s input such as age, sex, chest pain type, resting blood pressure, cholesterol in mg/dl, fasting blood sugar, etc. Once the prediction of the risk level is performed, the precautionary suggestions would be provided to the user, which would help them to control their risk level temporarily. “How can we turn data into useful information that can enable users to make effective clinical decisions”. This is the main objective of the project. The project is dedicated for a wide scope in the field of heart related disease. Later part of the scope discusses how the user can go about understanding the various parameters related to the prediction and also the various ways through which they can control those parameters.

* 1. **Problem Statement**

The Institute of Medicine at the National Academies of Science, Engineering and Medicine [reports](http://www.nationalacademies.org/hmd/~/media/Files/Report Files/2015/Improving-Diagnosis/DiagnosticError_ReportBrief.pdf) that “diagnostic errors contribute to approximately 10 percent of patient deaths,” and also account for 6 to 17 percent of hospital complications. It is important to note that physician performance is typically not the direct cause of diagnostic errors. In fact, researchers attribute the cause of diagnostics errors to a variety of factors including:

* Inefficient collaboration and integration of health information technologies (Health IT)
* Gaps in communication among clinicians, patients and their families
* A healthcare work system which, by design, does not adequately support the diagnostic process

To address these challenges many researchers and companies are leveraging machine learning algorithms to improve medical diagnostics.

Doctors always depend on perception and experience rather than on the knowledge rich data masked in the database to take clinical decisions. However unfortunately doctors with expert knowledge in every sub-specialty are a scarce resource. The information afforded by the patients may have redundant symptoms or repetitive symptoms when patients suffer from multiple diseases that may have the same symptoms. The physicians may not be able to diagnose it accurately. Diagnosing a disease accurately is at an earlier stage is a very difficult task because of the complex interdependence of multiple factors.

Due to the abnormal variations in the functioning of heart, there are numerous heart diseases. The main aim is to make people aware about the risks of having the heart disease and suggest the precautionary measures for it. Our objective is to recommend the precautionary measures based on the risk level of the disease.

* 1. **Aim and Objectives**

**Aim:**

To make the people aware about the risk of having a heart disease.

To suggest the precautionary steps the user could take to reduce the risk.

**Objectives:**

In today’s modern and technologically advanced world the precise measurements and readings of human body can be obtained in the medical field. There are hidden patterns in the data that has been collected over years, these hidden patterns can be extracted and can be used for more precise prescription. The patterns that are present in the data collected from the repository and can be used to for prediction. The prediction made by the analyzing the patterns can be used for predicting the risk for user.

* 1. **Organization of Report**

**Chapter-1 -:** This chapter helped the team to realize our problem statement. The team did the analysis of various requirements to be met in order to complete our project.

**Chapter-2 -:** The team has conducted a literature review. The team has studied number of exiting systems. Depending on the outcome of literature survey the goals were set.

**Chapter-3 -:** The task distribution among the team members was done in this phase. Designing of various UML diagram like DFD, use case and sequence diagram was also carried out.

**Chapter-4 -:** The team realized the hardware requirements needed for the system to work. The software requirements were also realized to develop our system.

**Chapter-5 -:** In this phase the team was able to implement all the modules of the system with the help of the system flow.

**Chapter-6 -:** The team has designed various test cases and the results of those test cases were observed in order to assure that the system error free.

**Chapter-7 -:** Analysis of number of conditions was carried out to see if the system is working properly. This helps us to maintain the quality of the system.

**Chapter-8 -:** After completion of implementation of the system, team has concluded what has been achieved and what can be done in future in order to enhance the system.

**CHAPTER 2**

**REVIEW OF LITERATURE**

**Review of Literature**

A literature review is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and do not report new or original experimental work.

**2.1 Literature Surveyed**

**Human Heart Disease Prediction System using Data Mining Techniques**

T. Princy, J. Thomas intends to give details about various techniques of knowledge abstraction by using data mining methods that are being used in today's research for prediction of heart disease. In this paper, data mining methods namely, Naive Bayes, Neural network, Decision tree algorithm are analyzed on medical data sets using algorithms.[1]

**Web Analytics Support System for Prediction of Heart Disease Using Naive Bayes Weighted Approach (NBwa)**

Priyanga and Naveen mainly focus on to create a decision support system using Naive Bayes algorithm for predicting heart disease. A web application is created to get user input and the application can retrieve hidden knowledge related to heart disease from a historical database (Cleveland dataset).[2]

**Predictions in Heart Disease using Techniques of Data Mining**

M. Gandhi, S. Singh focuses was on classification methods of data mining used in data discovery. Different classification techniques of data mining have merits and demerits for data classification and knowledge extraction.[3]

**Efficient Heart Disease Prediction System using Decision Tree**

Purushottam, K. Saxena and Richa Sharma depicted the extraction of risk level from the heart disease database. The input database contains the screening of clinical data of heart patients. Initially, the database is preprocessed to make the mining process more efficient. Decision tree is used for the prediction.[4]

**Heart disease prediction using machine learning techniques: a survey**

V.V. Ramalingam, A. Dandapath, M. Raja concluded that models based on Naïve Bayes classifier were computationally very fast and have also performed well. SVM performed extremely well for most of the cases. Systems based on machine learning algorithms and techniques have been very accurate in predicting the heart-related diseases but still, there is a lot scope of research to be done on how to handle high dimensional data and overfitting.[5]

**Early detection of clinical parameters in heart disease by improved decision tree algorithm**

The paper introduced a pruning decision tree method which is a combination of pre-pruning and post-pruning that takes into account of both classification accuracy and tree size.[6]

* **Existing System**

The heart disease database contains the screening of clinical data of heart patients. Initially, the database pre-processed to make the mining process more efficient. The "num" attributes notify to the presence of heart disease in the patient. The range of this attribute is from 0 (no presence) to 4. Most of the experiments associated with Cleveland database are focused on absence (Num” value 0) and presence (“Num” values from 1 to 4). Due to personal security patient’s personal identification information replaced with dummy values. Number of Instances: Cleveland: 303 The directory contains a dataset related with heart disease diagnosis. The data was collected from the following locations: Cleveland Clinic Foundation (cleveland.data). The Cleveland database contains total 76 raw attributes, but in experiments only 14 of them is actually used. The dataset used in this experiment contains different important parameters like ECR, cholesterol, chest pain, fasting sugar, MHR (maximum heart rate) and many more.

* **Proposed System**

The heart report when generated has terms that are only understood by the doctors and according to that, they provide the necessary medications. Our project aims to provide an Android application to the users, wherein they can provide the application the above-mentioned attributes which would then be used to predict the risk level of having a heart disease. The users are then provided with an in-depth explanation of all the terms that are responsible for the risk level generated in the prediction and are recommended with various ways to reduce their risk level.

Table 2.1.1: Comparing Existing and Proposed System

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Existing System** | **Proposed System** |
| Platform | Web Based | Android |
| Focus | Prediction | Prediction as well as recommendation |

The papers studied majorly focuses on finding the efficiency of various algorithms as the predictions in the domain of medical has to be accurate. While none of them provides recommendation to the user the proposed system uses an android application to provide the same.

**CHAPTER 3**

**PROJECT ANALYSIS AND DESIGN**

**Project Analysis and Design**

This chapter should give detail design of the project. It includes Block diagram of proposed system and UML diagrams (Use case diagram, Data flow diagram, Sequence diagram etc.) as applicable to the project.

**3.1 Project Timeline and Distribution**

Gantt chart:

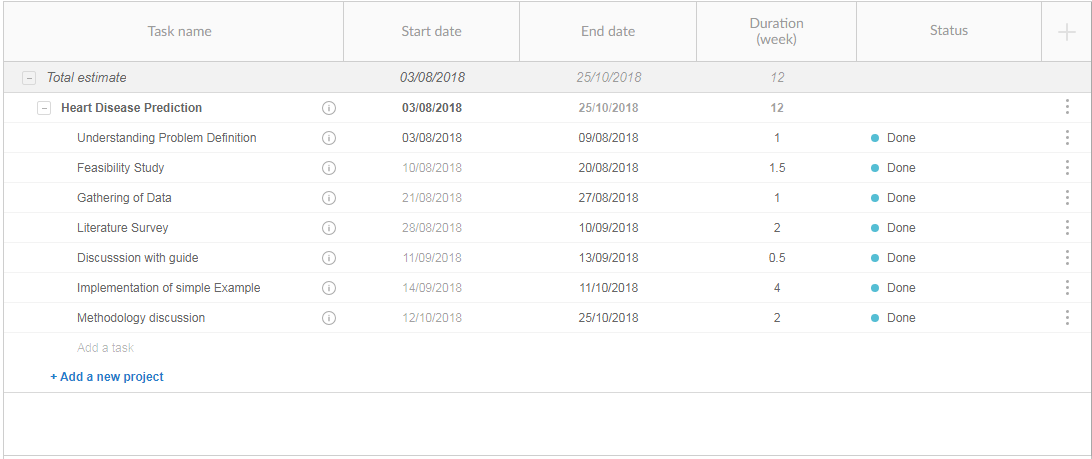
A Gantt chart is a type of bar chart, developed by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project.

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity.

This allows you to see at a glance:

* What the various activities are.
* When each activity begins and ends.
* How long each activity is scheduled to last.

Below images and tables are the detailed design of the project including project timelines and task distribution.

  
Fig 3.1.1: Project Timeline -I

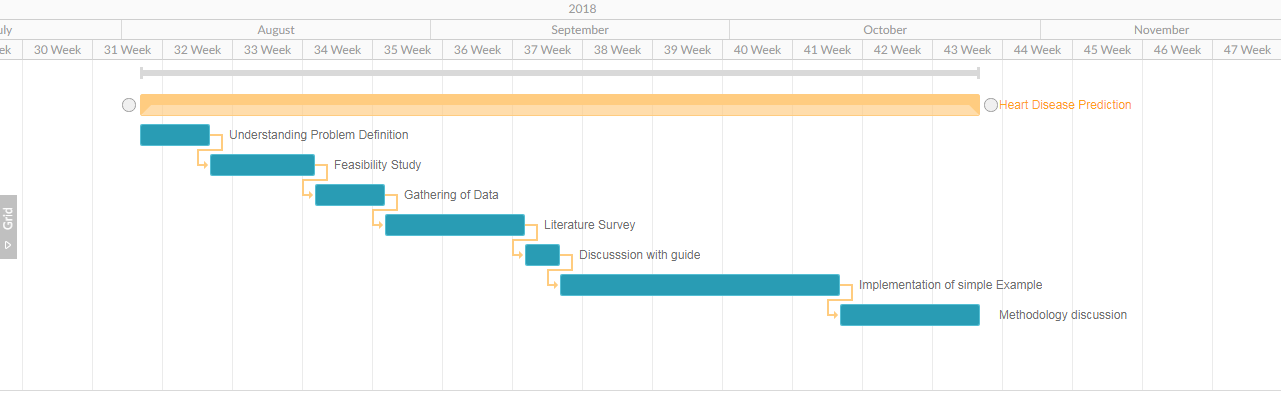


Fig 3.1.2: Project Timeline -II

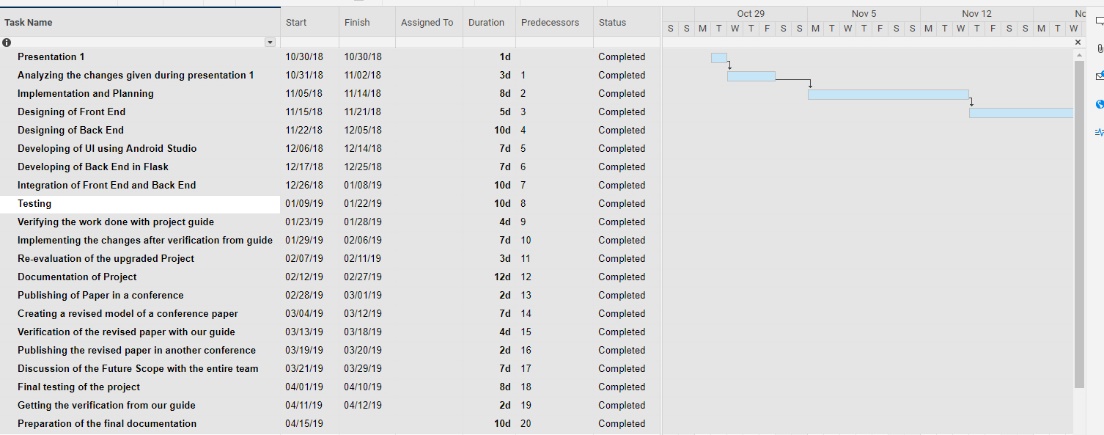
Fig 3.1.3: Project Timeline -III



Fig 3.1.4: Project Timeline -IV

Table 3.1.1: Task Distribution

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Assigned to** | **Status** |
| Selection of Topic | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |
| Literature Review | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |
| Introduction and Motivation | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |
| Requirement Gathering | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |
| Architecture Design | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |
| Design of Product | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |
| Testing | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |
| Final Documentation | Jay Jethwa  Akshit Modi  Maunish Shah | Completed |

## **3.2 Development Methodology**

## Description: Description: SDLC Waterfall ModelThis section describes the project as per the various stages of the Software Development life cycle. The model of software development life cycle used in this project is the waterfall method. The Waterfall Method is comprised of a series of very definite phases, each one run intended to be started sequentially only after the last has been completed, with one or more tangible deliverables produced at the end of each phase of the waterfall method of SDLC. Essentially, it starts with a heavy, documented, requirements planning phase that outlines all the requirements for the project, followed by sequential phases of design, coding, test-casing, optional documentation, verification (alpha-testing), validation (beta-testing), and finally deployment/release.

Fig 3.2.1: Waterfall Model

**3.3 Requirement Analysis**

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project Management.

* **Functional Requirement:**
  + User should be able have a simple android phone with this application installed on the system.
  + User just have to open the application and start detecting the presence of risk or not
  + User would then be provided with the recommendations if any.
  + System should be economical viable.
  + System should be adaptive and analysis can be performed.
* **Non-Functional Requirement:**
  + The system should produce on click result.
  + The system should be adaptive.
  + The system should be able to handle load on server when the number of user increases.

**3.4 Design Details**

**3.4.1 Use Case Diagram**

A use case diagram at its simplest is a representation of a user’s interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. A use-case diagram can help provide a higher-level view of the system.

User

Fig 3.4.1.2: Use Case Diagram

**3.4.2 Data Flow Diagram**

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

User

Dataset

Fig 3.4.2.1: Level 0 DFD

Registers into the app

Logs in the app

Enters the Parameters

User entered data sent to the server

Server runs the algorithm and predict risk

Recommendations are suggested on risk

Predicted risk and recommendations are sent to user

Algorithm is trained

Dataset

Fig 3.4.2.2: Level 1 DFD

**3.4.3 Sequence Diagram**

A sequence diagram in a Unified Modeling Language (UML) is a kind of 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. A sequence diagram represents 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.

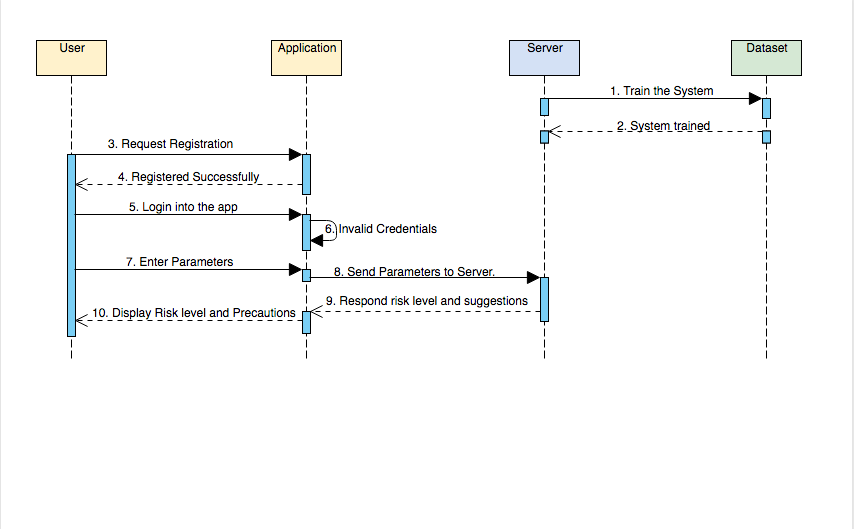


Fig 3.4.3.1: Sequence Diagram

A Sequence diagram represents the sequence of actions that occurs in system. A Sequence diagrams are read left to right and descending. A sequence diagram is 2-Dimensional in nature. On the horizontal axis, it shows the life of the object that it represents, while on the vertical axis, it shows the sequence of the creation or invocation of these objects.

**3.5 Proposed System**

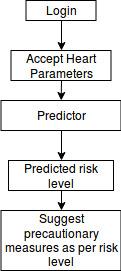
****

Fig 3.5.1: System-flow Diagram

The Proposed system consist of following modules:

* Login
* Fetching of Data
* Risk Predictor
* Precautionary Recommendation

The flow of the project is as follows:

* The user would first login into the system, verifying which the system would redirect the user to the home page through which the user could navigate the app.
* The user will enter in the app the parameters asked regarding to his heart in the fields provided which would then sent to the server.
* The user would have to enter 13 parameters to get the risk predicted of him of getting affected by a heart disease in future.
* The details entered by the user would then served as an input to the ‘Predictor’.
* The Predictor would then predict the risk level for the user. The risk levels would range from 0 to 4(0 - No risk; 4 - Maximum risk)
* Along with the precautionary suggestions that the user would apply on his level to reduce the level of risk. The precautionary measures would be suggested based on the risk level predicted for the user.
* **Methodologies and Algorithm**

Decision Tree - Decision Trees being a non-parametric supervised learning method they are used for [classification](https://scikit-learn.org/stable/modules/tree.html" \l "tree-classification). The goal is to create a model that predicts the value of a target variable (risk level) by learning simple decision rules inferred from the data features (Dataset). The major challenge in the decision tree is to identify the attribute for the root node in each level. This process is known as attribute selection. One of the main advantages of using decision tree was that it is able to handle both numerical and categorical data. Other techniques are usually specialized in analyzing datasets that have only one type of variable.

The decision tree learning algorithm learns recursively as follows:

Steps:

1.Compute the Gini index for data-set

2.For every attribute/feature:

1.Calculate Gini index for all categorical values

2.Take average information entropy for the current attribute

3.Calculate the Gini gain

3. Pick the best Gini gain attribute.

4. Repeat until we get the desired tree.

Machine learning library Scikit-learn is used for prediction. Scikit-learn has a range of supervised and unsupervised learning algorithms provided through a consistent interface in Python. The sklearn.tree module includes decision tree-based models for classification.

class sklearn.tree.DecisionTreeClassifier(criterion=’gini’,splitter=’best’, max\_depth=None, min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_features=None, random\_state=None, max\_leaf\_nodes=None)

The method uses gini as the splitting criterion for attribute selection. Gini provides a statistical measure of the degree of variation or inequality represented in the dataset.Gini index - Given a training set S, the target attribute takes on k different values (i.e. classes), the Gini index of S is deﬁned as

n

G(S) =  Σ P(i) \* (1 - P(i))

i=1  
Where P(i) is the probability of a certain classification i, per the training data set.

Project validity and feasibility are tested in the analysis phase where all details about the project are gathered and a formal report is formed which can be used as a future reference during the further project activities.

After the analysis is done, we move on to designing the project which is done using different UML diagrams which can be Sequence diagrams, use-case diagram and many others.

**CHAPTER 4**

**SYSTEM REQUIREMENTS**

**System Requirements**

The moto of this chapter is to identify the platform needed to run the proposed system. Team will

study the hardware as well as software requirements that will help in order to develop the system.

**4.1** **HARDWARE REQUIREMENTS**

Android Smartphones with following specifications.

* Processor 1.3GHz Quad core.
* RAM 512MB and above.
* Storage 8GB (min.).
* Android version 4.4 (KitKat) and above.

**4.2 SOFTWARE REQUIREMENTS**

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a mobile or a computer device to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

**1. Android**

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software. Patterned after the Linux kernel, the Android also was released as open source code. Development for the Android may be done through Windows, Linux or Mac. Although primarily written in Java, there is no Java Development Machine (JDM) in the platform.

**2. Android Studio**

Android Studio is the official integrated development environment ([IDE](https://searchsoftwarequality.techtarget.com/definition/integrated-development-environment)) for Android application development. It is based on the [IntelliJ IDEA](https://www.theserverside.com/definition/IntellJ-IDEA), a [Java](https://www.theserverside.com/definition/Java) integrated development environment for software, and incorporates its code editing and developer tools. It uses an Instant Push feature to push code and resource changes to a running application. A code editor assists the developer with writing code and offering code completion, refraction, and analysis.

**3. Python**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

**4. Flask**

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

According to above mentioned hardware configuration we require an Android device with an internet connection to run the application to send and receive data to and from server.

**CHAPTER 5**

**IMPLEMENTATION AND SOLUTION**

**Implementation and Solution**

This chapter contains the detailed explanation of all the modules and subordinate modules of the application.

**5.1 UI Modules**

**5.1.1 Registration and Login**

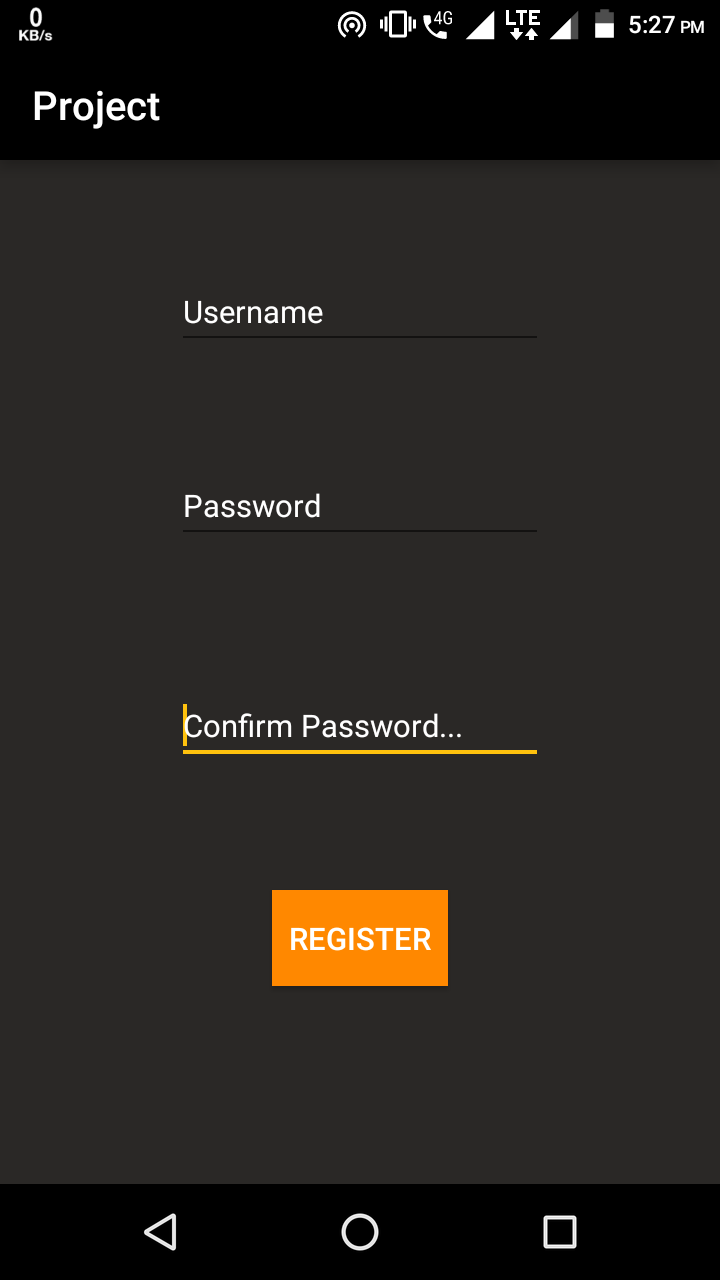
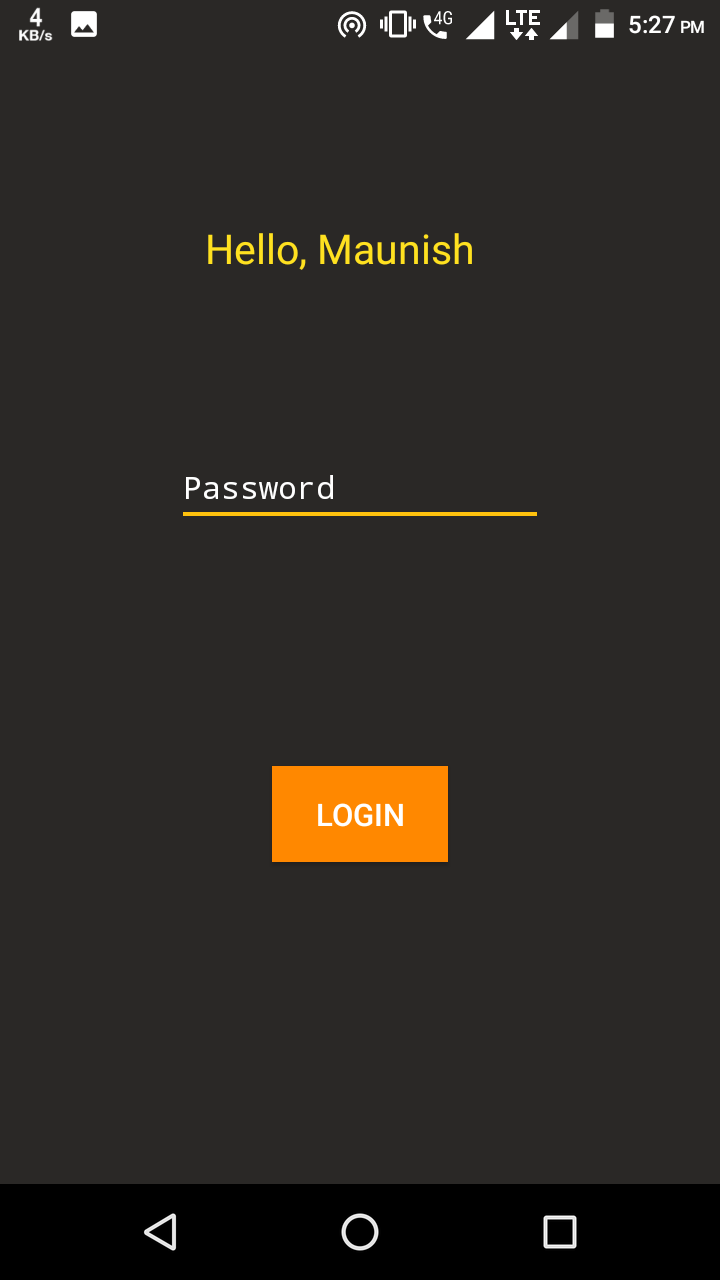
****

Fig 5.1.1: Registration and Login

The above diagram displays the login and registration activities of the project. First time users are made to register on the app which requires them to enter their username and password. Once the registration is complete the user is made to log in to the app.

**5.1.2 Explaining Parameters**

The various parameters that are to be entered by the user are explained in this activity. What each parameter is and what values are accepted for each of those parameters is described over here.

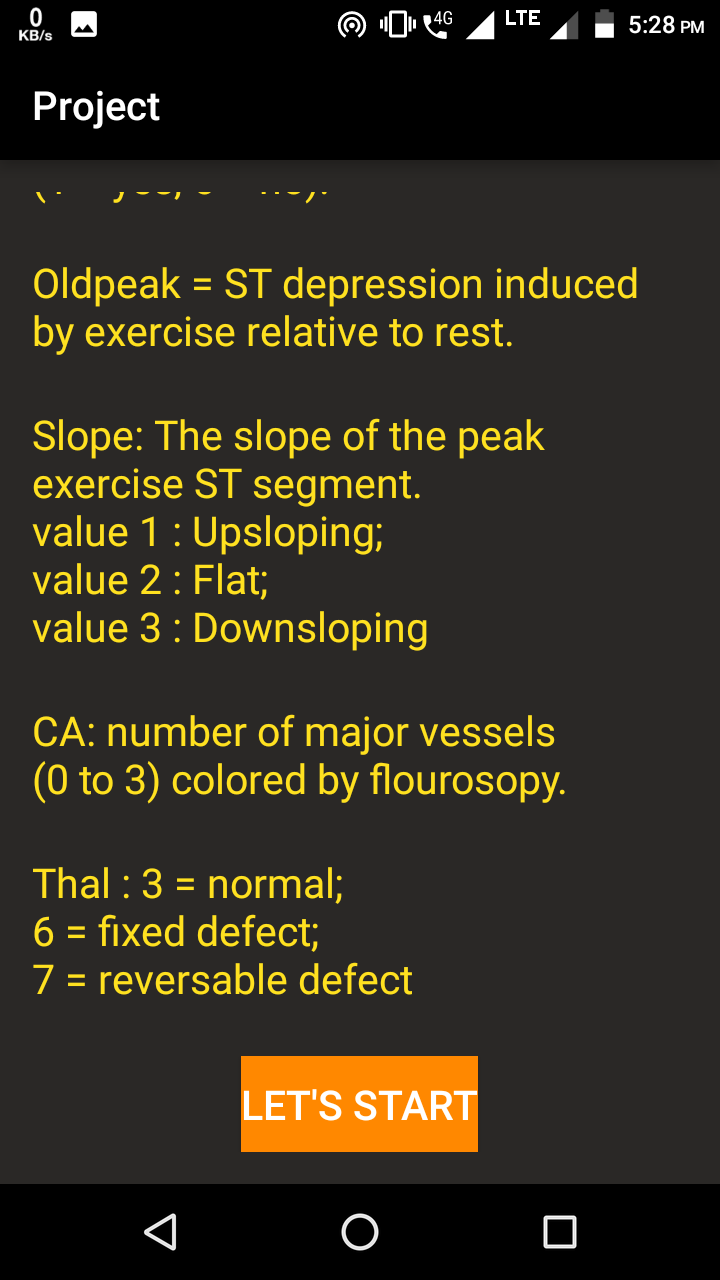
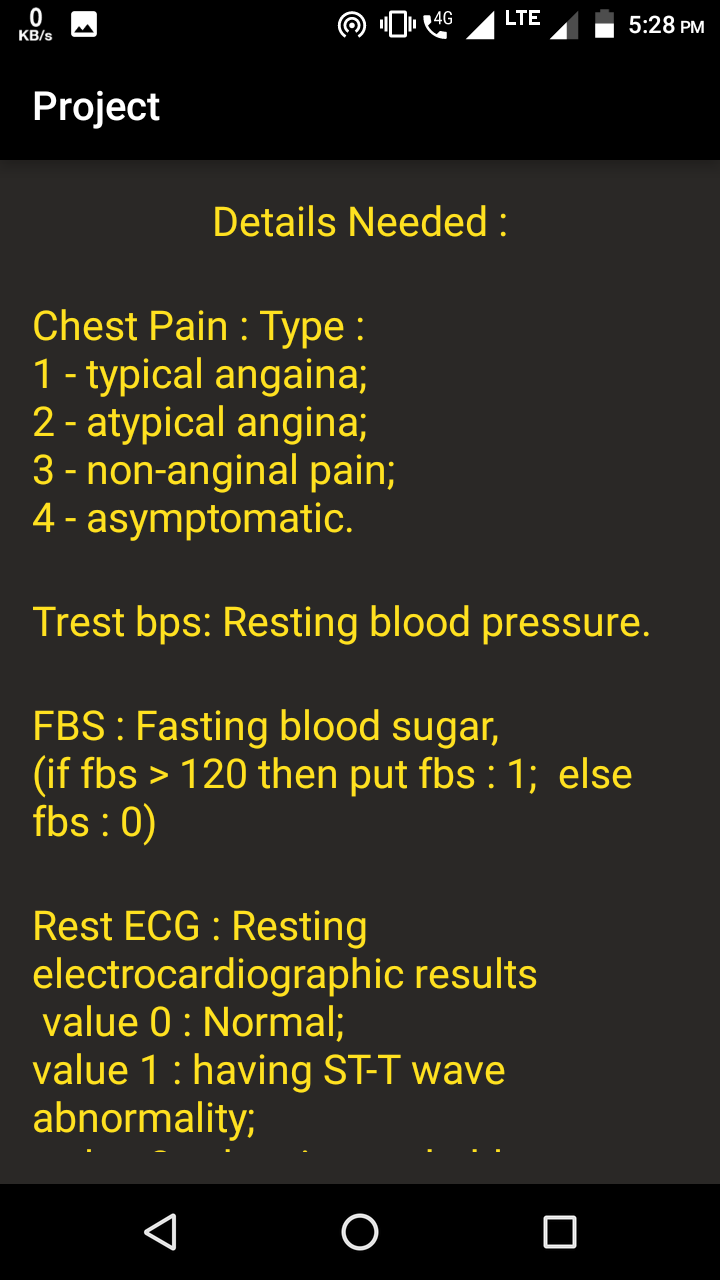
****

Fig 5.1.2: Explaining Parameters

**5.1.2 Accepting Parameters**

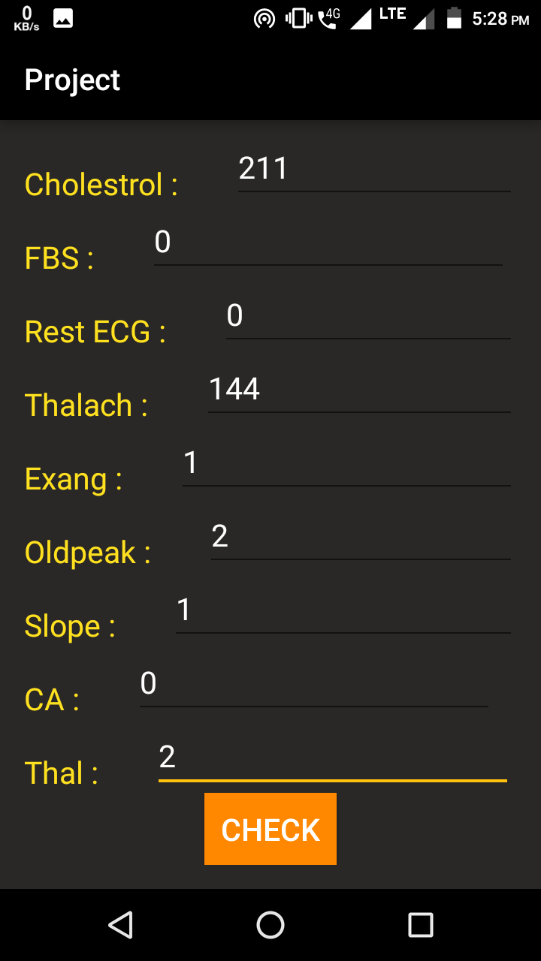
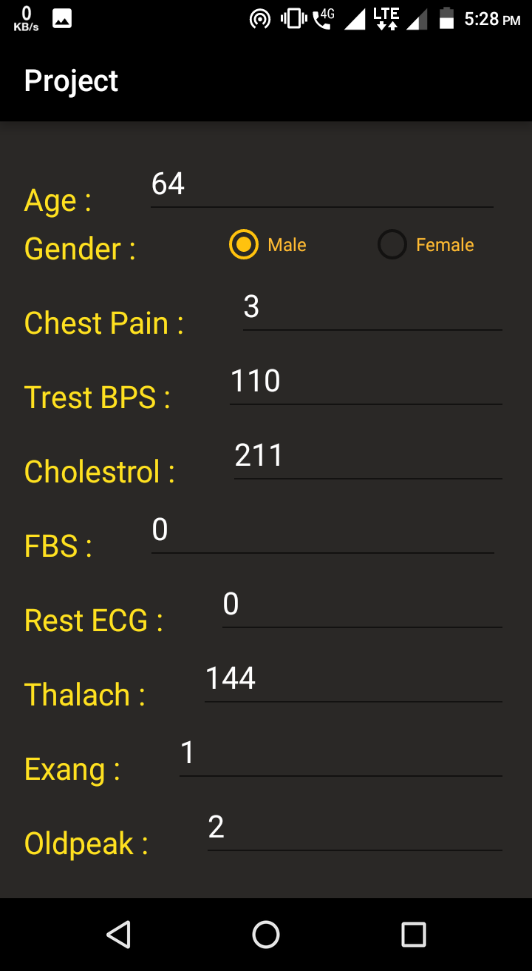
****

Fig 5.1.3: Accepting user parameters

The above screenshots display the UI part of the application which include several different parameters. The value of the 14 parameters are used by the algorithm to predict, whether the user has a risk of heart disease or not in the form of 1 and 0 respectively.

**5.1.3 Providing Recommendation**

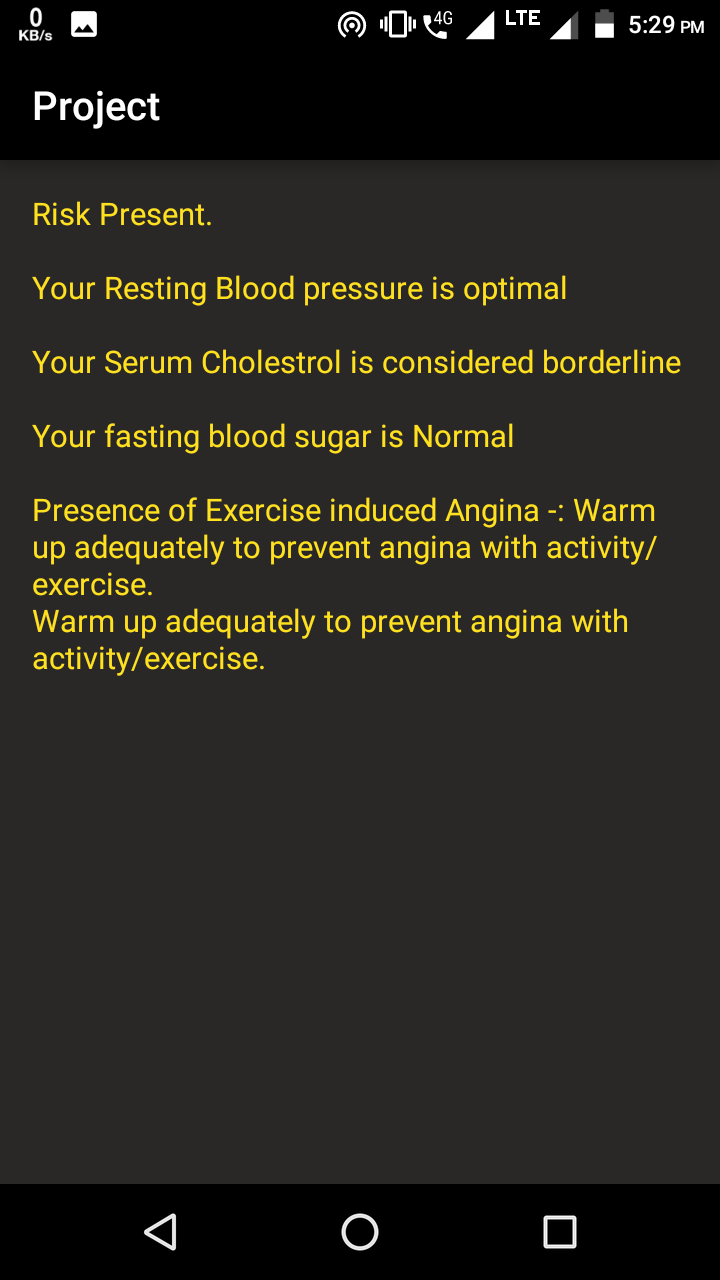


Fig 5.1.4:Recommendation provided by the system

Fig. shows the Recommendation provided to the user by the application. These recommendations are suggested depending upon which parameter’s value has surpassed the threshold limit. Based on that, the user must perform those actions which will help him bring down his parameter’s value within the normal range.

**5.2 Backend Server**

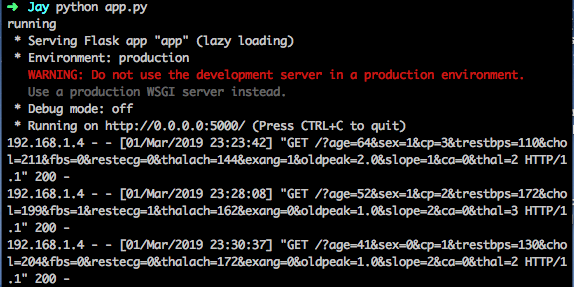


Fig 5.2.1: User details received by server

The screenshot attached is the server side of the application. This shows how the server accepts the value of the 14 parameters by the user, the how our algorithm i.e. decision tree uses those values to predict the risk level.

The major module of the proposed system is training and learning module. The training module is used to train the algorithm which is then used to predict the risk level of users. The figures mentioned above in this chapter provides an understanding about how the parameters entered by the user are sent to the server and how the server provides the presence of risk level and recommendations to the user.

**CHAPTER 6**

**TESTING**

**Testing**

This chapter gives information about the test results of the Application and Admin panel.

## **Testing Approach**

Details of white box testing / black box testing / Unit testing / Integration testing / Integration testing

1. **Black box testing**: Internal system design is not considered in this type of testing. Tests are based on requirements and functionality. The number of modules and no of python files required for each module is checked.
2. **White box testing**: This testing is based on knowledge of the internal logic of an applications code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions. All the modules are tested for their logic whether it functions properly or not. Code is checked by inserting different inputs to check its functionality.
3. **Unit testing**: Testing of individual software components or modules. Each module was run separately to check the output.
4. **Integration testing**: Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. Each module was checked to see whether data flows properly from one module to another properly. All the modules were tested to see if entire system is executed or not.

## **6.2 Test Cases**

Table 6.2.1: Test Cases

|  |  |  |
| --- | --- | --- |
| **Test Steps** | **Results** | **Status** |
| **Registration page** | User gets successful Registration message once he enters the necessary credentials. | Pass |
| **Login page** | User gets successfully logged in once the login id and password get verified. | Pass |
| User’s authentication fails if the login id and password do not match. | Fail |
| **Accepting Parameters** | The user enters his heart related 14 parameters into the application which is accepted by the algorithm. | Pass |
| **Risk Level Prediction** | User gets their risk of heart disease predicted based upon the values of parameters entered by them. The user has presence of heart disease | Pass |
| Sometimes the user does not have presence of heart disease | Fail |

|  |  |  |
| --- | --- | --- |
| **Test Steps** | **Results** | **Status** |
| **Providing Recommendations** | The application provides recommendation to the user based upon the values that have surpassed their threshold limit. | Pass |
| The application will not provide recommendation to the user based upon the values that have not surpassed their threshold limit. | Fail |

The generated test cases were tested against the proposed system giving the results as shown in the above table. Test cases like user registration, login etc. were generated and tested against the proposed system giving a pass/fail status for each test.

**CHAPTER 7**

**RESULT ANALYSIS**

**Result Analysis**

The testing which is performed in the above section verifies various modules independently to test the working of each of the same. The Registration page consists of a username and password field to get registered with the app. Next the Login page involves the user to enter the password so that they can start predicting the risk of having heart disease. The next page explains and provide a brief description of the various parameters that are needed from the user. What do the parameters mean and what all values are accepted for each parameter is explained. The user is then provided with a page wherein they can enter the various values of the parameters, as soon as they click on the check button the result of the prediction will be provides along with the recommendations for those parameters that exceeds the normal range of safety.

The successful testing of each module independently proves to be useful when integrated as a system. The android application including all modules in collaboration provides an excellent tool for predicting the risk of having heart disease.

**CHAPTER 8**

**CONCLUSION**

**Conclusion**

**8.1 Conclusion**

To make effective clinical decisions in the medical field, decision tree plays a vital role as there is a need for a reliable and accurate system. The project studies how the machine learning and decision tree algorithm works can be used for predicting the heart disease by taking into consideration certain parameters such as thal, cholesterol, heart rate, age, gender. Using these parameters, the algorithm predicts the risk level of heart disease between the range 0 - 1 where 0 depicts no risk and 1 depicts the presence of risk. Based on the risk level predicted, the user is suggested some precautionary measures in order to lower the risk and stay fit.

**8.2 Future Work**

Currently, the system is able to predict the risk level of the user and provides the recommendation based on the instantaneous values provided. In future, the system can be enhanced, where the application will be able to provide smart recommendations based on the user's history. When the user checks for his risk level for two or three times with varying parameters, the application will store that data in a database. So, in future the same user enters his parameters, the system will compare the current values with the existing values in the database. Based on those values, smart/customized recommendations will be provided. This application can also be further used for predicting the risk level of various other diseases and not just confined to heart-related disease.

**Bibliography**

**Papers**

1. T. Princy, J. Thomas, “Human Heart Disease Prediction System using Data Mining Techniques,” International Conference on Circuit, Power and Computing Technologies [ICCPCT], 5090-1277, 2016.
2. Priyanga, Naveen, “Web Analytics Support System for Prediction of Heart Disease Using Naive Bayes Weighted Approach (NBwa),” Asia Modelling Symposium (AMS), 2376-1172, 2017.
3. M. Gandhi, S. Singh, “Predictions in Heart Disease using Techniques of Data Mining,” 1st International Conference on Futuristic trend in Computational Analysis and Knowledge Management, 4799-8433, pp. 520-525, June 2015.
4. Purushottam, K. Saxena, R. Sharma, “Efficient Heart Disease Prediction System using Decision Tree”, 4799-8890, International Conference on Computing, Communication and Automation (ICCCA), pp. 72-77, July 2015.
5. V.V. Ramalingam, A. Dandapath, M. Raja, "Heart disease prediction using machine learning techniques: a survey," International Journal of Engineering & Technology [IJET], pp. 684-687, 2018.
6. A. Mahmood, M. Kuppa, “Early detection of clinical parameters in heart disease by improved decision tree algorithm,” Second Vaagdevi International Conference on Information Technology for Real World Problems, 2010.

**PUBLICATION AND CERTIFICATES**

1) Harsh Namdeo Bhor, Maunish Shah, Jay Jethwa, Akshit Modi, “Heart Disease Prediction and Recommendation”, *International Research Journal of Engineering and Technology*, Vol. 06., Issue 04., April 2019.

2) Maunish Shah, Jay Jethwa, Akshit Modi participated in National Level Project Competition, “KJSIEIT In-Tech”