MINOR PROJECT

Submitted in partial fulfilment of the requirements
for the
award of the degree of

Bachelor of Technology Computer Science and Engineering

Guide(s):
Ms. Richa Sharma
Assistant Professor
(Department of Computer Science)

Submitted By:

HMR INSTITUTE OF TECHNOLOGY & MANAGEMENT

HAMIDPUR, DELHI 110 036

Affiliated to

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY

Sector - 16C Dwarka, Delhi - 110075, India

2016-20

DECLARATION

We, student of B.Tech (CSE 7th) hereby declare that the minor project entitled **"HEALTH CARE SERVICE"** which is submitted to Department of CSE, HMR Institute of Technology & Management, Hamidpur Delhi, affiliated to Guru Gobind Singh Indraprastha University, Dwarka(New Delhi) in partial fulfilment of requirement for the award of the degree of Bachelor of Technology in CSE, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

This is to certify that the above statement made by the candidates are correct to the best of our knowledge. The list of members involved in the project are listed below: -

STUDENT NAME	ROLL NO.	STUDENT SIGNATURE
RAJA	04713302716	
RAHUL GOYAL	20113302716	

New Delhi	Ms. Richa Sharma
Date:	Signature of Guide

Mr. Mohd. Izhar

(Head of Department)

(Assistant Professor of Computer Science Department)

HMRITM Hamidpur, New Delhi-110036

ACKNOWLEDGEMENT

We would like to thank our minor's project guide Ms. Richa Sharma for her support, encouragement and guidance during the period of our training and helping us to make "**HEALTH CARE SERVICE**". She taught us a lot about python, data analytics, machine learning and guided us through every step of our project. She gave us lot of inspiration during the industrial training which help us a lot and due to her supportive feedback, We were able to complete our project named as health care service.

Raja
(04713302716)

Rahul Goyal
(20113302716)

ABSTRACT

The rapid growth in the field of data analysis and machine learning plays an important role in the healthcare research. Due to large amount of data growth in biomedical and healthcare field providing accurate analysis of medical data that has benefits from early detection, patient care, and community services. Previous system designed to analyse, manage and assimilate data produced by healthcare systems. Data analysis has been applied to help the disease-related information and treatment process. In this paper a decision tree is effectively used for predicting the outbreaks of diseases in society.

The paper proposes to experiment with the modified predictive models with medical data which is related to the symptoms of the disease. For the disease prediction using unstructured data, we used a decision tree classification algorithm which is based on multimodal disease risk prediction algorithm. Users can post their queries in order to seek information regarding diseases so that user get the proper answer to any kind of question and solving any problem related to the disease, and this model was also providing the name and link of the best doctor related to that diagnosis.

TABLE OF CONTENTS

LIST OF FIGURES

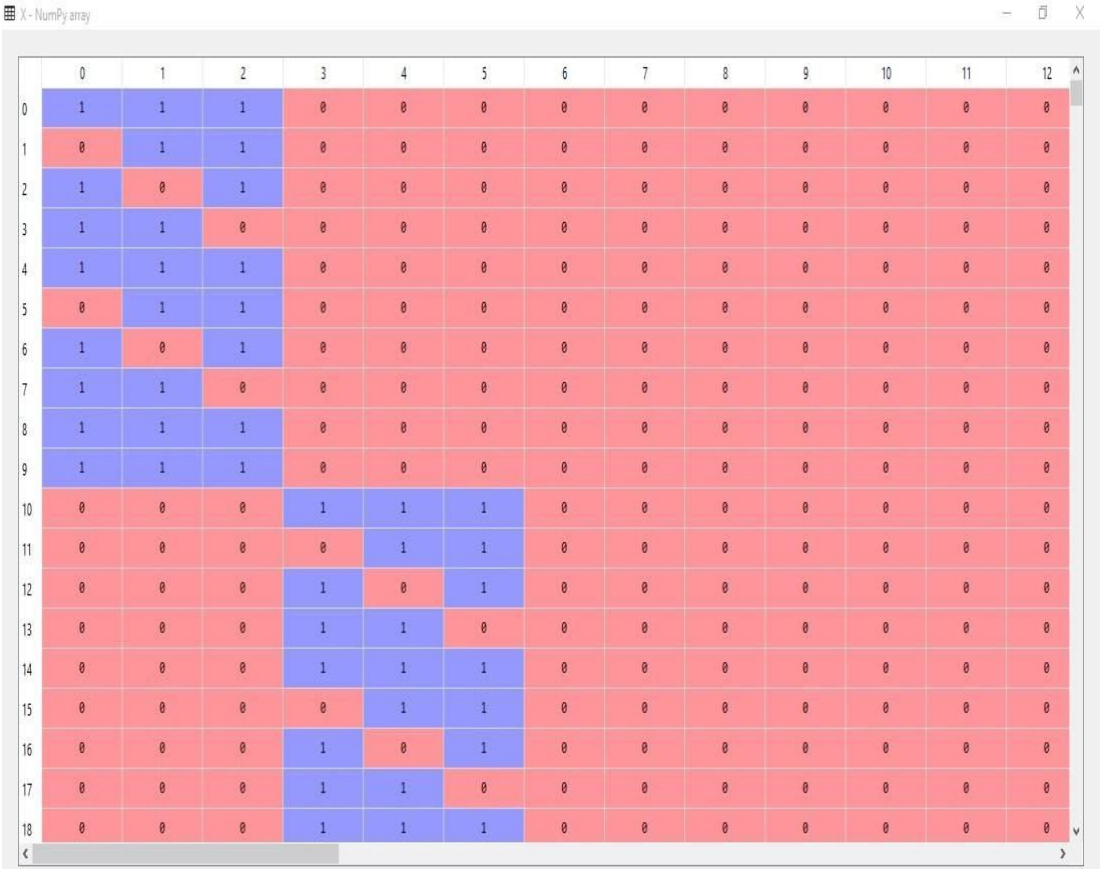
LIST OF TABLES

It has Four Columns: -

1. Name
2. Type
3. Size
4. Value

Name	Type	Size	Value
E	str	1	e
N	str	1	n
S	str	1	s
TclVersion	float	1	8.6
TkVersion	float	1	8.6
W	str	1	w

X_ DATASET: - It contains number of inputs given by users.

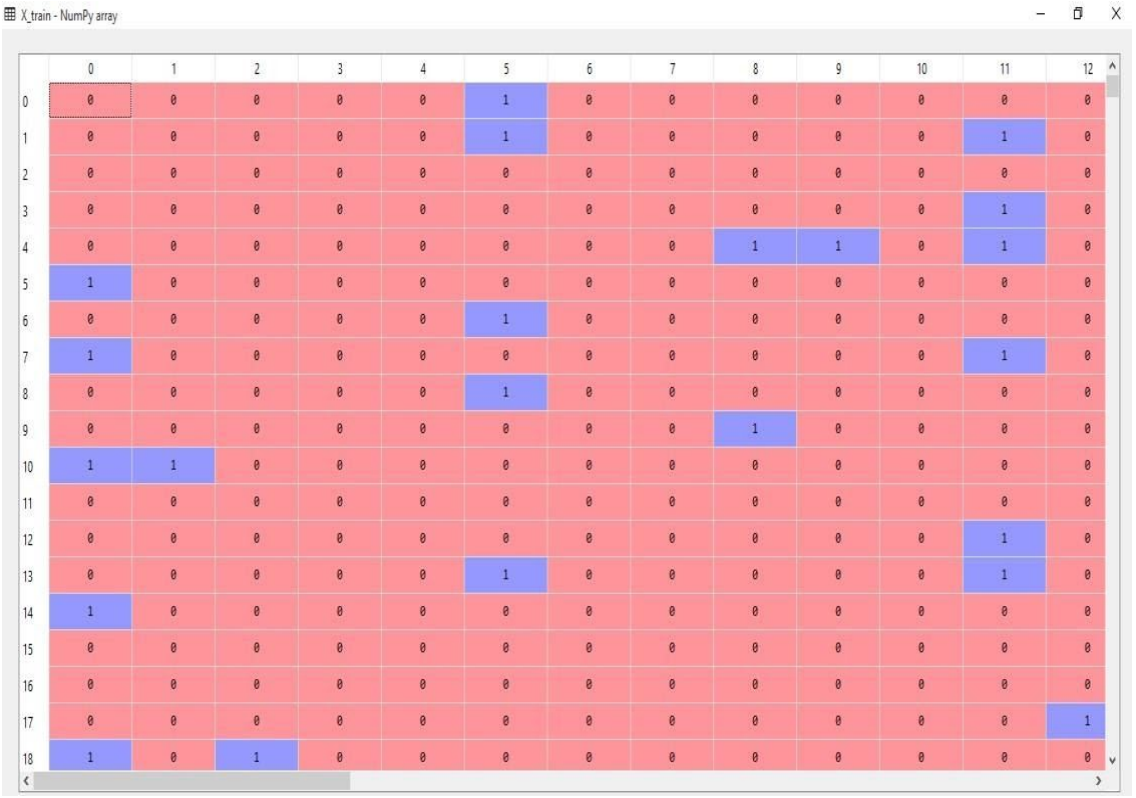


X_TEST DATASET:- It divides the x just to check if our output will be correct or not.

X_test - NumPy array

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	1	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	1	1	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	1	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0

X_TRAIN DATASET:- It also divide x dataset into another part called as training set.



	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	1	1	0	1	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0
10	1	1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	1	0	0	0	0	0	1	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	1
18	1	0	1	0	0	0	0	0	0	0	0	0	0

DIMENSIONALIY_REDUCTION: - To avoid repeat value and contain unique values.

dimensionality_reduction - DataFrame

Index	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	muscle_wasting	vomiting	um
(vertigo) Paroymsal P...	0	0	0	0	0	0	0	0	0	0	0	1	0
AIDS	0	0	0	0	0	0	0	0	0	0	1	0	0
Acne	0	1	0	0	0	0	0	0	0	0	0	0	0
Alcoholic hepatitis	0	0	0	0	0	0	0	0	0	0	0	1	0
Allergy	0	0	0	1	1	1	0	0	0	0	0	0	0
Arthritis	0	0	0	0	0	0	0	0	0	0	0	0	0
Bronchial Asthma	0	0	0	0	0	0	0	0	0	0	0	0	0
Cervical spondylosis	0	0	0	0	0	0	0	0	0	0	0	0	0
Chicken pox	1	1	0	0	0	0	0	0	0	0	0	0	0
Chronic cholestasis	1	0	0	0	0	0	0	0	0	0	0	1	0
Common Cold	0	0	0	1	0	1	0	0	0	0	0	0	0
Dengue	0	1	0	0	0	1	1	0	0	0	0	1	0
Diabetes	0	0	0	0	0	0	0	0	0	0	0	0	0
Dimorphic hemmorhoids(...	0	0	0	0	0	0	0	0	0	0	0	0	0
Drug Reaction	1	1	0	0	0	0	0	1	0	0	0	0	1
Fungal infection	1	1	1	0	0	0	0	0	0	0	0	0	0
GERD	0	0	0	0	0	0	0	1	1	1	0	1	0
Gastroenteri...	0	0	0	0	0	0	0	0	0	0	0	1	0
Heart attack	0	0	0	0	0	0	0	0	0	0	0	1	0
Hepatitis B	1	0	0	0	0	0	0	0	0	0	0	0	0

DISEASES: - Number of Diseases.

diseases - DataFrame

Index	prognosis
0	(vertigo) Paroymsal P...
1	AIDS
2	Acne
3	Alcoholic hepatitis
4	Allergy
5	Arthritis
6	Bronchial Asthma
7	Cervical spondylosis
8	Chicken pox
9	Chronic cholestasis
10	Common Cold
11	Dengue
12	Diabetes
13	Dimorphic hemmorhoids(...
14	Drug Reaction
15	Fungal infection
16	GERD
17	Gastroenteri...
18	Heart attack
19	Hepatitis B
20	Hepatitis C

DOCTER DATASET: - It will provide Doctor's name with their link where anyone can visit his/her site and also tell problem. It contains multiple rows and three columns.

doctors - DataFrame

Index	name	link	disease
0	Dr. Amarpreet Singh Riar	https:// www.practo.c…	(vertigo) Paroymsal P…
1	Dr. (Maj.)Sharad…	https:// www.practo.c…	AIDS
2	Dr. Anirban Biswas	https:// www.practo.c…	Acne
3	Dr. Aman Vij	https:// www.practo.c…	Alcoholic hepatitis
4	Dr. Mansi Arya	https:// www.practo.c…	Allergy
5	Dr. Sunil Kumar Dwivedi	https:// www.practo.c…	Arthritis
6	Dr. Chhavi Bansal	https:// www.practo.c…	Bronchial Asthma
7	Dr. Sneh Khera	https:// www.practo.c…	Cervical spondylosis
8	Dr. Inderjeet Singh	https:// www.practo.c…	Chicken pox
9	Dr. Suman Mohan	https:// www.practo.c…	Chronic cholestasis
10	Dr. Manish Munjal	https:// www.practo.c…	Common Cold
11	Dr. Ajay Jain	https:// www.practo.c…	Dengue
12	Dr. Anshul Gupta	https:// www.practo.c…	Diabetes
13	Dr. B B Khatri	https:// www.practo.c…	Dimorphic hemmorhoids(…
14	Dr. Rajeev Adhana	https:// www.practo.c…	Drug Reaction
15	Dr. Vidit Tripathi	https:// www.practo.c…	Fungal infection
16	Dr. Arun Wadhawan	https:// www.practo.c…	GERD
17	Dr. Neha Sood	https:// www.practo.c…	Gastroenteri…
18	Dr. Vineet Narula	https:// www.practo.c…	Heart attack
19	Dr. Yogesh Jain	https:// www.practo.c…	Hepatitis B
20	Dr. Rakesh Singh	https:// www.practo.c…	Hepatitis C

Y _TEST DATASET: - It contains output only for test dataset.

y_test - NumPy array

	0
0	18
1	40
2	36
3	25
4	33
5	23
6	31
7	39
8	40
9	37
10	38
11	27
12	16
13	39
14	15
15	32
16	23
17	1
18	26

Y_TRAINING DATASET: -

y_train - NumPy array

	0
0	34
1	36
2	24
3	21
4	16
5	19
6	37
7	9
8	34
9	30
10	8
11	24
12	25
13	37
14	19
15	13
16	26
17	38
18	15

CHAPTER - 1

1.1 INTRODUCTION

The project named as Health care service, which does a great job at reporting on various methods which helps to improve healthcare from a cost as well as a care perspective, which focused on improving healthcare cost by reducing wasteful procedures. For That Purpose, we will create a chat-bot that predicts the diseases based on the symptoms and provide the link of best doctor for the treatment of the diseases.

The purpose of this project is to provide admin to collect the patients' medical history of records and filter it appropriately by applying data pre-processing techniques. Once the data comes into the structured shape it can then be fed into the relational database structure of MS Excel file. The admin also needs to monitor the predictions and replies of the model to ascertain quality.

In this project we have given the options in chat-bot such as start, yes, no and clear from which if user wants to start the program, he/she has to click on start button, if user wants to clear the screen then he/she has to click on clear button, and such options like Yes/No is used to classify for the user that if disease belongs to his/her then click on Yes otherwise click on No. On the window, user will get the output such as predicted disease by Machine and user also will get the name and link of the doctor who is specialized on that specific disease.

1.2 OVERVIEW OF THE PROJECT

This project involves Admin's functionalities are to Collecting the appropriate medical records of the patients, handle missing values, handling categorical values, creating sparse matrix representation, feeding data to the autonomous pipeline for predictions, selecting and training an appropriate machine learning algorithm.

Visitor can perform the basic task of visitor is to access the chat bot from the front end and reply to its queries with a binary response (Yes/No). The visitor will be shown a confidence interval related to a certain prognosis which needs to be further investigated and experimented with for better results. The visitor can also contribute with the help of the admin to add new symptoms and diagnosis records to the database of medical history.

The window application purpose is to allow these clients an easy medium in which to check their health issues and provide best doctor according to their symptoms. The first step is to start their procedure, then one by one all the symptoms come in clients screen. They will have to reply with yes or no answer. Once a problem will be found then they will have to click yes, then patient can see their problem in screen.

The Best Part is that it will provide doctor's information like Doctor name and his/her website link. So that one can easily find their doctor with don't face any type of problem, and start their treatment. This will prepare with the help of chatbot so that one can even check their problem at any time. You have to just reply with clicking of button Yes or

No.

1.3 LITERATURE SURVEY

1. This project is based on the platform of Python, Data Analytics, and Machine Learning.
2. Symptoms and the user's input (yes/no) are based on the Data Analytics.
3. The confidence level predicted by the machine is based on the Machine Learning which shows the accuracy of the prediction by the machine.
4. Overall design and implementation are based on the python including Tkinter library to run and execute the process in graphical manner.
5. Software Used: Python, Anaconda, jupyter.
6. Data-Bases Connectivity: registered id will be stored in the database so that user can access it again and again by entering the correct login id and password.
7. Data Collection: Collection of the data is done by Kaggle website.

Data Analysis Procedure: Anaconda software to be used, which descriptive statistics and which test of significance if and when required, specifying variables where it will be applied.

Machine Learning Procedure: It splits the dataset into 2 parts (Train, Test), where test is to perform on the basis of train dataset, then after using the decision tree algorithm it will make some predictions which describe the accuracy of the machine.

1.4 PROBLEM STATEMENT

Health care service, which does a great job at reporting on various methods which helps to improve healthcare from a cost as well as a care perspective, which focused on improving healthcare cost by reducing wasteful procedures. For That Purpose, we will create a chat-bot that predict the diseases based on the symptoms and provide the link of best doctor for the treatment of the diseases.

The window application purpose is to allow the clients an easy medium in which to check their health issues. To provide best doctor according to their symptoms. Improved / Increased / Enhanced Patient Satisfaction.

1.5 SCOPE OF STUDY

The provision of high-quality, affordable, health care services is an increasingly difficult challenge. Due to the complexities of health care services and systems, investigating and interpreting the use, costs, quality, accessibility, delivery, organization, financing, and outcomes of health care services is key to informing government officials, insurers, providers, consumers, and others making decisions about health-related issues. Health services researchers examine the access to care, health care costs and processes, and the outcomes of health services for individuals and populations.

Health services research is the multidisciplinary field of scientific investigation that studies how social factors, financing systems, organizational structures and processes, health technologies, and personal behaviours affect access to health care, the quality and cost of health care, and ultimately our health and well-being. Its research domains are individuals, families, organizations, institutions, communities, and populations.

Patient Safety: Patients should not be harmed by health care services that are intended to help them. The IOM report, *To Err Is Human*, found that between 46,000 and 98,000 Americans were dying in hospitals each year due to medical errors. Subsequent research has found medical errors common across all health care settings. The problem is not due to the lack of dedication to quality care by health

professionals, but due to the lack of systems that prevent errors from occurring and/or prevent medical errors from reaching the patient.

Effectiveness: Effective care is based on scientific evidence that treatment will increase the likelihood of desired health outcomes. Evidence comes from laboratory experiments, clinical research (usually randomized controlled trials), epidemiological studies, and outcomes research. The availability and strength of evidence varies by disorder and treatment.

Timeliness: Seeking and receiving health care is frequently associated with delays in obtaining an appointment and waiting in emergency rooms and doctors' offices. Failure to provide timely care can deny people critically needed services or allow health conditions to progress and outcomes to worsen. Health care needs to be organized to meet the needs of patients in a timely manner.

Patient centred: Patient-centred care recognizes that listening to the patient's needs, values, and preferences is essential to providing high-quality care. Health care services should be personalized for each patient, care should be coordinated, family and friends on whom the patient relies should be involved, and care should provide physical comfort and emotional support.

Efficiency: The U.S. health care system is the most expensive in the world, yet there is consistent evidence that the United States does not produce the best health outcomes or the highest levels of satisfaction. The goal is to continually identify waste and inefficiency in the provision of health care services and eliminate them.
Equity: The health care system should benefit all people. The evidence is strong and convincing that the current system fails to accomplish this goal. The IOM report, *Unequal Treatment*, documented pervasive differences in the care received by racial and ethnic minorities. The findings were that racial and ethnic minorities are

receiving poorer quality of care than the majority population, even after accounting for differences in access to health services.

CHAPTER - 2

2.1 SYSTEM ANALYSIS

Software Engineers have been trying various tools, methods and procedures to control the process of software development in order to build high quality software with high productivity. This method provides “how it is” for building the software while the tools provide automated or semi-automated support for the methods. They are used in all stages of software development process, namely, planning, analysis, design, development and maintenance. The software development procedure integrates the methods and tools together and enables rational and timely development of the software system.

Systems development is systematic process which includes phases such as planning, analysis, design, deployment, and maintenance, we will primarily focus on −

1. Systems analysis
2. Systems design

1. System analysis:

System Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that

all the components of the system work efficiently to accomplish their purpose. They provide the guidelines as how to apply these methods and tools, how to produce the deliverable at each stage, what controls to apply, and what milestones to use to access the performance of the program. There exist a number of software development paradigms each using a different set of methods and tools. The selection of a particular paradigm depends on the nature of application of the programming language used for the controls and the deliverables required. The development of such successful systems depends not only on the use of appropriate methods and techniques but also the developers' commitment to the objective of the system.

A successful system must: -

1. Satisfy the user requirement
2. Be easy to understand by user and operator
3. Be easy to operate
4. Have a good user interface
5. Be easy to modify
6. Be expandable
7. Have adequate security control against the misuse of data
8. Handle the errors and exceptions satisfactorily
9. Be delivered on schedule within the budget

Within this analysis phase, the analyst is discovering and fact finding. Along with meeting with stakeholders, the analyst must meet with end users to understand what the user's needs are and to learn about problems that affect the current system in order to assist with designing a new and more efficient system. There are several activities that must occur within the analysis phase:

1. Gather Information
2. Define the new system's requirements
3. Build prototypes for the new system
4. Prioritize requirements
5. Evaluate alternatives
6. Meet with management to discuss new options

2. System Design:

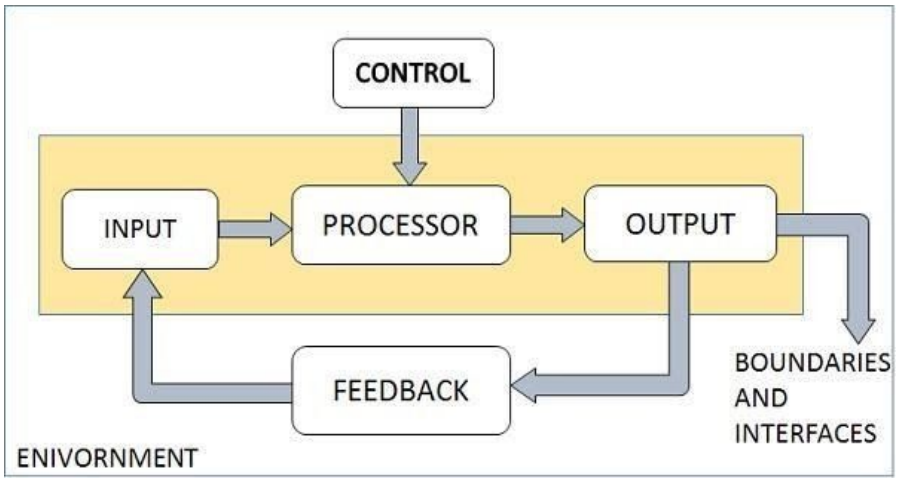
It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

System Design focuses on how to accomplish the objective of the system.

System Analysis and Design (SAD) mainly focuses on −

1. Systems
2. Processes
3. Technology

Systems Analysis and Design is an active field in which analysts repetitively learn new approaches and different techniques for building the system more effectively and efficiently. The primary objective of systems analysis and design is to improve organizational systems. This tutorial provides a basic understanding of system characteristics, system design, and its development processes. It is a good introductory guide that provides an overview of all the concepts necessary to build a system.



System design is most creative phase of the system development. The term describes a final system and the process by which it is developed. The question in system design is:

How the problem is to be solved?

A systematic method has to achieve the beneficial results at the end. It involves starting with a vague idea and developing it into a series of steps. The series of steps for successful system design are:

1. First step is to study the problem completely because we should know about the goal. We should see what kind of output we require and what king of input we give so that we can get desired result. We should see what kind of program should be developed to reach the final goal.

2. Then we write individual programs, which later on joining solve the specified problem.

3. Then we test these programs and make necessary corrections to achieve target of the programs.

While designing we had to consider all the requirements of the users to make such interface which makes the communication easy and accurate. The interface is designed to provide the efficient and clear information of each and every detail of application.

2.2 FEASIBILITY STUDY

A feasibility study is undertaken to determine the possibility or probability of either improving the existing system or developing a completely new system. It helps to obtain an overview of the problem and to get rough assessment of whether feasible solution exists or not. There are three aspects in feasibility study portion of the preliminary investigation.

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from

performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, legal requirements. Generally, such studies precede technical development and project implementation, such are given below: 1. Is there any new and better way to do the job that solves the problem?

2. What are the costs and savings of the alternatives?
3. Are there any legal restrictions imposed by the government or any other regulatory body?
4. Is the proposed solution economically feasible?
5. Does there exists any bottleneck that may turn the process of development futile exercise?
6. What is recommended?

Although this project has very good feasibility due to required less resources to complete this project, where the requirement features are given in the software and hardware specification unit.

Five Areas of Project Feasibility

A feasibility analysis evaluates the project's potential for success; therefore, perceiving objectivity is an essential factor in the credibility of the study for potential investors and lending institutions.

1. Economical feasibility
2. Legal feasibility
3. Technical feasibility

4. Operational feasibility
5. Behavioural feasibility

There are three types of feasibility study separated areas that a feasibility study examines.

1. Technical Feasibility

2. Economic Feasibility

3. Operational Feasibility

2.2.1 Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves evaluation of the hardware, software, and other technical requirements of the proposed system.

Organization objective is to serve the plan to create a standard application which addresses the dual problems like availability of information and format conversions. The proposed system will fulfil both objectives because the use of web site provides most secure.

Health Care Service is the best application in manner of technical feasibility because the technical resources requirement for this project is very less. This project involves Python software just to be coding in python programming language, which is perform on PiCharm. Pi-Charm is the type of open source application where we can implement

python programming language easily. For user interface we have used TkInter toolkit window which is python library and for database connectivity we have used PyMySQL server.

Technical feasibility centres on existing computer system (hardware and software) etc. and to what extent it can support the proposed edition. It is technically feasible, since the whole system is designed into the technologies like Python which are the most recent technologies to develop windows-based systems.

2.2.2 ECONOMICAL FEASIBILITY

The tools that will be used for the system are latest one and thus the cost involved in tools, designing and developing the system will be a good investment for the organization. The benefits of using the system are not in monetary terms, but its increased interaction between Users & administrators. The modules designed can be easily navigated. The hardware available with the company is already the best available and hence no new purchase is required.

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

Applications and features used in this project:

1. Python 3.6
2. PyMySQL server

3. TkInter window Toolkit
4. OOPs concept

All of the above are free of cost and Eco-friendly in nature as well as all of the above are open source so that anyone can use it. For user interface we have used TkInter toolkit window which is python library and it is free of cost just need to download. And for database connectivity we have used PyMySQL server which is also free of cost. Rest of these we have used the platform named as Pi-Charm which is also free of cost and open source environment.

2.2.3 OPERATIONAL FEASIBILITY

The systems analyst must still consider the operational feasibility of the requested project. Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will operate and be used once it is installed. The proposed system provides the functionality based on user characteristics to End user or authorized user. This assessment involves undertaking a study to analyse and determine whether—and how well—the organization's needs can be met by completing

the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

Once it is determined that system is both technically and economically feasible then it has to be seen if it is operationally feasible. Operational feasibility refers to projecting whether the system will operate and be used once it is installed.

The proposed system is operationally feasible, as there no need for the faculty to keep a check on use of internet at the time of execution. There no need to keep a manual eye on the behaviour of client on the internet. Site blocking, service blocking (FTP/HTTP) and content filtering features of proposed system cater to this very task.

We have designed front end by getting the information from the end user, which help us in designing the GUI according to the end user's requirements. The end users can easily understand and expand it in the future.

As the faculty members expressed the need for an improved system, they put in all efforts to see that it becomes feasible.

2.3 SOFTWARE AND HARDWARE SPECIFICATIONS

Software specification:

The aim of the system is to develop **"HEALTH CARE SERVICE"**, which should automate the process to create and store user details. The system is supposed to be used as a subsystem in a large office system, which could be manual system or a computerized one. Therefore, the proposed system must be able to function under both circumstances.

The project demands a page of user personal details that include:

1. User name
2. User password

Project Category

This project will be a client server application and also a Window based application, develops on Python (TkInter) technology using the following tools:

- OOPS (Object Oriented Programming System)
- PYTHON
- PyMySQL
- TKINTER

1. OOPs (Object Oriented Programming System) in Python:

1.1 Object.

1.2 Class.

1.3 Method.

1.4 Inheritance.

1.5 Polymorphism.

1.6 Data Abstraction.

1.7 Encapsulation.

2. Python:

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site and may be freely distributed. The same site also contains distributions of and pointers to many free thirdparty Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

Python Features

- Easy to Learn and Use
- Interpreted Language
- Cross-platform Language
- Free and Open Source
- Object-Oriented Language
- Extensible
- Large Standard Library
- GUI Programming Support
- Integrated

Disadvantages of Python

- Speed
- Mobile Development
- Memory Consumption
- Database Access
- Runtime Errors

3. PyMySQL

In order to connect Python to a database we need a driver, which is a library used to Interact with the database. For MySQL database, we have such 3 Driver choices:

3.1 MySQL/connector for Python

3.2 MySQLdb

3.3 PyMySQL

PyMySQL is an interface for connecting to a MySQL database **server** from Python. It implements the Python Database API v2.0 and contains a pure-**Python MySQL** client library. This is a library that connects to MySQL from Python and it is a pure Python library. PyMySQL's goal is to replace MySQLdb.

TkInter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, TkInter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with TkInter outputs the fastest and easiest way to create the GUI applications.

To create a tkinter:

3.4 Importing the module – tkinter

3.5 Create the main window (container)

3.6 Add any number of widgets to the main window

3.7 Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x is 'tkinter'.

```
import tkinter    import tkinter
```

Software requirements

1. Operating System: Windows 7 or Above
2. Front End Programming: - TkInter

3. Back End Database: - MS EXCEL

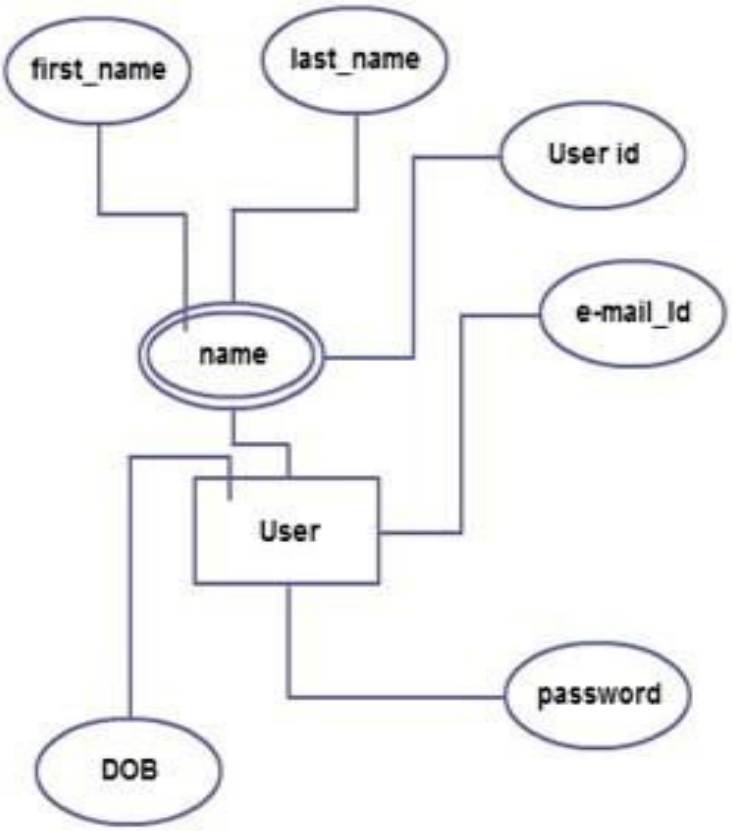
4. Modules Used: - PyMySQL, Tkinter, Chatbot

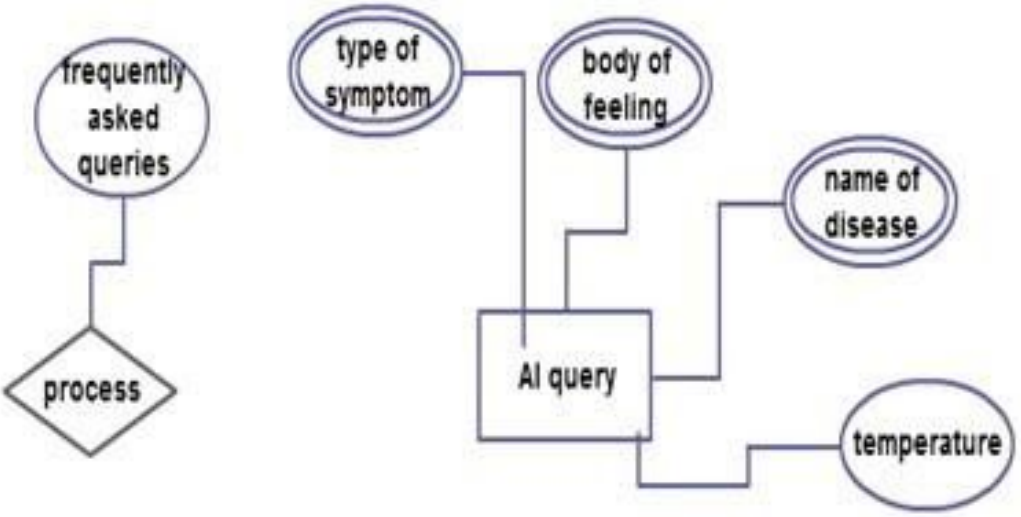
Hardware requirements

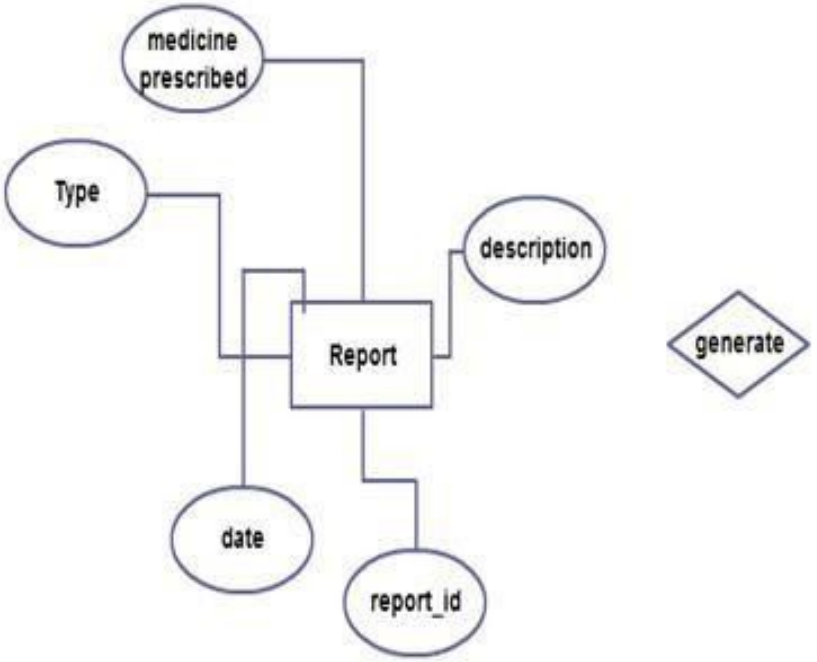
1. Intel Core (TM)2 dual core and above
2. 240 GB HD
3. Minimum 2 GB RAM
4. Standard USB Keyboard and Mouse
5. Window-7 or above

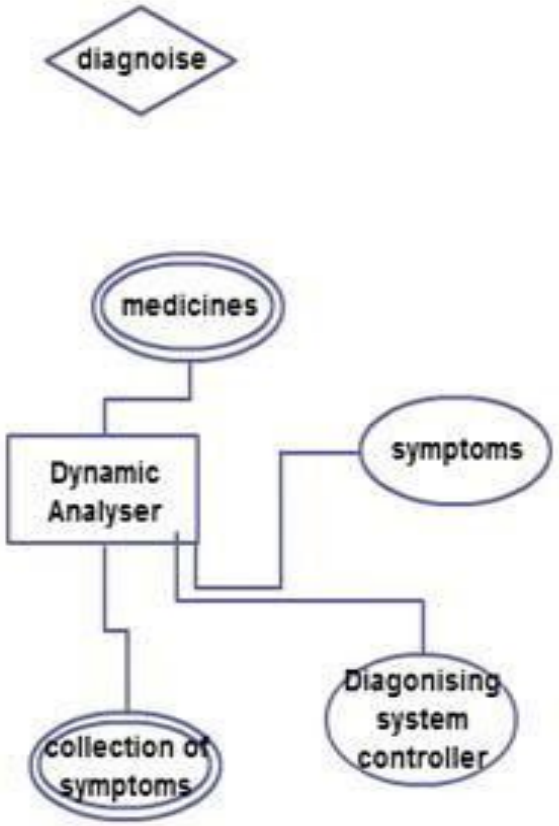
Tools: IDE Pi-Charm

2.4 DATA FLOW DIAGRAM




frequently asked queries
process
type of symptom
body of feeling
name of disease
AI query
temperature


medicine prescribed
Type
description
Report
generate
date
report_id



CHAPTER – 3

SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing

systems to satisfy specified requirements of the user. Until the 1990s, systems design had a crucial and respected role in the data processing industry. In the 1990s, standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Architectural design

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behaviour and more views of that system and analysis.

Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an overabstract model of the actual system. In the context of systems, designs are included.

Logical design includes entity-relationship diagrams (ER diagrams).

Physical design

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing requirements,
5. System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

Related Disciplines:

1. Benchmarking – is an effort to evaluate how current systems perform.

2. Computer programming and debugging in the software world, or detailed design in the consumer, enterprise or commercial world - specifies the final system components.
3. Design – designers will produce one or more 'models' of what they see a system eventually looking like, with ideas from the analysis section either used or discarded.

   A document will be produced with a description of the system, but nothing is specific – they might say 'touchscreen' or 'GUI operating system', but not mention any specific brands.
4. Requirements analysis – analyses the needs of the end users or customers.
5. System architecture – creates a blueprint for the design with the necessary structure and behaviour specifications for the hardware, software, people and data resources. In many cases, multiple architectures are evaluated before one is selected.
6. System testing – evaluates the system's actual functionality in relation to expected or intended functionality, including all integration aspects.

3.1 DESIGN METHODOLOGY

In this chapter, you will learn about the methodology for the database design stage of the database system development lifecycle for relational databases. The methodology is depicted as a bit by bit guide to the three main phases of database design, namely: conceptual, logical, and physical design.

The primary aim of each phase is as follows:

1. **Conceptual database design** - to build the conceptual representation of the database, which has the identification of the important entities, relationships, and attributes.

2. **Logical database design** - to convert the conceptual representation to the logical structure of the database, which includes designing the relations.
3. **Physical database design** - to make a decision how the logical structure is to be physically implemented (as base relations) in the target Database Management System (DBMS).

A structured approach which uses procedures, techniques, tools, and documentation help to support and make possible the process of design is called **Design Methodology**.

A design methodology encapsulates various phases each containing some stages, which guide the designer in the techniques suitable at each stage of the project. A design methodology also helps the designer to plan, manage, control, and evaluate database development and managing projects. Furthermore, it is a planned approach for analysing and modelling a group of requirements for a database in a standardized and ordered manner.

1. Conceptual Database Design:

In this design methodology, the process of constructing a model of the data is used in an enterprise, independent of all physical considerations. The conceptual database design phase starts with the formation of a conceptual data model of the enterprise that is entirely independent of implementation details such as the target DBMS, use of application programs, programming languages used, hardware platform, performance issues or any other physical deliberations.

Critical Success Factors in Database Design:

The following planning strategies are often critical to the success of database design:

1. Deal with task interactively with the users as much as possible.

2. Follow a prearranged methodology throughout the data modelling process.
3. Make use of a data-driven approach.
4. Incorporate structural and integrity considerations into the data models.
5. Combine conceptualization, normalization, and transaction validation methods into the data modelling methodology.
6. Use figures for representing as much of the data models as possible.
7. Use a Database Design Language (DBDL) to represent additional data semantics that cannot normally be represented in a diagram.
8. Build a data dictionary to add-on the data model diagrams and the DBDL.
9. Be willing to repeat steps.

These factors are constructed into the methodology that is presented for database design.

What are the steps for Conceptual Database Design?

Conceptual database design steps are:

1. Build a conceptual data model
2. Recognize entity types
3. Recognize the relationship types
4. Identify and connect attributes with entity or relationship types
5. Determine attribute domains
6. Determine candidate, primary, and alternate key attributes
7. Consider use of improved modelling concepts (optional step)
8. Check model for redundancy
9. Validate the conceptual model against user transactions
10. Review the conceptual data model with user

2. Logical Database Design:

A local logical data model is used to characterise the data requirements of one or more but not all user views of a database, and a universal logical data model represents the data requirements for all user views. The final step of the logical database design phase is to reflect on how well the model can support possible future developments for the database system.

Logical Database Design Methodology for the Relational Model

The objective of logical database design methodology is to interpret the conceptual data model into a logical data model and then authorize this model to check whether it is structurally correct and able to support the required transactions or not.

In this step of database development life cycle, the main purpose is to translate the conceptual data model created in conceptual methodology (of the previous chapter) into a logical data model of the data requirements of the enterprise. This objective can be achieved by following the activities given below:

1. Obtain the relations for the logical data model
2. Authorize those relations using normalization
3. Validate those relations against user transactions
4. Check integrity control and its limitation
5. Evaluate logical data model with user
6. Combine logical data models into the global model (This step is an optional one)
7. Check for future growth and development

The structure of the relational schema is authorized using normalization and then make sure to ensure that the relations are capable of supporting the transactions given

in the users' requirements specification. We can then check those all-important integrity constraints that are characterized by the logical data model. At this stage, the logical data model is authorized by the users to ensure that they consider the model to be a true demonstration of the data requirements for the enterprise.

3. Physical Database Design:

This physical methodology is the third and final phase of the database design methodology. Here, the designer must decide how to translate the logical database design (i.e., the entities, attributes, relationships, and constraints) into a physical database design which can ultimately be implemented using the target DBMS. As the various parts of physical database design are highly reliant on the target DBMS, there may be more than one method of implementing any given portion of the database. Consequently, to do this work appropriately, the designers must be fully aware of the functionality of the target DBMS and must recognize the advantages and disadvantages of each alternative approach for a particular accomplishment. For some systems, the designer may also need to select a suitable storage space/strategy that can take account of intended database usage.

Physical database design is the process of making a description of the execution of the database on secondary storage which describes the base relations, file organizations as well as indexes used to gain efficient access to the data and any associated integrity constraints and security measures.

The steps of the physical database design methodology are as follows:

1. Transform the logical data model for target DBMS
    a. Design base relations

b. Design representation of derived data

c. Design general constraints

2. Design file organizations and indexes

a. Analyse transactions

b. Choose file organizations

c. Choose indexes

d. Estimate disk space requirements

e. Design user views

f. Design security mechanisms

g. Consider the introduction of controlled redundancy

h. Monitor and tune the operational system

Common characteristics of a Physical Data Model

1. It typically illustrates data requirements for a single project or application.

   Sometimes even a part of an application

2. May be incorporated into other physical data models by means of a repository of shared entities

3. It typically includes 10-1000 tables; although these numbers are highly variable, depending on the scope of the data model

4. It has the relationships between tables that address cardinality and nullability (optionality) of the relationships

5. Designed and developed to be reliant on a specific version of a DBMS, storage location of data or on technology

6. Database columns will have data types with accurate precisions and lengths assigned to them. Columns will have nullability (optional) assigned

7. Tables and columns will have specific definitions

After implement the design methodology, the whole window looks like this:

3.2 USER INTERFACE DESIGN

What is user interface design?

User interface design or UI design generally refers to the visual layout of the elements that a user might interact with in a website, or technological product. This could be the control buttons of a radio, or the visual layout of a webpage. User interface designs must not only be attractive to potential users, but must also be functional and created with users in mind.

Why is user interface design important for usability?

User interface design can dramatically affect the usability and user experience of an application. If a user interface design is too complex or not adapted to targeted users, the user may not be able to find the information or service they are looking for. In

website design, this can affect conversion rates. The layout of a user interface design should also be clearly set out for users so that elements can be found in a logical position by the user.

How to optimize user interface design

User interface designs should be optimized so that the user can operate an application as quickly and easily as possible. Many experts believe that UI design should be simple and intuitive, often using metaphors from non-computer systems. With a more intuitive user interface design, users will be able to navigate around a website easily, finding the product or service they want quickly. One way to check the intuitiveness of a user interface design is through usability testing. The feedback from usability testing can then be used to optimize the user interface design of a prototype or final product.

Python GUI – tkinter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter:

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

There are two main methods used you the user need to remember while creating the Python application with GUI.

Tk(screenName=None, baseName=None, className='Tk', useTk=1):

To create a main window, tkinter offers a method to change the name of the window, you can change the className to the desired one.

The basic code used to create the main window of the application is:

m=tkinter.Tk( ) where m is the name of the main window object.

mainloop(): There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

m.mainloop( )

```
  import tkinter
m = tkinter.Tk()
  '''
  widgets are added here
  '''
  m.mainloop()
```

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1. **pack() method:** It organizes the widgets in blocks before placing in the parent widget.

2. **grid() method:** It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method:** It organizes the widgets by placing them on specific positions directed by the programmer.

There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

1. Button: To add a button in your application, this widget is used.

The general syntax is:

w=Button(master, option=value)

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the Buttons. Number of options can be passed as parameters separated by commas. Some of them are listed below:

1. **activebackground**: to set the background colour when button is under the cursor.
2. **activeforeground**: to set the foreground colour when button is under the cursor.
3. **bg**: to set he normal background colour.
4. **command**: to call a function.
5. **font**: to set the font on the button label.
6. **image**: to set the image on the button.
7. **width**: to set the width of the button.
8. **height**: to set the height of the button.

```
  import tkinter as tk
r = tk.Tk()
  r.title('Counting Seconds')
  button = tk.Button(r, text='Stop', width=25, command=r.destroy)
button.pack()  r.mainloop()
```

2. **Canvas:** It is used to draw pictures and other complex layout like graphics, text and widgets.

   The general syntax is:

   w = Canvas(master, option=value) master is the parameter used to represent the parent window.

   There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below:

   1. **bd**: to set the border width in pixels.
   2. **bg**: to set the normal background colour.
   3. **cursor**: to set the cursor used in the canvas.
   4. **highlightcolor**: to set the colour shown in the focus highlight.
   5. **width**: to set the width of the widget.
   6. **height**: to set the height of the widget.

```
  from tkinter import *
master = Tk()
  w     =     Canvas(master,     width=40,
height=60)    w.pack()    canvas_height=20
canvas_width=200  y  =  int(canvas_height
/ 2)
  w.create_line(0, y, canvas_width, y
)  mainloop()
```

3. **CheckButton:** To select any number of options by displaying number of options as as toggle buttons. The general syntax is:

w = CheckButton(master, option=value)

4. **Entry:** It is used to input the single line text entry from the user. For multi-line text input, Text widget is used.

The general syntax is:

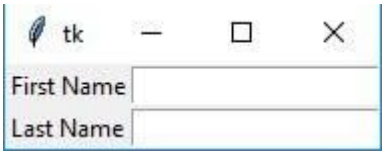
w=Entry(master, option=value) master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

1. bd: to set the border width in pixels.
2. bg: to set the normal background colour.
3. cursor: to set the cursor used.
4. command: to call a function.
5. highlightcolor: to set the colour shown in the focus highlight.
6. width: to set the width of the button.
7. height: to set the height of the button.

```
  from tkinter import *
master = Tk()
 Label(master, text='First Name').grid(row=0)
Label(master,  text='Last  Name').grid(row=1)
```

```
e1 = Entry(master)     e2 = Entry(master)
e1.grid(row=0,  column=1)      e2.grid(row=1,
column=1)  mainloop()
```



5. **Frame:** It acts as a container to hold the widgets. It is used for grouping and organizing the widgets.

   The general syntax is:

   w = Frame(master, option=value) master is the parameter used to represent the parent window.

   There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

1. **highlightcolor**: To set the colour of the focus highlight when widget has to be focused.
2. **bd**: to set the border width in pixels.
3. **bg**: to set the normal background colour.
4. **cursor**: to set the cursor used.
5. **width**: to set the width of the widget.
6. **height**: to set the height of the widget.

```
 from tkinter import *  root =
Tk()  frame = Frame(root)
frame.pack()  bottomframe =
Frame(root)  bottomframe.pack(
side = BOTTOM )
 redbutton = Button(frame, text = 'Red', fg ='red')
redbutton.pack( side = LEFT)
 greenbutton = Button(frame, text = 'Brown', fg='brown')
greenbutton.pack( side = LEFT )
```

```
  bluebutton = Button(frame, text ='Blue', fg ='blue')
bluebutton.pack( side = LEFT )
  blackbutton = Button(bottomframe, text ='Black', fg
='black')  blackbutton.pack( side = BOTTOM)  root.mainloop()
```

6. Message: It refers to the multi-line and non-editable text. It works same as that of Label.

The general syntax is:

w = Message(master, option=value) master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

1. **bd**: to set the border around the indicator.
2. **bg**: to set he normal background color.
3. **font**: to set the font on the button label.
4. **image**: to set the image on the widget.
5. **width**: to set the width of the widget.
6. **height**: to set the height of the widget.

```
  from tkinter import * main = Tk()
ourMessage ='This is our Message' messageVar =
Message(main, text = ourMessage)
messageVar.config(bg='lightgreen')
messageVar.pack( )  main.mainloop( )
```

CHAPTER – 4

TESTING

During systems testing, the system is used experimentally to ensure that the software does not fail. In other words, we can say that it will run according to its specifications and in the way users expect. Special test data are input for processing, and the results examined. A limited number of users may be allowed to use the system so that analyst can see whether they try to use it in unforeseen ways. It is desirable to discover any surprises before the organization implements the system and depends on it.

Software modules are tested for their functionality as per the requirements identified during the requirements analysis phase. To test the functionality of the file transfer and chat application, a Quality Assurance (QA) team is formed. The requirements identified during the requirements analysis phase are submitted to the QA team. In this phase the QA team tests the application for these requirements. The development team submits a test case report to the QA team so that the application can be tested in the various possible scenarios.

The software development project, errors can be injected at any stage during development i.e. if there is an error injected in the design phase then it can be detected in the coding phase because there is the product to be executed ultimately on the machine, so we employ a testing process.

During the testing the program to be tested is executed with certain test cases and output of these test cases is evaluated to check the correctness of the program. It is the testing that performs first step in determining the errors in the program.

Test Cases and Test Criteria

During Test Cases that are good at revealing the presence of faults is central to successful testing. The reason for this is that if there is a fault in the program, the program can still provide the expected behaviour on the certain inputs. Only for the set of inputs the faults that exercise the fault in the program will the output of the program devise from the expected behaviour. Hence, it is fair to say that testing is as good as its test case.

The number of test cases used to determine errors in the program should be minimum.

There are two fundamental goals of a practical testing activity: -

1. maximize the number of errors detected and.
2. minimize the number of test cases.

As these two goals are contradictory so the problem of selecting test cases is a complex one. While selecting the test cases the primary objective is to ensure that if there is an error or fault in the program, it is exercised by one of its test cases. An ideal test case is one which succeeds (meaning that there are no errors, revealed in its execution) only it there are no errors in the program one possible set of ideal test cases is one which includes all the possible inputs to the program. This is often called "exhaustive testing", however it is impractical and infeasible as even a small program can have an infinite input domain.

So, to avoid this problem we use "test criteria" in selection of the test cases. There are two aspects of the test case selection: -

1. specifying a criterion for evaluating the test cases
2. generating the set of cases that satisfy a given criteria.

The fully automated process of generating test criteria has not been yet found rather guidelines are only the automated tool available to us previously. The two fundamental properties for a testing criterion are: -

1. **Reliability**: a criterion is reliable if all the sets that satisfy the criteria detect the same error.
2. **Validity**: a criterion is valid for any error in the program if there is some set satisfying the criteria that will reveal the error.

The fundamental theorem of testing is that if a testing criterion is valid and reliable, if a set satisfying criteria succeeds then the program contains no errors.

Levels of Testing:

The basic levels of testing are unit testing, integration testing and system and acceptance testing. These different levels of testing attempt to detect different types of faults.

Code/Unit Testing:

Code testing and implementation is a critical process that can even consume more than sixty percent of the development time.

Testing:

The system development life cycle involves the phases of testing and debugging after the requirement analysis, designing and coding. The project in question was tested, debugged and implemented successfully.

Two strategies of software testing adopted for the new system are as follows:

Code testing

Specification testing

Code testing:

Code testing was carried out to see the correctness of the logic involved and the correctness of the modules. Tests were conducted based upon sample. All the modules are checked separately for assuming correctness and accuracy in all the calculations.

Specification testing:

It examines the specification stating about what program should do and how it performs under various conditions. This testing strategy is better strategy since it focuses on the way the software is expected to work.

Unit Testing:

During the phase of unit testing different constituent modules were testing against the specifications produced during the design for the modules. Unit testing is essentially for the verification of the code produced during the coding the phase, and goal is to test the internal logic of the modules. The modules once tested were then considered for integration and use by others.

Overview of Testing

1. **Testing:** Testing involves executing the program (or part of it) using sample data and inferring from the output whether the software performs correctly or not. This can be done either during module development (unit testing) or when several modules are combined (system testing).

2. **Defect Testing:** Defect testing is testing for situation where the program does not meet its functional specification. Performance testing tests a system's performance or reliability under realistic loads. This may go some way to ensuring that the program meets its non-functional requirements.

3.Debugging: Debugging is a cycle of detection, location, repair and test. Debugging is a hypothesis testing process. When a bug is detected, the tester must form a hypothesis about the cause and location of the bug. Further examination of the execution of the program (possible including many returns of it) will usually take place to confirm the hypothesis. If the hypothesis is demonstrated to be incorrect, a new hypothesis must be formed. Debugging tools that show the state of the program are useful for this, but inserting print statements is often the only approach. Experienced debuggers use their knowledge of common and/or obscure bugs to facilitate the hypothesis testing process. After fixing a bug, the system must be reset to ensure that the fix has worked and that no other bugs have been introduced. This is called regression testing. In principle, all tests should be performed again but this is often too expensive to do.

Overview of Testing Strategies:

Large systems are usually tested using a mixture of strategies. Different strategies may be needed for different parts of the system or stages of the process.

Top-down testing: This approach tests high levels of system before detailed components. This is appropriate when developing the system top-down and is likely to show up structural design errors early (and therefore cheaply). But this often has advantage that a limited, working system is available early on. Validation (as distinct from verification) can begin early. Its disadvantage is that stubs need to be generated (extra effort) and might be impracticable if component is complex (e.g. converting an array into a linked list; unrealistic to generate random list; therefore, end up

implementing unit anyway). Test output may be difficult to observe (needs creation of artificial environment). This is not appropriate for OO systems (except within a class).

Bottom-up testing: This is opposite of top-down testing. This tests low-level units then works up the hierarchy. Its advantages and disadvantages mirror those of top-down testing. In this testing there is need to write test drivers for each unit. These are as reusable as the unit itself. Combining top-down development with bottom-up testing means that all parts of system must be implemented before testing can begin, which does not accord with incremental approach discussed above. Bottom-up testing is less likely to reveal architectural faults early on. However, bottom-up testing of critical low-level components is almost always necessary. Appropriate for OO systems.

Stress testing: Tests system's ability to cope with a specified load (e.g. transactions per second). Tests should be planned to increase load incrementally. This type of testing goes beyond design limit until system fails (this test is particularly important for distributed systems like checking degradation of performance a s network traffic increases).

Back-to-back testing: Comparison of test results from different versions of the system (e.g. compare with prototype, previous version or different configuration). Process – Run first system, saving test case results. Run second system, again saving its results. Compare result files. The key point to be noted is that no difference does not mean no bugs. Both systems may have made the same mistake.

Defect testing: A successful defect test is a test that causes the system to behave incorrectly. Defect testing is not intended to show that a program meets its specification.

If tests do not show up defects it may mean that the tests are not exhaustive enough. Exhaustive testing is not always practically applicable. Subset has to be defined (this should be part of the test plan, not left to the individual programmer).

4.1 TESTING TECHNIQUE AND TESTING STRATERGIES

There are two fundamental approaches to testing they are:

FUNCTIONAL TESTING/BLACK BOX TESTING:

In functional testing the structure of the program is not considered. Test cases are solely determined on the basis of requirement or specification of the program or module.

Internals of the modules and the programs are not considered for selection of test cases. Due to this nature it is also called "black box testing".

The most obvious functional testing procedure is "exhaustive testing", which is impractical. The other criteria for generating the test case are to generate them "randomly" this strategy has a little chance of resulting in a test case that is close to optimal.

There is no appropriate criterion for developing the test case so we have certain heuristic approaches-:

a) **Equivalence Class Partitioning**: in which the domain of all inputs is divided into a set of equivalence classes, so that if any test in that class succeeds, then every test in that class will succeed i.e. the success of one set element implies the success of the other.

b) **Boundary Value Analysis**: It has been observed that programs work correctly for a set of values in an equivalence class fail on certain values. These values generally lie on the boundary of equivalence class. So, in the boundary value analysis we chose

an input for a test case from an equivalence class, such that input lies on the edge of equivalence class. Boundary value test cases are also called "extreme cases".

c) **Cause-Effect Graphing**: The problem with the prior approaches is that they consider each input separately i.e. both concentrate on classes and conditions of one input. The combination of inputs is not considered which is used in many cases. One way to exercise the combination of various input conditions is to consider all the valid combinations of the equivalence class of the input conditions. The technique starts with identifying the causes and the effect of system under testing.

STRUCTURAL TESTING:

In the structural approach the test cases are generated the basis of the actual code of the program or the module to be tested, this structural approach is sometimes called "glass box testing", This testing is concerned with the implementation of the program. The content is not to exercise the various input conditions rather different programming structures and data structures used in the program.

There are three different approaches to structural testing they are-

a) **Control flow-based criteria**:

Most common structure-based criteria use control flow-based testing in which the control flow graph of the program is considered and coverage of various aspects of graph are specified as criteria. The various control flow-based strategies are

- statement coverage,
- branch coverage and
- all path coverage

b) **Data Flow based testing:**

The basic idea behind the data flow-based testing is to make sure that during testing, the definitions of variables and their subsequent use is tested. To implement the data flow-based testing the data flow graph is first made from the control flow graph.

c) **Mutation Testing:**

In the above two testing techniques the focus is on which path to be executed, but mutation testing is not a path-based approach. The mutation testing requires that the set of test cases must be such that they can distinguish between the original programs and is mutants. In software world there is no fault model as in hardware so most of the techniques do guess work regarding where the fault must lie and then select the test cases to reveal those faults. In Mutation testing faults of some pre-decided types are introduced in the program being tested. Then those faults are found in the mutants.

The aim of the system testing process was to determine all defects in our project. The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not.

4.2 DEBUGGING AND CODE IMPROVEMENT

In ideal worlds, all programmers would be so skilled and attentive to detail that they would write bug-free code. Unfortunately, we do not live in an ideal world. As such, debugging, or tracking down the source of errors and erroneous result, is an important task that all developers need to perform before they allow end-user to use their applications. We will discuss some techniques for reducing the number of bugs in code up front.

There are three categories of bugs

a) Syntax error:

These errors occur when code breaks the rule of the language, such as visual Basic sub statement without a closing End sub, or a forgotten closing curly brace ({}) in c#. Theses error the easiest to locate. The language complier **or** integrated development environment (IDE) will alert you to them and will not allow you to compile your program until you correct them.

b) Semantic error

These errors occur in code that is correct according to rules of the compiler, but that causes unexpected problems such as crashes or hanging on execution. A good example is code that execute in a loop but never exists the loop, either because the loop depends on the variable whose values was expected to be something different than it actually was or because the programmer forgets to increment the loop counter. Another category of errors in this area includes requesting a field from a dataset, there is no way to tell if the field actually exists at compile time. these bugs are harder to detect and are one type of running error.

c) Logic error

Logic errors are like semantic errors, logic errors are runtime error. That is, they occur while the program is running. But unlike semantic errors, logic errors do not cause the application to crash or hang. Logic error results in unexpected values or output. This can be a result of something as simple as a mistyped variables name that happens to match another declared variable in the program. This type of error can be extremely difficult to track down to eliminate.

Preventing Debug Write readable code

Develop and make consistent use of naming and coding standards. It not that important which standard we use, such as Hungarian notation or Pascal, Casing (First Name) or other naming conventions, as long as we use one. We should also strive for consistency in our comments and encourage liberal commenting code.

Create effective test plan

The only effective way to eliminate logic error is to test very path of your application with every possible data value that a user could enter. This is difficult to manage without effective planning. We should create our test plan at the same time we are designing the application, and we should update these plans as you modify the application design.

Maintenance is necessary to eliminate errors in the working system during its working life and to tune the system to any variations in its working environment often small system deficiencies are found as a system is brought into operations and changes are made to remove them.

CHAPTER – 5

IMPLEMENTATION

Implementation is the process of having systems personnel check out and put new equipment into use, train users, install the new application and construct any files of data needed to use it. This phase is less creative than system design. Depending on the size of the organization that will be involved in using the application and the risk involved in its use, systems developers may choose to test the operation in only one area of the firm with only one or two persons. Sometimes, they will run both old and new system in parallel way to compare the results. In still other situations, system developers stop using the old system one day and start using the new one the next.

Evaluation of the system is performed to identify its strengths and weaknesses. The actual evaluation can occur along any of the following dimensions:

1. **Operational Evaluation**: Assessment of the manner in which the system functions, including case of use, response time, overall reliability and level of utilization.
2. **Organizational Impact**: Identification and measurement of benefits to the organization in such areas as financial concerns, operational efficiency and competitive impact.
3. **User Manager Assessment:** Evaluation of the attitudes of senior and user manager within the organization, as well as end-users.
4. **Development Performance**: Evaluation of the development process in accordance with such yardsticks as overall development time and effort, conformance to budgets and standards and other project management criteria.

LOGIN MENU: A login is a set of credentials used to authenticate a user. Most often, these consist of a username and password. However, a login may include other information, such as a PIN number, passcode, or passphrase. Some logins require a biometric identifier, such as a fingerprint or retina scan.

Logins are used by websites, computer applications, and mobile apps. They are a security measure designed to prevent unauthorized access to confidential data. When a login fails (i.e., the username and password combination does not match a user account), the user is disallowed access. Many systems block users from even trying to log in after multiple failed login attempts.

Examples of logins include:

1. **Operating System login** – Windows and Mac systems can be configured to require a login in order to use the computer after it is turned on or woken from

sleep mode. A login may also be required to install software or modify system files.

2. **Website login** – Webmail interfaces, financial websites, and many other sites require a username and password in order to access account information.
3. **App store login** – App stores like Google Play and Apple's App Store require a login to download mobile apps, music, and other files.
4. **FTP login** – file transfer programs often require a login in order to browse, send, and receive files from an FTP server.
5. **Router login** – Wired and wireless routers typically require an administrator login to modify the settings.

At a basic level, logins make user accounts possible. Most systems require unique usernames, which ensures every user's login is different. On a more advanced level, logins provide a security layer between unsecured and secure activity. Once a user logs in to a secure website, for example, all data transfers are typically encrypted. This prevents other systems from viewing or recording the data transferred from the server.

SIGNUP MENU: The Sign-up module has been developed using the Tkinter GUI framework written in Python. It facilitates the user to save his/her data into the Oracle database. The signup form will be explicitly used to insert the records of doctors who will be using the disease prediction system. The doctor’s data has been scraped from Internet for research purposes. The signup module opens the prediction window for a legitimate user and displays a message box in case of failed authentication.

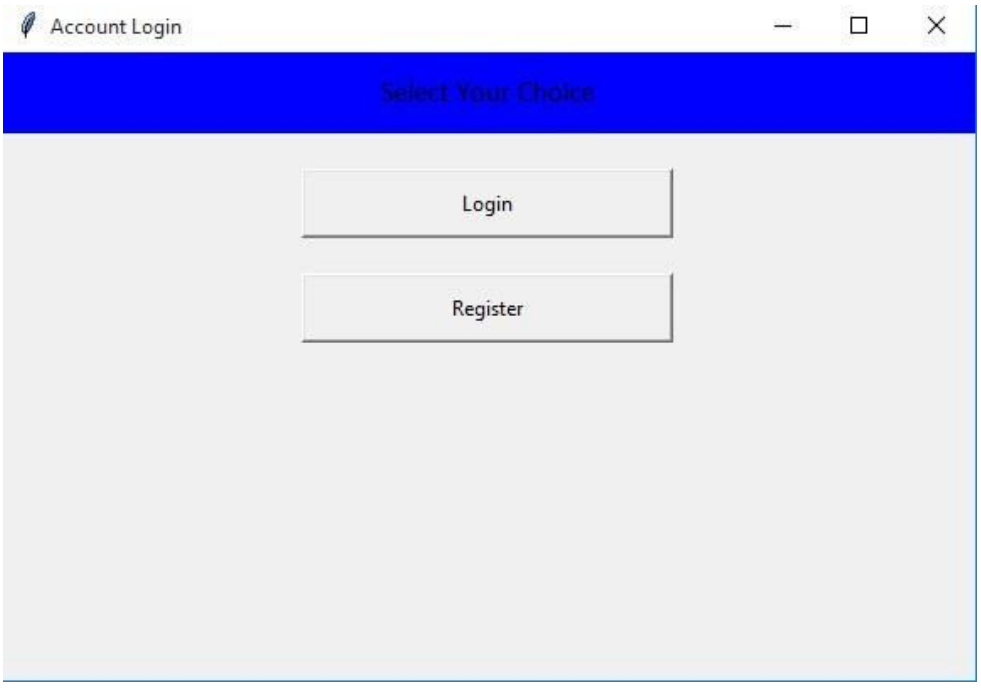
PREDICTION WINDOW: The prediction window can be accessed only by legitimate

doctors saved in our portal. It provides a start button to start the analysis process of the symptoms. The prediction window poses questions in the form of symptoms to the user and expects a binary response which can be provided by pressing the yes or no buttons. The system records the entries of the user and finally gives a prediction. It also describes the symptoms inserted along with the symptoms expected. The prediction window also displays a dynamic link to book an appointment of the specialized doctor suggested by the model.

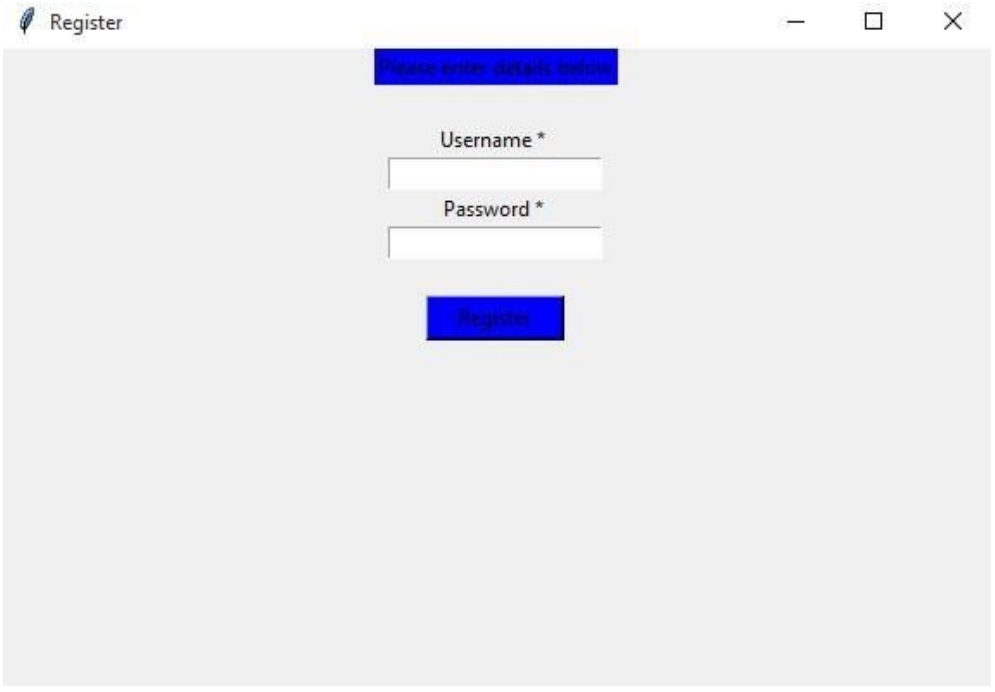
SYMPTOMS WINDOW: The Symptoms window is created or called at run time when the user is inserting the symptoms into the model. When the model is satisfied with an appropriate number of inputs. It then generates a response in the form of the predicted disease, symptoms given, confidence interval and the recommendation for the doctor to visit next. The symptoms window also provides a link to book an appointment with the concerned doctor which can copy and pasted into the browser by the user for further operations.

5.1 SYSTEM IMPLEMENTATION

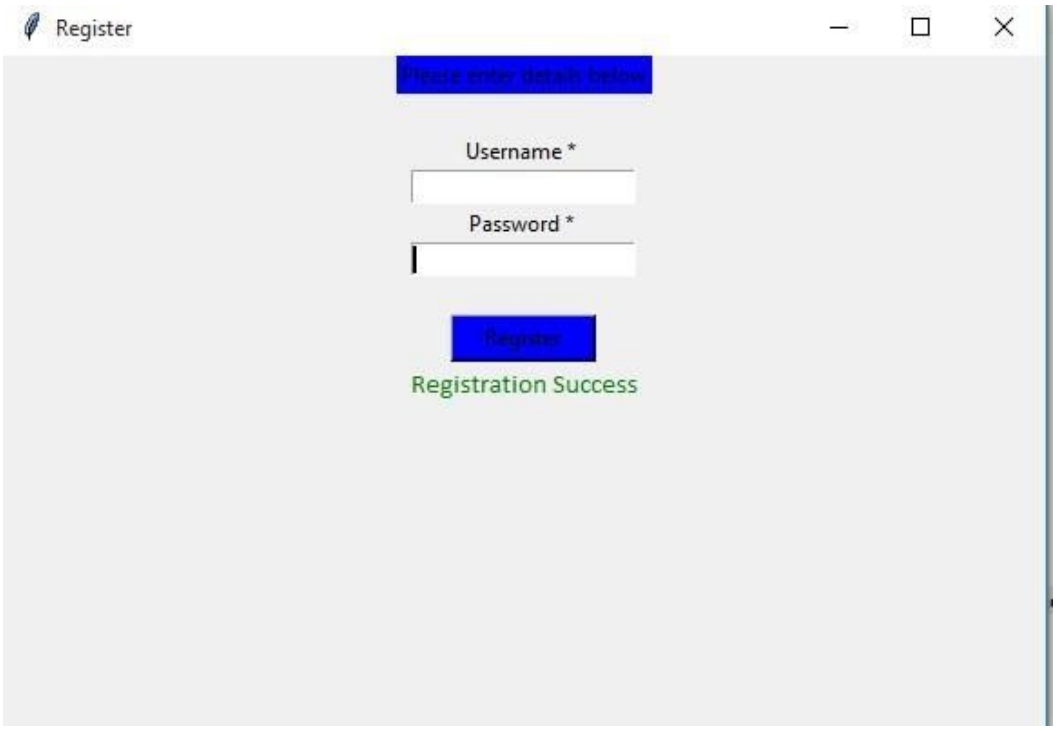
1. When someone execute this window application, this page will be open first. Then if user have account then they will login otherwise they will have to register first.



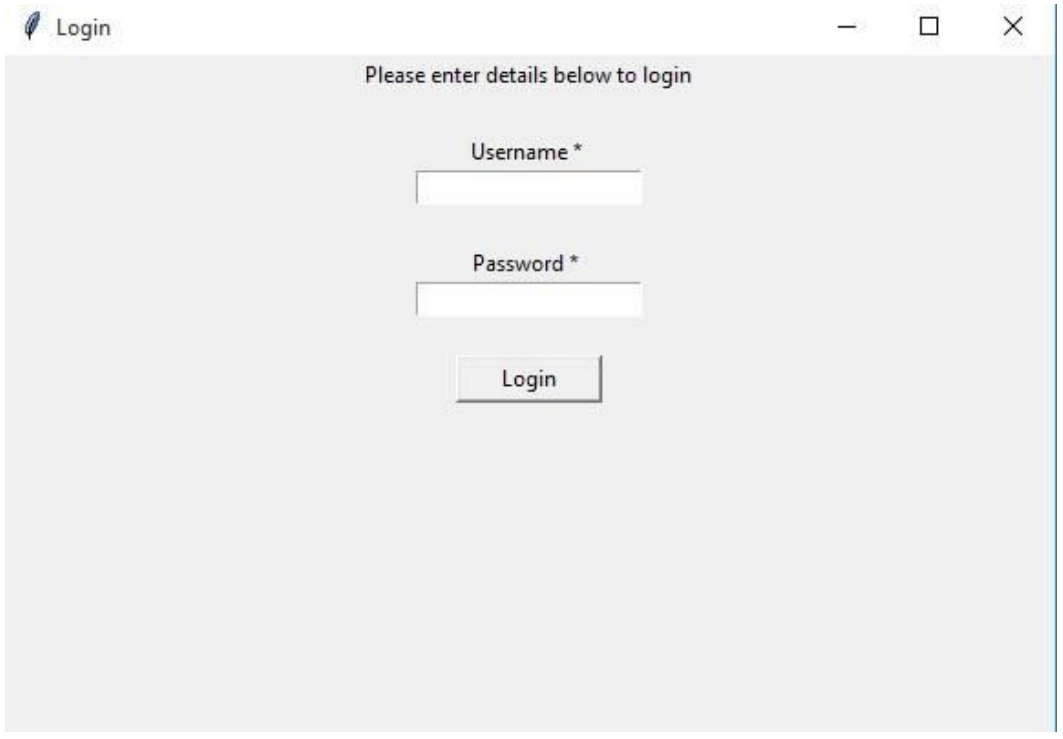
2. SIGNUP WINDOW: - To create an account, they will have to give a unique user name and solid password.



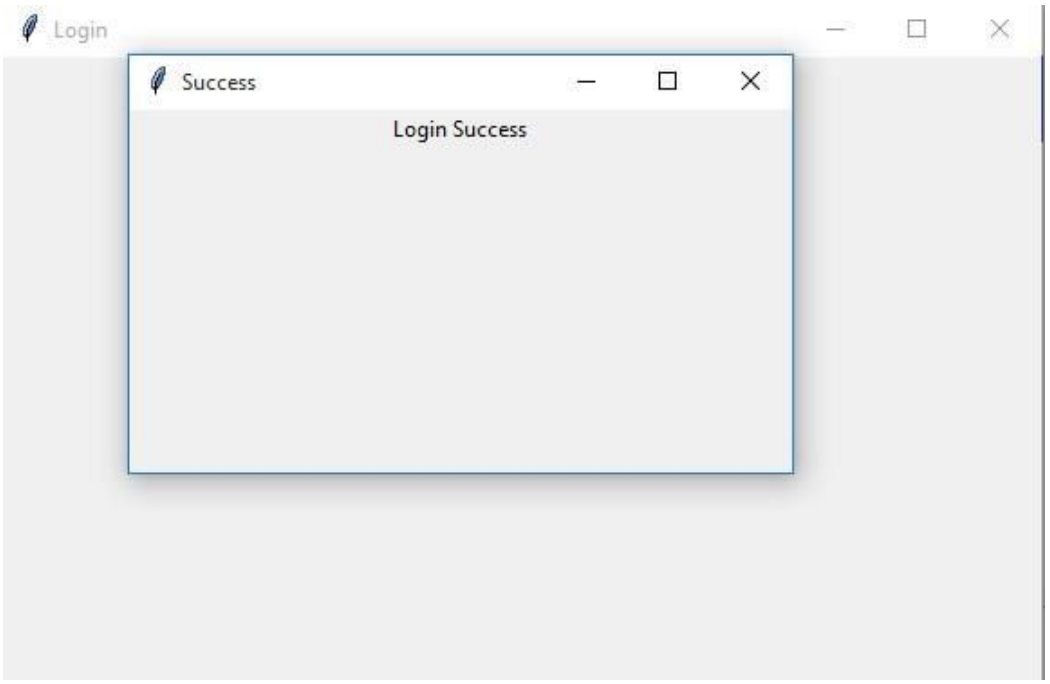
3. SIGNUP SUCCESSFULLY: - This window will tell user that registration is successfully done or account will be created successfully.



4. LOGIN WINDOW:- If user has an account then they can give its username and password to this window and can move to next phase.

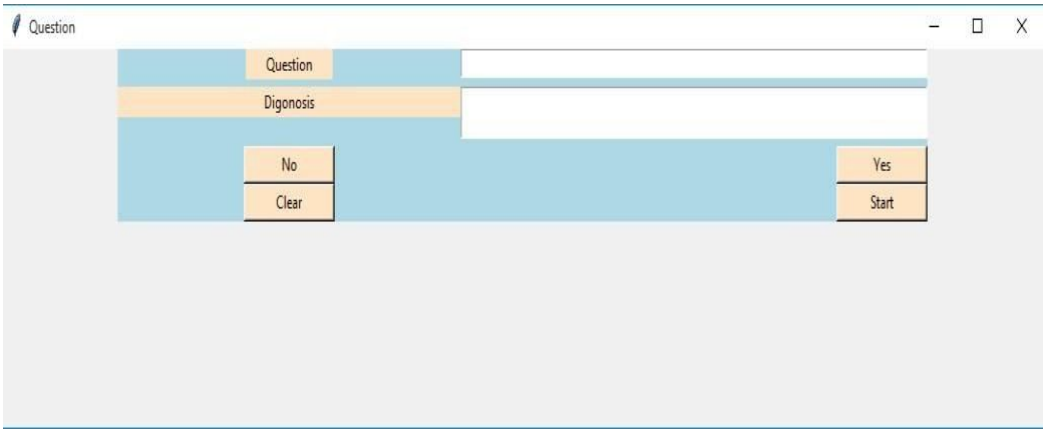


5. LOGIN SUCCESSFUL:- When we give an username and password correct then other window will open to tell that login will be successfully happen.



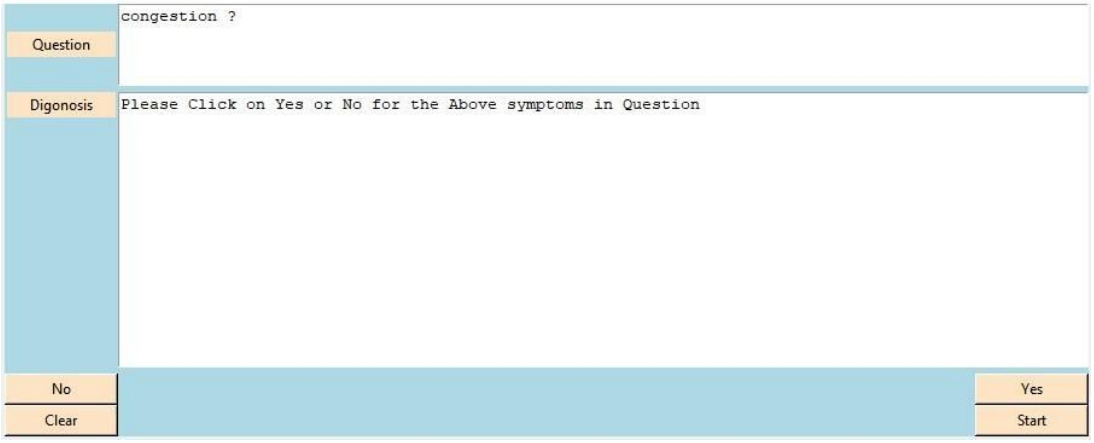
6. START WINDOW: - After registration or login, when we have an account then we can easily visit to this window.

   Then one can start easily by clicking on START button.



7. SYMTOMS WINDOW: - After clicking on start button, then we can see in our screen some type of Question, which is called as Symptoms of a patient.

They will have to just click on button that is Yes or No.



8. CLEAR WINDOW: - It will clear all the Question from the Screen.

9. CONSOLE BASED: In this, we have to write Yes or No only.

   If our Symptoms are not matched then we have to write no in our screen.

   When our Symptoms will be matched then we just have to write yes.

```
Please reply with yes/Yes or no/No for the following symptoms
slurred_speech ?

no
pain_behind_the_eyes ?

no
receiving_blood_transfusion ?

no
red_spots_over_body ?

no
unsteadiness ?
```

10. SYMPTOMS WINDOW: - When we write Yes on our console screen, then our matched problem will be found on screen. And it will also tell the Symptoms which may a patient have.

```
increased_appetite ?

yes
['You may have Diabetes ']

symptoms present  ['increased_appetite']

symptoms given ['fatigue', 'weight_loss', 'restlessness', 'lethargy',
'irregular_sugar_level', 'blurred_and_distorted_vision', 'obesity',
'excessive_hunger', 'increased_appetite', 'polyuria']
```

11. SUGGESTION BY MODEL: - After all this, Model will suggest the doctor and also provide his/her link where one can easily visit with the help of chrome etc.

```
The model suggests:

Consult  ['Dr. Anshul Gupta']

Visit  ['https://www.practo.com/delhi/doctor/dr-anshul-gupta-ear-nose-
throat-ent-specialist-1?specialization=Ear-Nose-Throat%20(ENT)
%20Specialist&practice_id=712546']

In [4]:
```

5.2 SOFTWARE IMPLEMENTATION

Software implementation begins with the effort of software fabrication. Fabrication is an act of making something. Software fabrication involves programmatic design, source code editing or programming, and testing of each software unit. This series of technical tasks represents how software procedures, routines, modules, objects, or graphical models are produced. Each software unit is presumed to be suitable for their intended purpose or role in the overall architectural context. The result of software fabrication should be a documented unit of source code that has been tested against its structural unit specification. This source code (software unit) is then available to be assembled, integrated, and compiled with other fabricated software elements to craft larger software components. These integrated software components are tested against structural component specifications to ensure their correctness. This assembly, integration, and testing series of events continues to generate larger, more complex software components. Software integration progresses until a completely integrated and tested software configuration item is realized and available for acceptance testing.

Verify and Validate

Like any other software implementation, newly developed logic must be properly tested. The challenge with data projects is to make sure to compile a comprehensive list of data scenarios, evaluate data integrity throughout their life cycles, and make maintenance as easy as possible. Usually, software testing focuses on validating functional requirements, and not necessarily other data management aspects. Granted, functional testing will require looking at many data scenarios, which are sure to test many of the aspects pertaining to data. But how the integrity of the information will be

validated, how data will stay consistent with time, and how data quality is maintained are often left untested. This particular issue can cause degradation over time.

To address these issues, develop a data-driven approach. It was explained earlier in this chapter how a solid data model is fundamental to creating a scalable, high-quality, and easily maintained data system. Make sure to test those aspects of the integrated domains. This challenge increases as more domains are added. Data models become more complex when modelling relationships across multiple domains and when multiple business teams are using the master data. As data are integrated into the multi-domain MDM model, many aspects of such integration must be validated.

Verification and validation of data integration projects must be done from a technical point of view, as well as from the standpoint of business usage of data. As always, the question concerns data's fitness for use. But data can be fit for use when they are initially migrated but become less so as time passes. If the model doesn't preserve data integrity properly over time, or if error prevention is not a focus during data entry, data quality can easily decline and affect the data's fitness for use. Unit, integration, and acceptance testing will not necessarily uncover those issues. Data architects, data modelers, and data stewards need to work together to test for those scenarios and methods for maintaining high-quality data.

Software Implementation Process:

The first step is building a business case, defining your end users and analysing how they will benefit. After careful cost-benefit analysis, the next step is to find capable back-end support. This can be filled internally or outsourced with a software

development provider. You must also decide custom or off-the-shelf, educating yourself on drawbacks of each.

Boil down the project to the most granular details possible prior to kick-starting the campaign. The more robust the project specifications, the less discrepancies will exist between your vision and the final deliverable.

When the codes complete, educate your staff on how to integrate the new IT system into their pre-existing processes. An uneducated staff results in underutilization and debilitates ROI. Collect feedback from your staff and incrementally evolve the software to better fit your needs.

Python Implementations

Python currently has three production-quality implementations, known as CPython,

JPython, and IronPython, and several other experimental implementations, such as PyPy. This book primarily addresses CPython, the most widely used implementation, which I refer to as just Python for simplicity. However, the distinction between a language and its implementations is an important one.

Python programming language has its 4 type of implementations:

1) Python
2) CPython
3) JPython
4) IronPython

5.3 SOFTWARE INSTALLATION

5.3.1 Python Software Installation Process:

To get started working with Python 3, you'll need to have access to the Python interpreter. There are several common ways to accomplish this:

1. Python can be obtained from the **Python Software Foundation** website at python.org. Typically, that involves downloading the appropriate **installer** for your operating system and running it on your machine.
2. Some operating systems, notably Linux, provide a **package manager** that can be run to install Python.
3. On macOS, the best way to install Python 3 involves installing a package manager called **Homebrew**. You'll see how to do this in the relevant section in the tutorial.
4. On mobile operating systems like Android and iOS, you can install apps that provide a Python programming environment. This can be a great way to practice your coding skills on the go.

Alternatively, there are several websites that allow you to access a Python interpreter online without installing anything on your computer at all.

Windows

It is highly unlikely that your Windows system shipped with Python already installed. Windows systems typically do not. Fortunately, installing does not involve much more than downloading the Python installer from the python.org website and running it. Let's take a look at how to install Python 3 on Windows:

Step 1: Download the Python 3 Installer

1. Open a browser window and navigate to the Download page for

Windows at python.org.

2. Underneath the heading at the top that says **Python Releases for Windows**, click on the link for the **Latest Python 3 Release - Python 3.x.x**. (As of this writing, the latest is Python 3.6.5.)
3. Scroll to the bottom and select either **Windows x86-64 executable installer** for 64bit or **Windows x86 executable installer** for 32-bit. (See below.)

Sidebar: 32-bit or 64-bit Python?

For Windows, We can choose either the 32-bit or 64-bit installer. Here's what the difference between the two comes down to:

1. If our system has a 32-bit processor, then you should choose the 32-bit installer.
2. On a 64-bit system, either installer will actually work for most purposes. The 32-bit version will generally use less memory, but the 64-bit version performs better for applications with intensive computation.
3. If we're unsure which version to pick, go with the 64-bit version.

Note: Remember that if we get this choice "wrong" and would like to switch to another version of Python, we can just uninstall Python and then re-install it by downloading another installer from python.org.

Step 2: Run the Installer

Once we have chosen and downloaded an installer, simply run it by double-clicking on the downloaded file. A dialog should appear that looks something like this:

Important: we want to be sure to check the box that says **Add Python 3.x to PATH** as shown to ensure that the interpreter will be placed in your execution path.

Then just click **Install Now**. That should be all there is to it. A few minutes later we should have a working Python 3 installation on your system.

CONCLUSION & FUTURE SCOPE

A part from planning major task of preparing the implementation is education of users. The more complex system is implemented, the more involved will be the system analysis and design effort required just for implementation. The most critical stage is in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain types of transaction while using the new system.

Here we are concluding that health care service is the major requirement for the healthy life of a patient so we have provided this way to help them. Here Model can predict the diseases with higher accuracy and its prediction is helping the patients to save their time in caring their diseases.

FUTURE SCOPE:

1. **Better Efficiency in Delivering Services**:

   Project management provides a "roadmap" that is easily followed and leads to project completion. Once we know where to avoid the bumps and pots holes it stands to reason that we're going to be working smarter and not harder and longer.

2. **Improved / Increased / Enhanced Customer Satisfaction**:

Whenever we get a project done on time and under budget, the client walks away happy. And a happy client is one we'll see again. Smart project management provides the tools that enable this client/manager relationship to continue.

3. **Enhanced Effectiveness in Delivering Services**:

The same strategies that allowed you to successfully complete one project will serve you many times over.

4. **Greater Standing and Competitive Edge**:

This is not only a good benefit of project management within the workplace but outside of it as well; word travels fast and there is nothing like superior performance to secure your place in the marketplace.

5. **Better Flexibility**:

Perhaps one of the greatest benefits of project management is that it allows for flexibility. Sure, project management allows you to map out the strategy you want to take see your project completed. But the beauty of such organization is that if you discover a smarter direction to take, you can take it. For many small-to-midsize companies, this alone is worth the price of admission.

6. **Increased Risk Assessment**:

When all the players are lined up and your strategy is in place potential risks will jump out and slap you in the face. And that's the way it should be. Project management provides a red flag at the right time: before you start working on project completion.

7. **Increase in Quantity**:

we saved the best for last and increase in quality is often the result of better efficiency, a simple reminder regarding the benefits of project management.

REFRENCES

- [1]Harini D K1, Natesh M2 ," PREDICTION OF PROBABILITY OF DISEASE BASED ON SYMPTOMS USING MACHINE LEARNING ALGORITHM",International Research Journal of Engineering and Technology (IRJET),vol.05 Issue: 05,may 2018.
- [2]Pravin Shinde and Prof. Sanjay Jadhav,"Health Analysis System Using Machine Learning", International Journal of Computer Science and Information Tachnologies, Vol. 5(3) ,2014, 3928-3933 .
- [3]Sayali Ambekar and Dr. Rashmi Phalnikar" , DISEASE PRIDICTION BY USING MACHINE LEARNING", International Journal of Computer Science and Applications, Vol. XII ,Special Issue, May 18, ISSn 2321-3469 .
- [4]Dhaval Raval , Dvijesh Bhatt , Malaram K Kumhar , Vishal Parikh , Daiwat Vyas.
, "Medical Diagnosis System Using Machine learning" ,International Journal of Computer Science Vol. 7 Issue,March 2015 pp.177-182.
- [5]Mrs.J.Sukanya and Vijaya Kumar ,"Applications of Big Data Analytics and Machine Learning Techniques in Health Care Sector" , International Journal Of Engineering And Computer Science ,Vol. 6 Issue 7 July 2017.

➢ [6]Vikas Sir(Director).”DirecotorMessage”.cetpainfotech (accessed Aug. 19, 2019). ➢ [7]make me analyst.”statical data analysis.makemeanalyst.com

➢ [8]Sudhir Rawat.”Understand Azure data Factory”.it-books (accessed Aug. 19, 2019) .

➢ [9]arthur Samuel.”Machine Learning”.en.wikipedia.org (accessed Aug. 19, 2019) .

➢ [10]Developer.”Machine Learning”.stackoverflow.com (accessed Aug. 19, 2019)