A PROJECT ON

DISEASE PREDICTION AND DETECTION SYSTEM

SUBMITTED IN PARTIAL FULFILMENT OF

REQUIREMENT FOR THE AWARD OF THE DEGREE

BACHELOR OF COMPUTER APPLICATION (BCA)

OF

KANNUR UNIVERSITY

SUBMITTED BY
JAMES DOMINIC (NJ18BCAR08)
SHAMAL N S (NJ18BCAR09)
DOMINIC JOSE (NJ18BCAR21)

NAVAJYOTHI COLLEGE

CHERUPUZHA, CHUNDA – 670511

2020-2021

KANNUR UNIVERSITY

NAVAJYOTHI COLLEGE

CHERUPUZHA, CHUNDA – 670511

CERTIFICATE

This is to certify that the project entitled **“DISEASE PREDICTION AND DETECTION SYSTEM”** submitted in partial fulfilment of the requirement of the award of BCA degree is a result of bonafide work carried out by **JAMES DOMINIC (NJ18BCAR08), SHAMAL N S (NJ18BCAR09) and DOMINIC JOSE (NJ18BCAR21)** during the year of 2020-2021.

Internal guide:
Mrs Aney Suphalamma N Mathew

Head of Department

External examiners:

1)

2

DECLARATION

We James Dominic, Shamal NS and Dominic Jose final year BCA students of Navajyothi College under Kannur University do hereby declare that the project entitled "**DISEASE PREDICTION AND DETECTION SYSTEM**" is an original work carried out by us under the supervision and guidance, towards the partial fulfilment of BCA Degree and that no part thereof has been presented for the award of any other degree.

Place: Cherupuzha

Date:

James Dominic
Shamal N S
Dominic Jose

ACKNOWLEDGEMENT

First all of we thank the Lord Almighty for his immense grace and blessing showered on us at every stage of this work.

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all effort with success.

We grateful to our project guide Mrs Aney Suphalamma N Mathew for the guidance, inspiration and constructive suggestions that helped us in the preparation of this project.

We also thankful to our colleagues who have helped us in the successful completion of the project.

CONTENTS

Chapters

Chapter 1

INTRODUCTION

1.2.1 INTRODUCTION

This is an android application using flutter for the disease prediction and detection. By this project the building of an accurate and easy to use disease prediction and detection system utilizes the concepts that could help patients to a great extend. By using relevant training data collected from multiple anonymized lab test results and public datasets that could accurately predict various diseases even at an early stage. This application also help doctors to identify the disease faster, and can also reduce the human error factor . we are trying to detect various diseases that could be identified from different types of lab test results like Common Blood Count (CBC) test, Thyroid and Diabetic test which are some of the common tests conducted by people.

Modules:

- Admin
- User

Chapter 2

SYSTEM ANALYSIS

2.1 DESCRIPTION OF ANALYSIS

System analysis is a process of collecting factual data, understand the process involved, identifying problems and recommending feasible suggestions for improving the system functioning. It defines the requirement of the system, independent of how these requirements will be accomplished. The deliverable result at the end of this phase is a requirement document.

System analysis is an iterative process that continues until a preferred and acceptable solution emerges.

LIMITATIONS OF EXISTING SYSTEM

- Only applicable to some diseases
- Regular update is not available

2.1.2 PROPOSED SYSTEM

The existing problem is the absence of a frequently accessible system that utilizes the capacity of numerous and abundant clinical and laboratory test data that has the potential to help common people and health care professionals in diagnosing and predicting various diseases that can be inferred from those results. The aim of this project is to implement one such system that is easily accessible and accurate.

There are numerous types of tests conducted in laboratories. Currently, the focus is on blood tests and among the various blood tests available, we are limiting the work to accommodate only disease diagnosis using CBC Thyroid and Diabetes test. For each of these tests, different datasets and machine learning algorithms are used. The first step was to collect relevant data in each and every aspect towards the goal of the system. As stated above, there are different types of blood tests and each of them measure different parameters. We were able to find a few resources on the internet that provide public datasets on clinical tests that matched our requirement. Using those resources, we created the required dataset for the training of ML models.

2.2 FEASIBILTY STUDY

Feasibility study is the analysis of a problem to determine if it can be solved effectively. The operational (will it work?), economical (cost and benefits) and technical (can it be build?) aspects are part of the study.

2.2.1 ECONOMICAL FEASIBILITY

The developing system must be justified by cost and benefit. A criterion is to ensure that effort taken on the project give the best return at the earliest. One of the factors, which affect the development and the economical questions during the preliminary investigation are verified to estimate the following:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.
- The benefits in the form of reduced cost.
- This feasibility checks whether the system can be developed with the available funds.

2.2.2 TECHNICAL FEASIBILITY

This study carried out to check the technical facilities; those are the technical requirement of the system. This evaluation determine whether the technology needed for the proposed system is available or not.

- Can the work for the project be done with current equipment existing software technology & available personal?
- Can the system be upgraded if developed?
- If new technology is needed then what can be developed?

Chapter 3

SYSTEM SPECIFICATION

3.1 SYSTEM ENVIRONMENT

Here shows the target environment (hardware and software) for which the software is being developed.

3.1.1 Hardware Specification

Processor	: Intel core i5 8th Gen.
RAM	: 8GB
Hard Disc	: 500 GB

3.1.2 Software Specification

Operating System	: windows 10
Front end tool	: Dart
Back End Tool	: Python Using Flask Framework
Server	: Heroku
IDE	: Android Studio

3.2 TECHNOLOGY USED:

DART

The Frond end of the Android application is written in Dart language using the Flutter framework from Google. Dart is an open-source, general-purpose, object-oriented, class based, garbage-collected programming language with C-style syntax. It is developed by Google in 2011 and later approved by standard by ECMA. The project was founded by Lars Bak and Kasper Lund. The purpose of Dart programming is to create a frontend user interfaces for the web and mobile apps. It is also used in server side.

It is under active development, compiled to native machine code for building mobile apps, inspired by other programming languages such as Java, JavaScript, C#, and is Strongly typed. Dart is a compiled language and we cannot execute the code directly. The compiler parses it and transfer it into machine code. Google has introduced Flutter for native mobile app development in both Android and ios. Flutter is a mobile app SDK, with framework, widgets, and tools from a single codebase. Flutter is a free open source platform and work with Firebase and other mobile app SDKs.

PYTHON

The functionality of the backend of the application is to accept REST API calls which contains the necessary parameters that are to be fed into the trained (and saved) model for it to make prediction or diagnosis. Each model we discussed above is reference by an API endpoint. Basic structure of the API is given below.

```
endpoint  :  /predict-TESTNAME
{
    “data”  :  {
           “parameter1”  :  value,
             …
            “parameterN” :  value
      },
      “token”  :  API-TOKEN
 }
```

The APIs are written is Python using Flask framework. Flask is a micro web framework written in Python which is used to write server programs. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python was developed by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress. The server accepts POST requests to the API end points, gets the parameters passed to it, feeds it to the appropriate model, gets the diagnosis result and returns it back to the client as HTTP response.

Chapter 4

SYSTEM DESIGN

4.1 SYSTEM DESIGN

Designing is the most important phase of software development. The purpose of the design phase is to solution of the problem specified in the requirements document. This is the first step in moving from the problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases is the Design Document. This document is similar to a blueprint or a plan for the solution and is used later during implementation, testing and maintenance.

The goal of the design process is to produce a model of a system, which can be used later to build that system. The produced model is called design of the system.

4.2 MODULES DESCRIPTION

This project consists of the following modules;

- Admin
- User

Admin

The admin is the person who controls every data and Procedure.

User

The user may be the Patient or a person seeking information about various diseases with original result in his hand and the system detects and diagnoise the results.

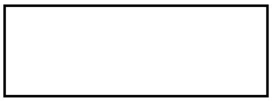
4.3 LOGICAL DESIGN

4.3.1 Data Flow Diagram

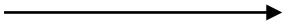
A data flow diagram is a graphical representation that depicts information flow and transforms that are applied as data move from input to output.

The successive expansion of a DFD from the context diagram to those giving more details is known as levelling of DFD. Thus a top down approach is used, starting with an overview and then working out the details. The main merit of DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what files are used, and where the results flow.

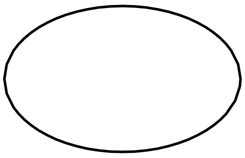
Basic DFD symbols:



Represents source or destination of data

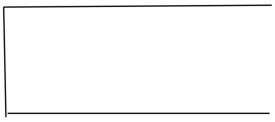


Represents data flow



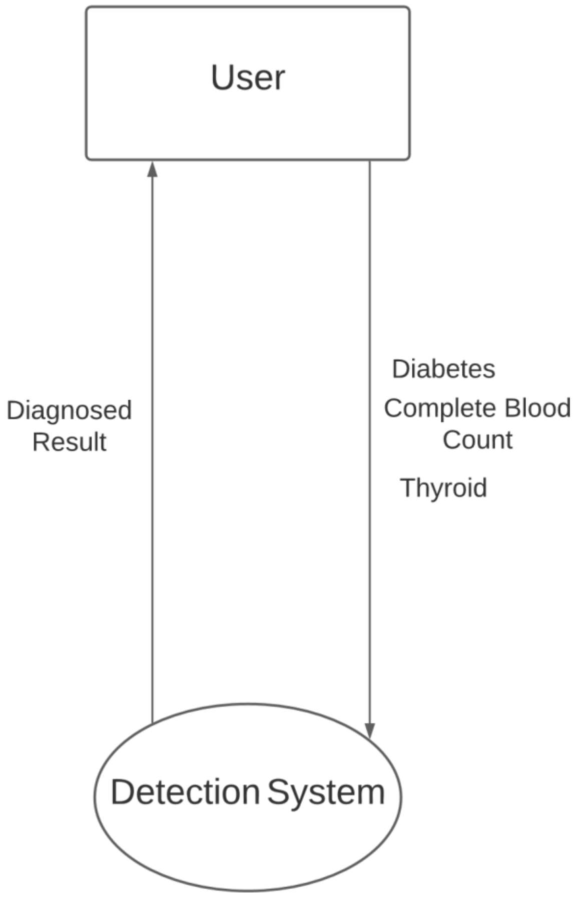
Represents a process transforms incoming

data into outgoing flow.



Represents data store

Context Diagram



4.4 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access key easy, quick, inexpensive and flexible for the user. Database is designed, built and populated with the data for the specific purpose. It has an intended group of users and some preconceived application on which the users are interested.

USE CASE DIAGRAM

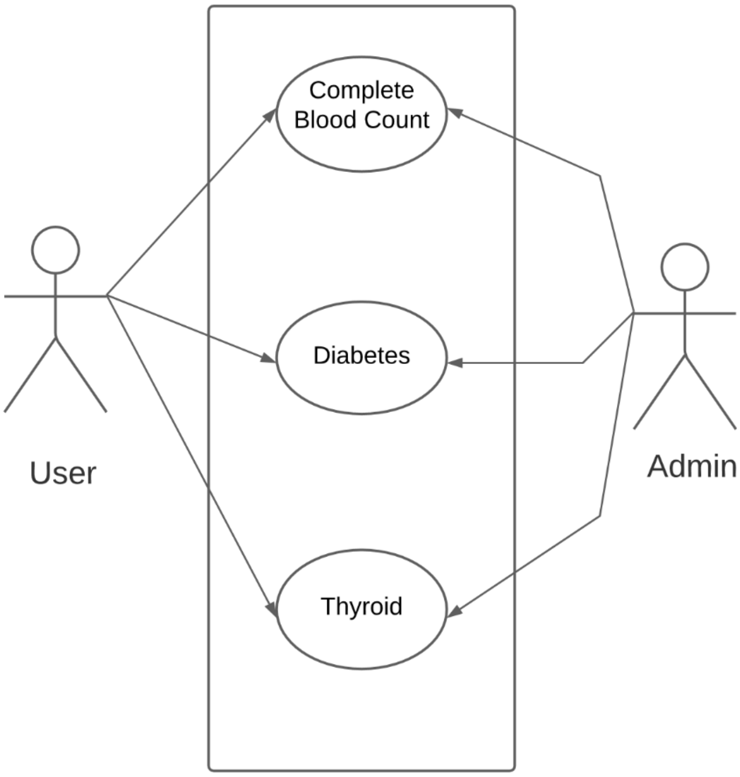
A use case represents the functionality of an actor. It is defined as a set of actions performed by a system, which yields an observable result. An ellipse containing its name inside the ellipse or below it represents that it is placed inside the system boundary and connected to an actor with in an association.

Use case analysis is an important and valuable requirement analysis technique that has been widely used in modern software engineering.

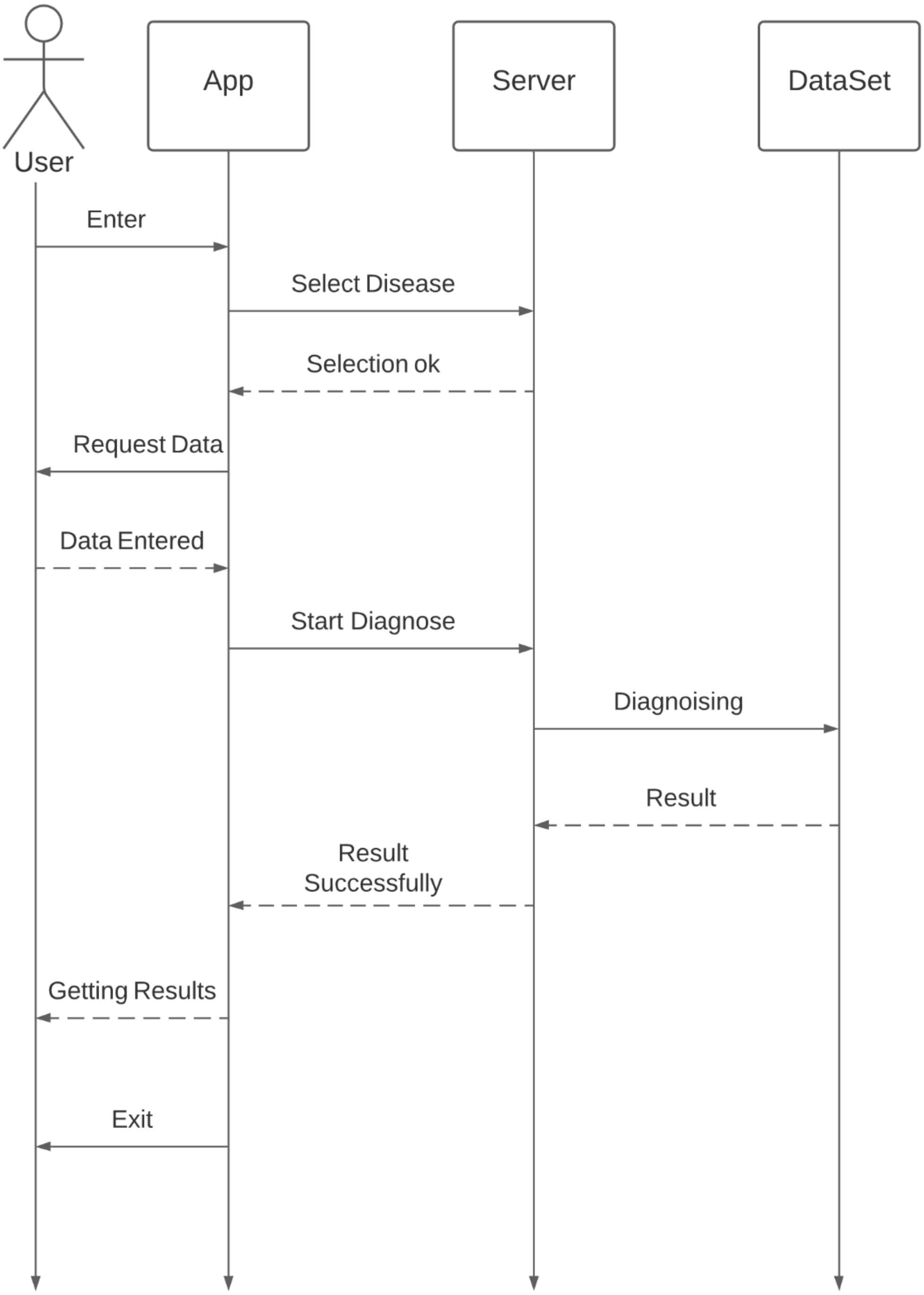
In this section, we identified the actors such as

- Admin
- User

The Use case diagram is shown below.


Complete Blood Count
Diabetes
Thyroid
User
Admin

SEQUENCE DIAGRAM



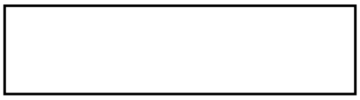
4.6 E-R DIAGRAM

An entity- relationship diagram is a data modelling technique that creates a graphical representation of the entities, and the relationship between entities, within an information system.

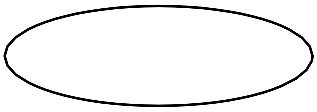
There are three basic elements in ER models:

- *Entities* are the “things” about which we seek information.
- *Attributes* are the data we collect about the entities.
- *Relationship* provide the structure needed to draw information from multiple entities.

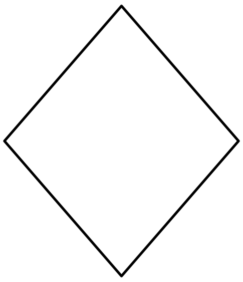
E-R Diagram symbols



Entity

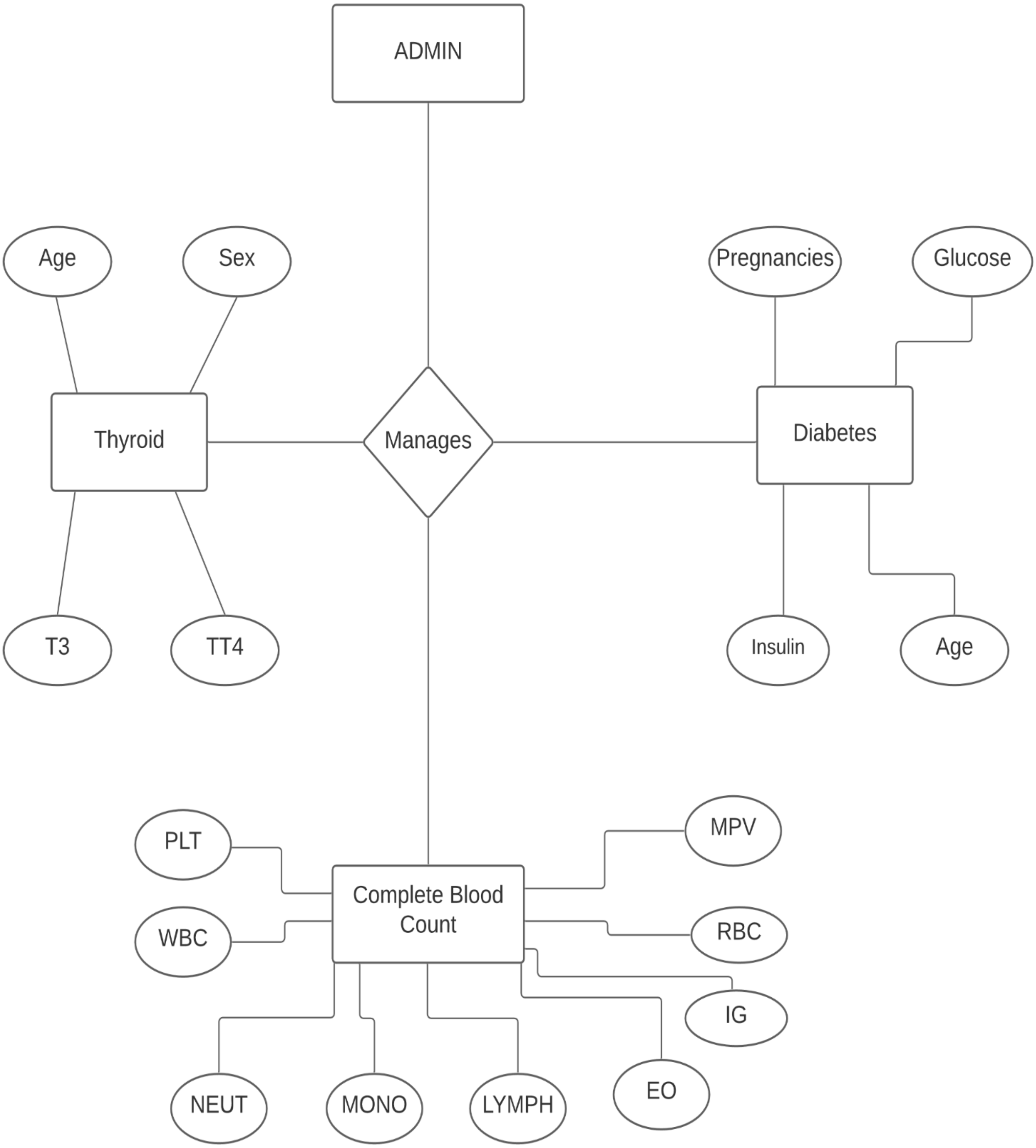


Attributes



Relationship

ER DIAGRAM


ADMIN
Age
Sex
Pregnancies
Glucose
Thyroid
Manages
Diabetes
T3
TT4
Insulin
Age
PLT
MPV
Complete Blood Count
WBC
RBC
IG
NEUT
MONO
LYMPH
EO

4.7 INTERFACE DESIGN

The interface design describes how the software communicates within itself with system that operates with it and with users who use it. The user interface is the link between user and the capabilities of our software. A well designed user interface makes it very easy for the audience to learn software. Through frequent consultation and reviews by the client we could develop a template for the interface, based on which other interfaces were constructed, as needed for specific requirements of the system.

User interface design is the design of the computer appliances and software applications with the focus of the user's experience and interaction.

The user interface design process encompasses four distinct framework activities:

- User, task, and environment analysis and modeling
- Interface design
- Interface construction
- Interface validation

CHAPTER 5

SYSTEM DEVELOPMENT

CODING STRUCTURE

System development is a series of operations performed to manipulate data to produce output from a computer system. This is highly dependent on the programming language used. The principle activities performed during the development phase can be divided into major sequences.

They are:

1. External system development
2. Internal system development

The major external system development activities are:

- Planning
- Equipment acquisition
- Installation
- Implementation

The major internal system development activities are

- Coding Structure
- Computer program development
- Performance testing

Coding translates a detailed representation of software into programming language realization. Here we used PHP for coding. Code design has been implemented giving priority to understandability, simplicity and clarity. Coding style has been chosen in such a way that it provide execution speed and minimum memory requirement. Programming language used for coding has greater influence over the final output.

5.1 SYSTEM SECURITY MEASURES

Various measures of security are laid down by the system. This provides facility for validating the identification code and password to make the system as reliable and interactive as possible. Necessary validation have been provided in filling input data. For any operation to be performed a user has to fill certain mandatory fields. If user leaves out any of them, a warning is provided to ensure proper and correct input.

CHAPTER 6

SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated and perform allocated functions.

Testing is the final verification and validation activity within the organization itself. Testing is done to achieve the following goals: to affirm the quality of the product, to find and eliminate any residual errors from previous stages, to validate the software as a solution to the original problem, to demonstrate the presence of all specified functionality in the product, to estimate the operational reliability of the system. During testing the major activities are concentrated on the examination and modification of the source code.

TESTING METHODOLOGIES ADOPTED:

- Unit testing
- Integration testing
- Validation testing
- Output testing

6.1.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, it is called unit testing.

6.1.2 INTEGRATION TESTING

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of integration testing:

1. **Top down integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breathe first manner.

2. **Bottom up integration**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low level modules are combined into clusters that perform a specific software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed clusters are combined moving upward in the program structure.

6.1.3 VALIDATION TESTING

In validation testing the requirements established as part of the requirement analysis are validated against the software that has be constructed. Validation testing provides final assurance that software meets all functional, behavioral, and performance requirements. Black box testing techniques are used exclusively during validation.

6.1.4 OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. The outputs generated or displayed by the system under consideration are tested by asking the users about the format required by them. Hence the output format is considered in two ways – one is on screen and another in printed format.

Alpha Testing

The developer in collaboration does alpha testing with the end user. The developer guides the end user all the way while observing the reaction and feedback.

Beta Testing

Beta testing releases the product to the end user. He has the liberty to proceed as the requirements dictate. The user records all the problems that are encountered during beta testing and reports these to the developer at regular intervals. As a result of this the software developer makes modification and then prepares for the release of the software product to entire customer base

6.2 SYSTEM IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause chaos and confusion.

The process of putting the development system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications.

The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover.

The system implementation has three main aspects. They are education and training, system testing and changeover.

The implementation stage involves following tasks.

- Careful planning
- Investigation of system and constraints
- Design of methods to achieve the change over Evaluation of the changeover method

CHAPTER 7

CONCLUSION

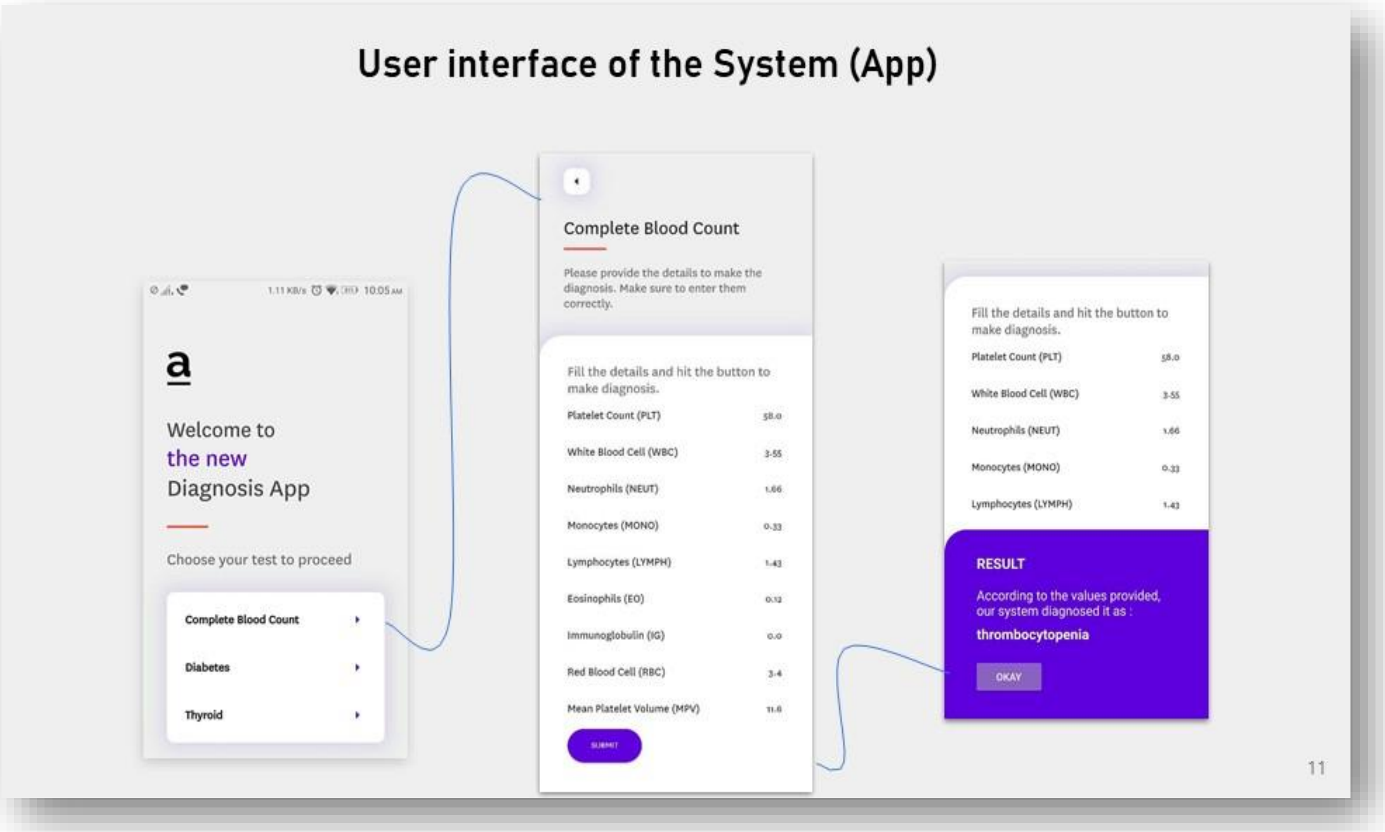
CONCLUSION

This Project “**DISEASE PREDICTION AND DETECTION SYSTEM**” has been completed successfully and marks the expectation of those who have guided us through this project. The constraints are met and overcame successfully. The system is designed as like it was decided in the design phase.

The aim of the project was to develop an accurate and easy to use disease prediction and detection system that utilizes the concepts of machine learning that could help patients and medical professionals in diagnosing and identifying diseases and predicting the chance of having a disease in the future. The project utilizes various public and private datasets regarding clinical laboratory tests and can be used to train and develop useful machine learning models. The generated models can be accessed by users through a user friendly end-to-end application available through the internet. Laboratory test results can be fed into the application to get corresponding diagnosis and predictions.

CHAPTER 8

SCREEN SHORTS



User interface of the System (App)

1.11 KB/s 10:05 AM

a

Welcome to
the new
Diagnosis App

Choose your test to proceed

Complete Blood Count

Please provide the details to make the diagnosis. Make sure to enter them correctly.

Fill the details and hit the button to make diagnosis.

Platelet Count (PLT)	58.0
White Blood Cell (WBC)	3.55
Neutrophils (NEUT)	1.66
Monocytes (MONO)	0.33
Lymphocytes (LYMPH)	1.43
Eosinophils (EO)	0.12
Immunoglobulin (IG)	0.0
Red Blood Cell (RBC)	3.4
Mean Platelet Volume (MPV)	11.6

Fill the details and hit the button to make diagnosis.

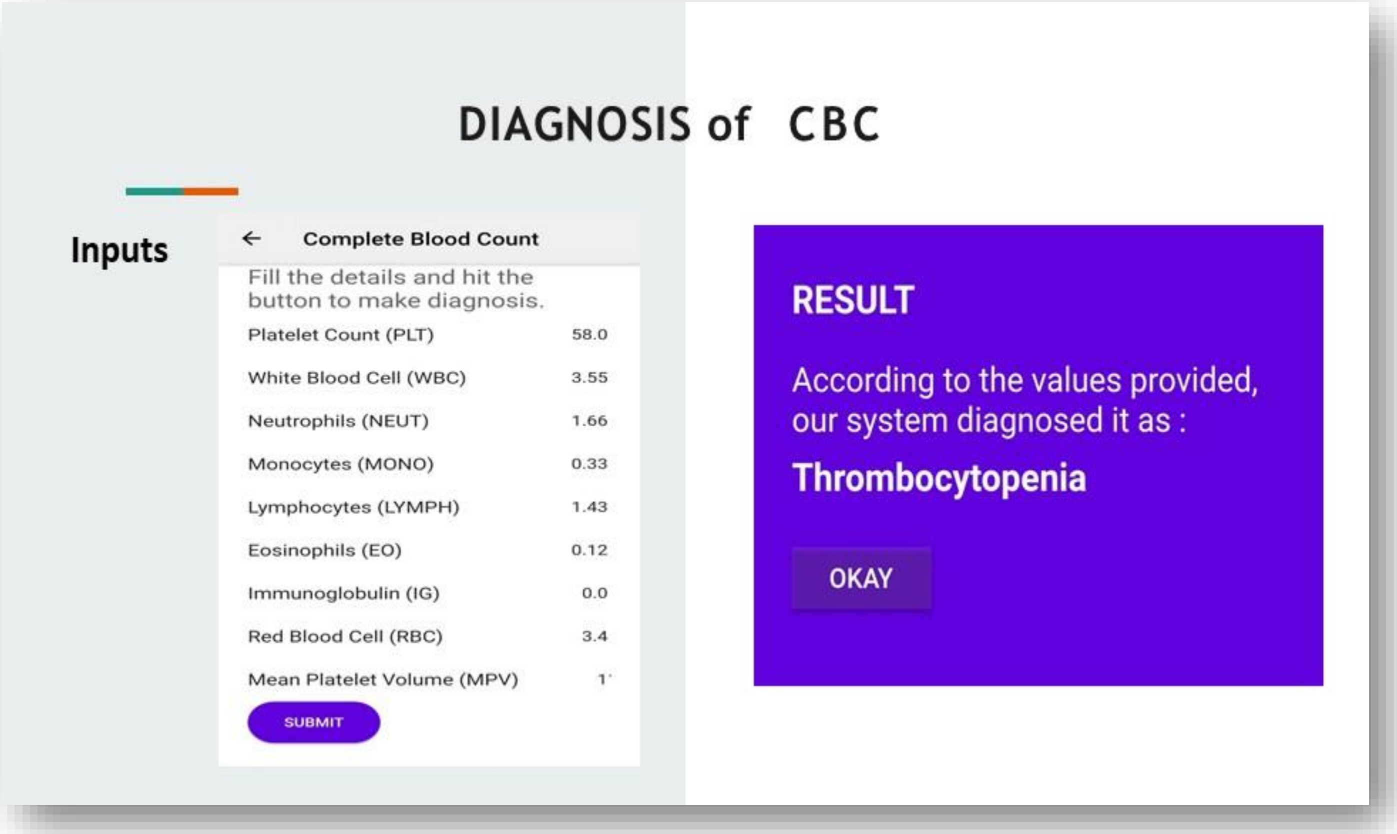
Platelet Count (PLT)	58.0
White Blood Cell (WBC)	3.55
Neutrophils (NEUT)	1.66
Monocytes (MONO)	0.33
Lymphocytes (LYMPH)	1.43

RESULT

According to the values provided, our system diagnosed it as :

thrombocytopenia

11



DIAGNOSIS of CBC

Inputs

Complete Blood Count

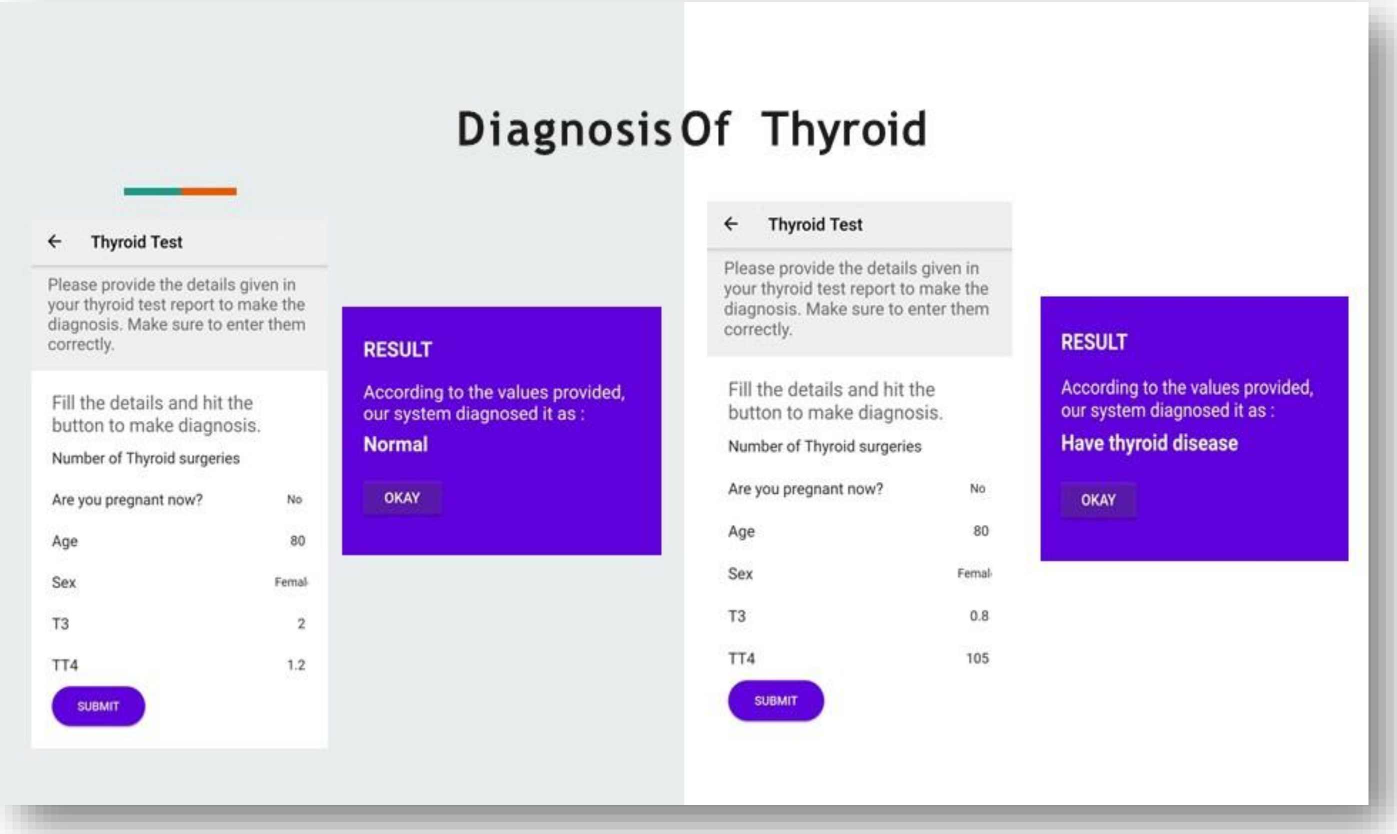
Fill the details and hit the button to make diagnosis.

Platelet Count (PLT)	58.0
White Blood Cell (WBC)	3.55
Neutrophils (NEUT)	1.66
Monocytes (MONO)	0.33
Lymphocytes (LYMPH)	1.43
Eosinophils (EO)	0.12
Immunoglobulin (IG)	0.0
Red Blood Cell (RBC)	3.4
Mean Platelet Volume (MPV)	1

RESULT

According to the values provided, our system diagnosed it as :

Thrombocytopenia



Diagnosis Of Thyroid

← Thyroid Test

Please provide the details given in your thyroid test report to make the diagnosis. Make sure to enter them correctly.

Fill the details and hit the button to make diagnosis.

Number of Thyroid surgeries

Are you pregnant now?	No
Age	80
Sex	Femal
T3	2
TT4	1.2

RESULT

According to the values provided, our system diagnosed it as :

Normal

← Thyroid Test

Please provide the details given in your thyroid test report to make the diagnosis. Make sure to enter them correctly.

Fill the details and hit the button to make diagnosis.

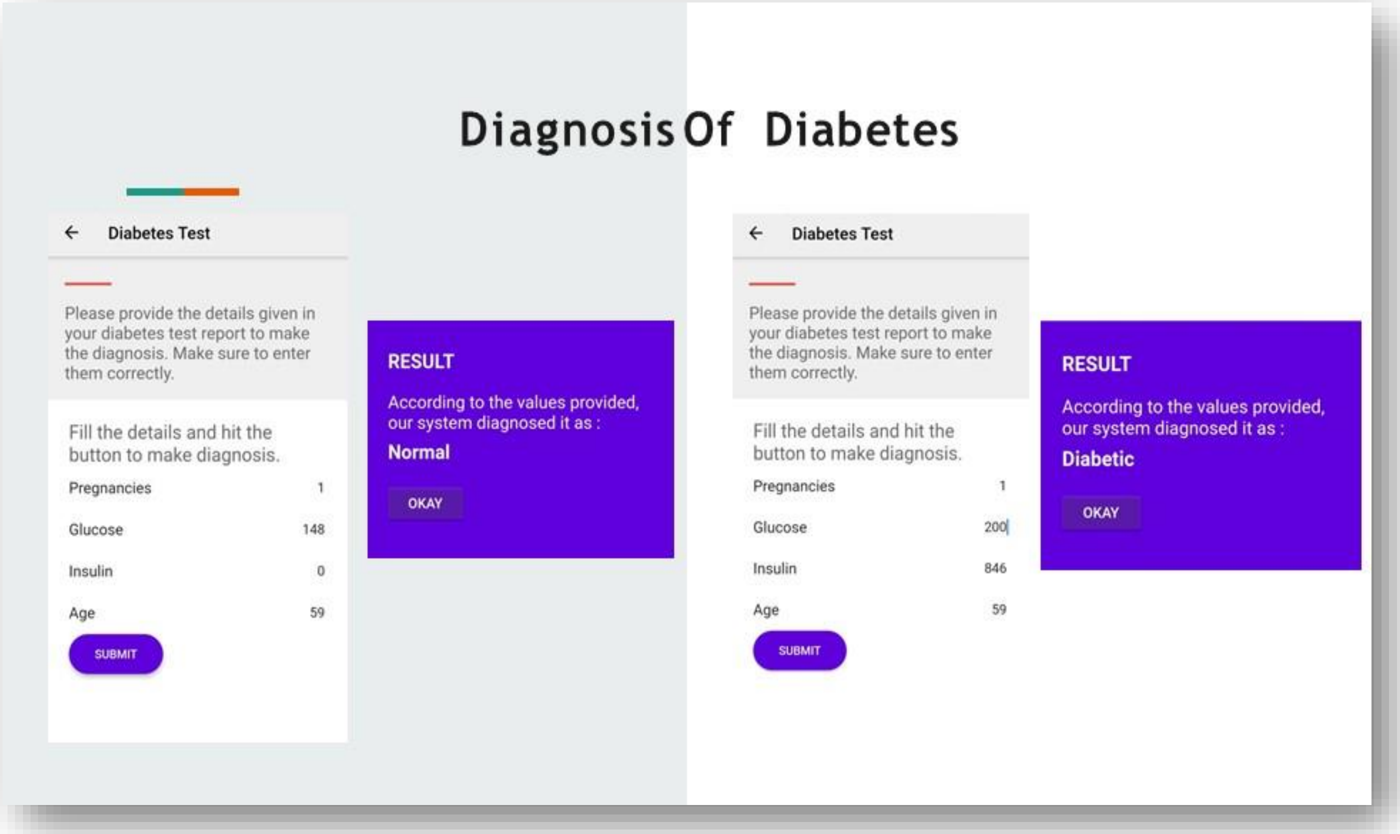
Number of Thyroid surgeries

Are you pregnant now?	No
Age	80
Sex	Femal
T3	0.8
TT4	105

RESULT

According to the values provided, our system diagnosed it as :

Have thyroid disease



Diagnosis Of Diabetes

Diabetes Test

Please provide the details given in your diabetes test report to make the diagnosis. Make sure to enter them correctly.

Fill the details and hit the button to make diagnosis.

Pregnancies	1
Glucose	148
Insulin	0
Age	59

RESULT

According to the values provided, our system diagnosed it as :

Normal

Diabetes Test

Please provide the details given in your diabetes test report to make the diagnosis. Make sure to enter them correctly.

Fill the details and hit the button to make diagnosis.

Pregnancies	1
Glucose	200
Insulin	846
Age	59

RESULT

According to the values provided, our system diagnosed it as :

Diabetic