# Finger Gesture Analysis System

AN PROJECT SUBMITTED
IN PARTIAL – FULFILMENT OF THE REQUIREMENT
FOR THE AWARD OF THE DEGREE
OF

## BACHELOR OF SCIENCE

(COMPUTER APPLICATION)

BY

**ILIYAS ANSARI**
**(18B511507)**

**DEPARTMENT OF CA.**
**DR. SHYAMA PRASAD MUKHERJEE UNIVERSITY**
**MORHABADI, RANCHI -834008**
**(2018 - 2021)**

# DECLARATION CERTIFICATE

This is to certify that the work presented in the Assignment entitled **"FINGER GESTURE ANALYSIS SYSTEM"** in partial fulfillment of the requirement for the award of degree of **Bachelor of Science in Computer Applications**, of Dr. Shyama Prasad Mukherjee University, Morabadi Ranchi, is an authentic work carried out under my supervision.

To the best of my knowledge, the content of this assignment form a basis for the award of any previous Degree to anyone else.

Date: - 11/09/2021

RANJAY KUMAR
Assistant Professor
Dept. of CA&IT
DSPMU, Ranchi

**Co-ordinator Dept.Of CA &IT**
Dr. Shyama Prasad Mukherjee University

## CERTIFICATE FOR PROJECT

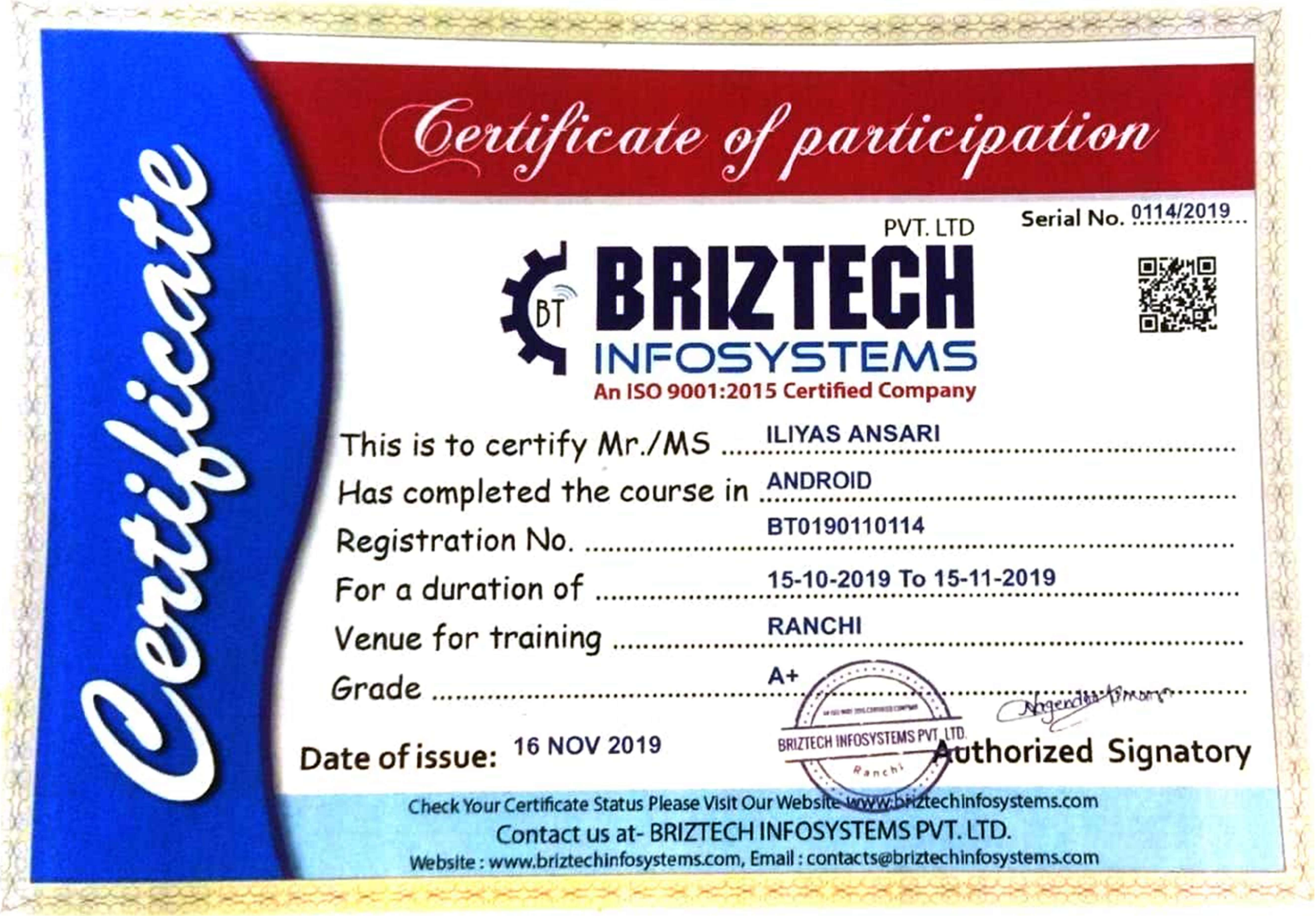
This is to certify that this is a bona fide record of the project work entitled **“FINGER GESTURE ANALYSIS SYSTEM”** done satisfactory at “D.S.P.M.U” by **Iliyas Ansari (18B511507), in** partial fulfillment of B.Sc (CA) Examination.

This report or similar report on the topic has not been submitted for any other examination and doesn’t form part of any other course undergone by the candidate.

**(Internal Examiner)** **(External Examiner)**

**UNDER THE GUIDANCE OF**

**RANJAY KUMAR**



Certificate

# Certificate of participation

Serial No. 0114/2019

BRIZTECH INFOSYSTEMS PVT. LTD

An ISO 9001:2015 Certified Company

This is to certify Mr./MS ILIYAS ANSARI

Has completed the course in ANDROID

Registration No. BT0190110114

For a duration of 15-10-2019 To 15-11-2019

Venue for training RANCHI

Grade A+

Date of issue: 16 NOV 2019

An ISO 9001:2015 Certified Company
BRIZTECH INFOSYSTEMS PVT. LTD.
Ranchi

Authorized Signatory

Check Your Certificate Status Please Visit Our Website www.briztechinfosystems.com

Contact us at- BRIZTECH INFOSYSTEMS PVT. LTD.

Website : www.briztechinfosystems.com, Email : contacts@briztechinfosystems.com

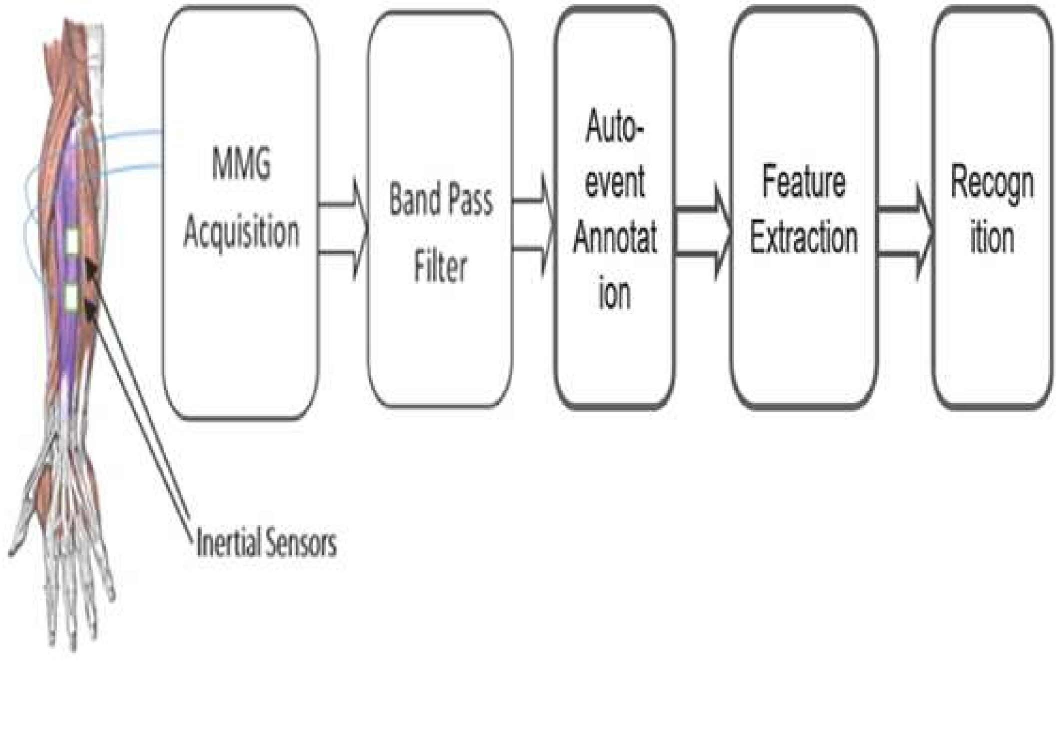
# Introduction

Modeling and recognizing human hand gesture is an extremely important research topic, and it is the core of any intelligent human–computer interaction system with applications in automatic control, virtual reality, augmented reality, human–robotic interaction, and computer animation. In addition, it attracts more attentions and interests in biomedical engineering recently, e.g., prosthesis control and wearable rehabilitation system. As it is difficult to model the gesture accurately from images and the gesture appearance varies a lot, it is still a difficult task to recognize and track hand gestures. On the other hand, stroke is the leading cause of disability in adults worldwide. Exercise disorders as the most common sequelae of stroke can seriously affect normal activity and quality of life. Exercise and training have long been used to restore motor function after stroke. Among that, hand movement as a kind of fine action is more difficult to recover. Finger gesture recognition technique can be used to train and lead the patients to do the rehabilitation exercises, which normally requires real-time interaction and long-term monitoring. Thus, portable system with real-time processing technique is a valuable topic to pay attention and be explored in the stroke application area.

Currently, most hand-gesture recognition systems are using either hand-motion-based (HMB) technologies or motion-intent-based (MIB) technologies. HMB methods are capable of directly detecting the tracking of hand motions. Non-skin attached sensors such as optical sensors, inertial sensors, and force sensors (1) are usually deployed in HMB technologies. Many commercial products, such as Kinect and Data Glove, are built on HMB technologies. However, the performance of HMB-based hand-gesture recognition system degrades when the lighting variations present. The recognition accuracy will also drop significantly if skin colors are too similar to background colors or the usage is in dark environment. In addition, the coverage of the system is limited by the sensors' measurement range and attaching sensors to fingers and hands make user hand movements unsmoothly and uncomfortable.

## System Architecture

The proposed system relies on two MMG signal channels for identifying movements of five fingers. The architect of the proposed FGR system is shown in Figure 1. The first module contains the MMG acquisition system obtaining the two-channel MMG signals from the forearm muscle by the inertial sensor. Then, the detected MMG signals are going through a band-pass filter to reduce noise distortion. The third stage of the system consists of the tapping event detection (TED) algorithm to extract the MMG signal segments with the muscle activity information. The next stage is the feature extraction process on the obtained MMG signal segments. Finally, different classifiers including SVM, KNN, and NBC are built for the recognition purpose based on the features extracted



## Feature Selection

There are 32 features from each MMG channel after the feature compression stage. However, these features have different degrees of relevance for the hand-motion classification. The MMG signal obtained by the MMG acquisition system contains both the mechanical signal generated by the hand motions and noises from different sources such as electrical noise and the vibration noise from the surroundings. The features containing the noise usually have an adverse effect for the recognition. It should be noted that a larger number of SVD features do not always lead to more effective classification results . Therefore, a feature ranking algorithm is required to select the most relevant features to further improve the recognition results.

### *NBC*

NBC, as a subclass of Bayes classification algorithm, is widely applied in the pattern recognition because of its simplicity and effectiveness. In some applications, its performance is comparable with other classifiers with increased computational complexity . Naive Bayes classification model assumes that each feature is independent for the classification. Although this unrealistic assumption limits its scope of applications, the time and space complexities are reduced. Therefore, the NBC is still a popular choice in various applications . The implementation of NBC is based on the reference .

### *KNN*

KNN is a classification model based on statistical analysis. The KNN algorithm classifies the sample according to the class of its k-nearest neighbor samples in the feature space. And the sample is assigned to the class where most of its neighbor samples belong to. KNN is a simple classifier without priori statistical knowledge. Thus, it is widely used in applications with non-normal or unknown sample distributions . However, its performance will be degraded when the number of samples is large . The implementation of KNN is based on the reference .

### *SVM*

SVM is a machine learning algorithm based on the statistical learning theory. It is good to deal with the situations when small samples are available and high dimension pattern recognition task is given . The libSVM3.12 package  is adopted for programming implementation in our system

## Experimental Procedure

All participants are instructed to use five fingers of their dominant hand to tap, including thumb tapping, index finger tapping, middle finger tapping, ring finger tapping, and little finger tapping. Following a metronome with 30 beats per minute, every participant is asked to tap five fingers one by one from thumb to little finger, and each finger gesture is repeated five times before moving to next finger. The above process is repeated five times with 5 min interval to avoid the unreliable samples due to large number of consecutive tappings on the same finger. Therefore, there are totally 125 finger-gesture movements (5 finger gestures × 5 times × 5 rounds) for every participant. A total of 25 movements on each finger are obtained from tapping with either short time interval or long time interval which increases the variety of data set.

## FUNCTIONAL REQUIREMENTS

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

- Functional Requirements concerns with the specific functions delivered by the system.

So, Functional requirements are statements of the services that the system must provide.

- The functional requirements of the system should be both complete and consistent

- Completeness means that all the services required by the user should be defined.

• Consistency means that requirements should not have any contradictory definitions.

• The requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in details, its inputs and outputs, exceptions and so on.

• Take user id and password match it with corresponding file entries. If a match is found then continue else raise an error message.

## NON-FUNCTIONAL REQUIREMENTS

• Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools.

• The non-functional requirements can be built on the basis of needs of the user, budget constraints, organization policies and etc.

1. **Performance requirement:** All data entered shall be up to mark and no flaws shall be there for the performance to be 100%.
2. **Platform constraints:** The main target is to generate an intelligent system to predict the adult height.
3. **Accuracy and Precision**: Requirements are accuracy and precision of the data
4. **Modifiability:** Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).
5. **Portability:** Since mobile phone is handy so it is portable and can be carried

and used whenever required.

6. **Reliability**: Requirements about how often the software fails. The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error Prediction, and a strategy for correction.
7. **Security**: One or more requirements about protection of your system and its data.
8. **Usability:** Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

### *ACCESSIBILITY:*

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the cloud can access the cloud to store and retrieve their data with the help of a secret key sent to their email ids. User interface is simple and efficient and easy to use.

### *MAINTAINABILITY:*

In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using .net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

### *SCALABILITY:*

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

### *PORTABILITY:*

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

### *VALIDATION:*

It is the process of checking that a software system meets specifications and that it fulfils its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements, not as specification artefacts or as needs of those who will operate the software only; but, as the needs of all the stakeholders.

## HARDWARE REQUIREMENTS

- System : Pentium 4, Intel Core i3, i5, i7 and 2 GHz Minimum
- RAM : 512Mb or above
- Hard Disk : 10 GB or above
- Input Device : Keyboard and Mouse
- Output Device : Monitor or PC

## SOFTWARE REQUIREMENTS

- Operating System : Windows 7, 10 or Higher Versions
- Platform : Jupiter Notebook
- Front End : Python Tkinter
- Back End : Python and Files
- Programming Lang : Python

## DESIGN GOALS

The Design goals consist of various design which we have implemented in our system disease prediction using machine learning. This system has built with various designs such as data flow diagram, sequence diagram, class diagram, use case diagram, component diagram, activity diagram, state chart diagram, deployment diagram. After doing these various diagrams and based on these diagrams we have done our project.

We have designed our system in such a way that whenever user log in into the system, the user has to register to the system, and new user cannot use the system without registering in the system. After that for registration the user requires basic credentials such as username, age, email, phone, password. Then the user has to login to the system using the same username and password. Here are the things that this system can perform.

a. Entering Symptoms

b. Disease Prediction

**Entering Symptoms**: Once user successfully logged in to the system then he/she has to select the symptoms from the given drop-down menu.

**Disease prediction:** The predictive model predicts the disease of a person he might have, based on the user entered symptoms.

.

## SYSTEM ARCHITECTURE

Disease prediction using machine learning predicts the presence of the disease for the user based on various symptoms and the information the user gives such as sugar level, haemoglobin level and many more such general information through the symptoms. The architecture of the system disease prediction using machine learning consist of various datasets through which we will compare the symptoms of the user and predicts it, then the datasets are transformed into the smaller sets and from there it gets classified based on the classification algorithms later on the classified data is then processed into the machine learning technologies through which the data gets processed and goes in to the disease prediction model using all the inputs from the user that is mentioned above. Then after user entering the above information and overall processed data combines and compares in the prediction model of the system and finally predicts the disease. An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. The diagram explains about the system software in perception of overview of the system.

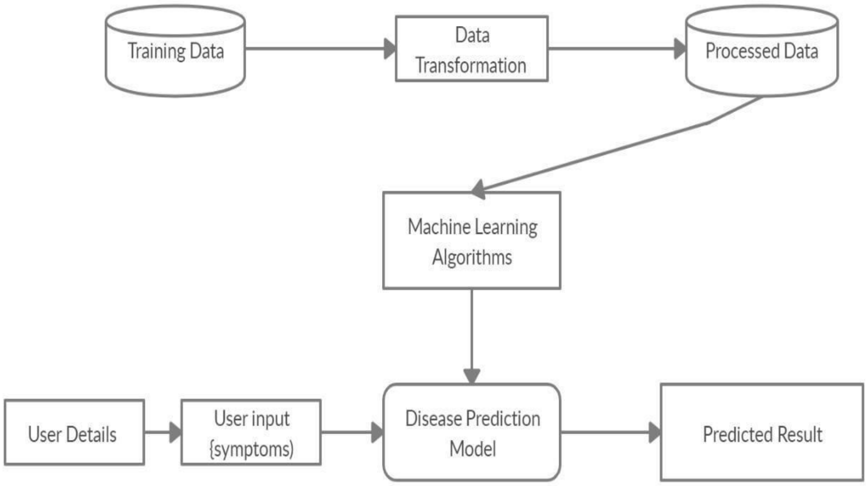
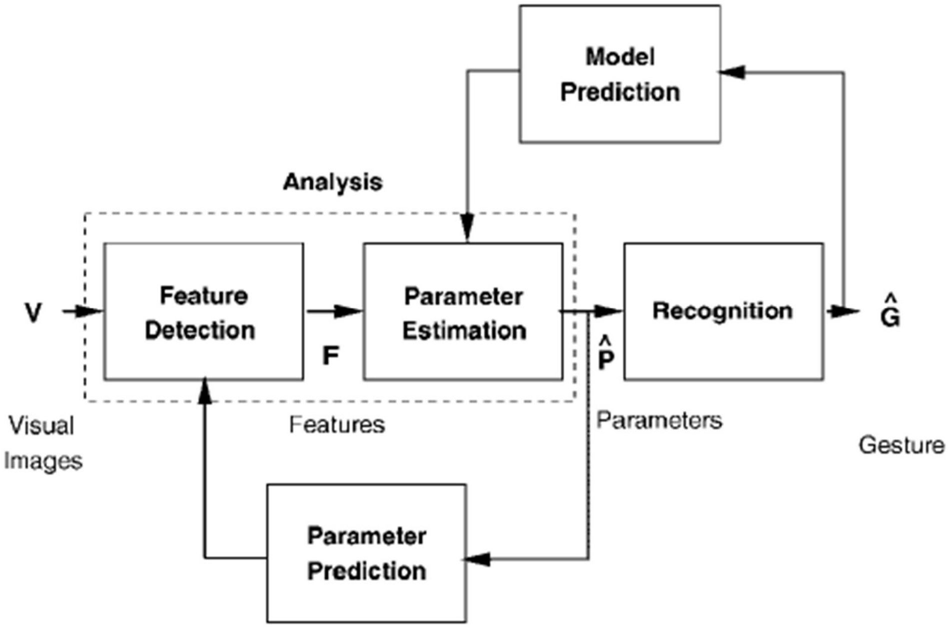


**Fig System Architecture**

## DATA FLOW DIAGRAM

The dataflow diagram of the project disease prediction using machine learning consist of all the various aspects a normal flow diagram requires. This dataflow diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information.



## USE CASE DIAGRAM

The Use Case diagram of the project Finger analysis using machine learning consist ofall the various aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the use case diagram of all the entities are linked to each other where the user gets started with the system.

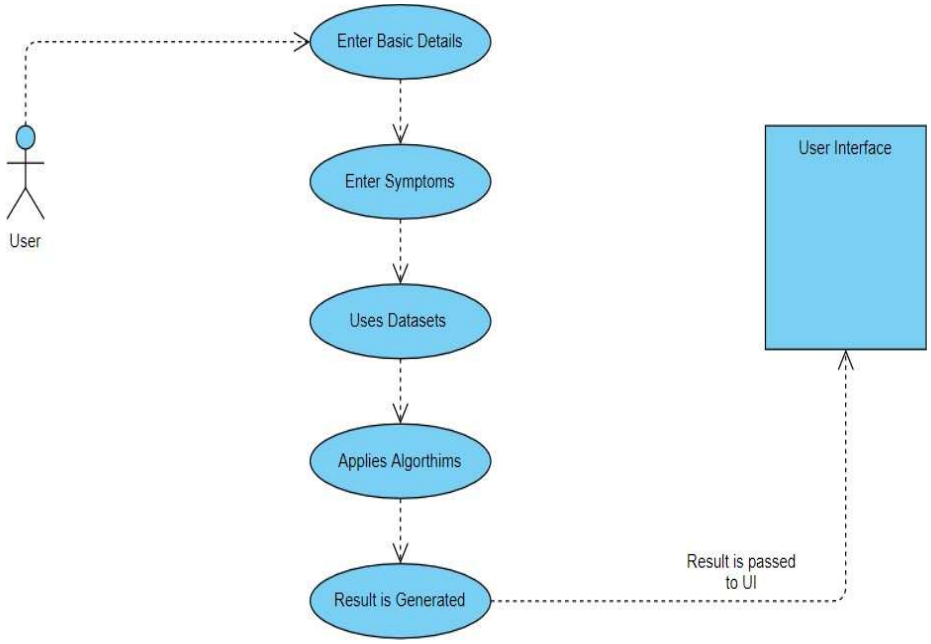


**Fig Use Case Diagram**

## DEPLOYMENT DIAGRAM

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modelling the physical aspects of an object-oriented system. Here the deployment diagram show the final stage of the project and it also shows how the model looks like after doing all the processes and deploying in the machine. Starting from the system how it processes the user entered information and then comparing that information with the help of datasets, then training and testing those data using the algorithms such as decision tree, naïve Bayes, random forest. Then finally processing all those data and information the system gives the desired result in the interface.

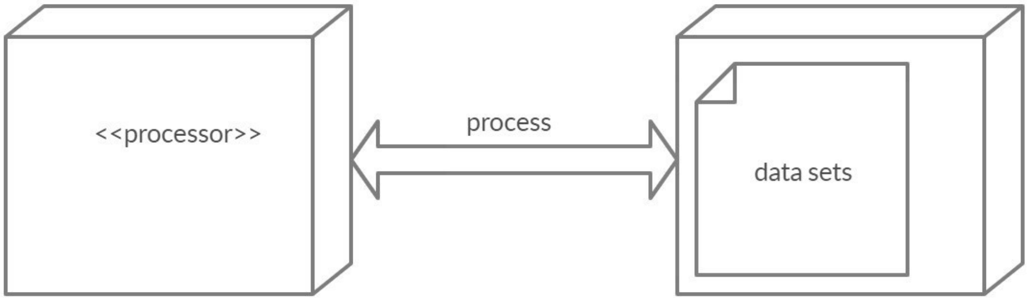


**Fig Deployment Diagram**

## INTERFACE AND FRAMEWORK DIAGRAM

- Below is the structure which we will use in our project Disease Prediction using Machine learning.
- The User Interface of this system consists of Python's library interface called tkinter.
- Then it goes into the framework model where all the actions and services are combined and then the result is processed.
- It also consists of file system where all the user related information is stored such as username, password, age, phone, email.
- Below is the structure of the User Interface along with necessary implementations.

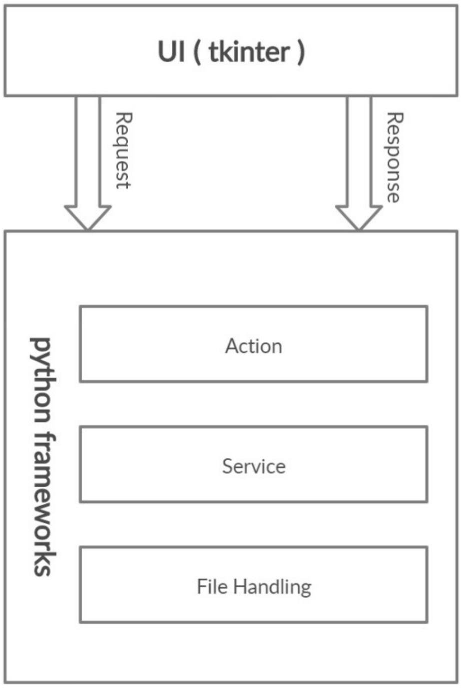
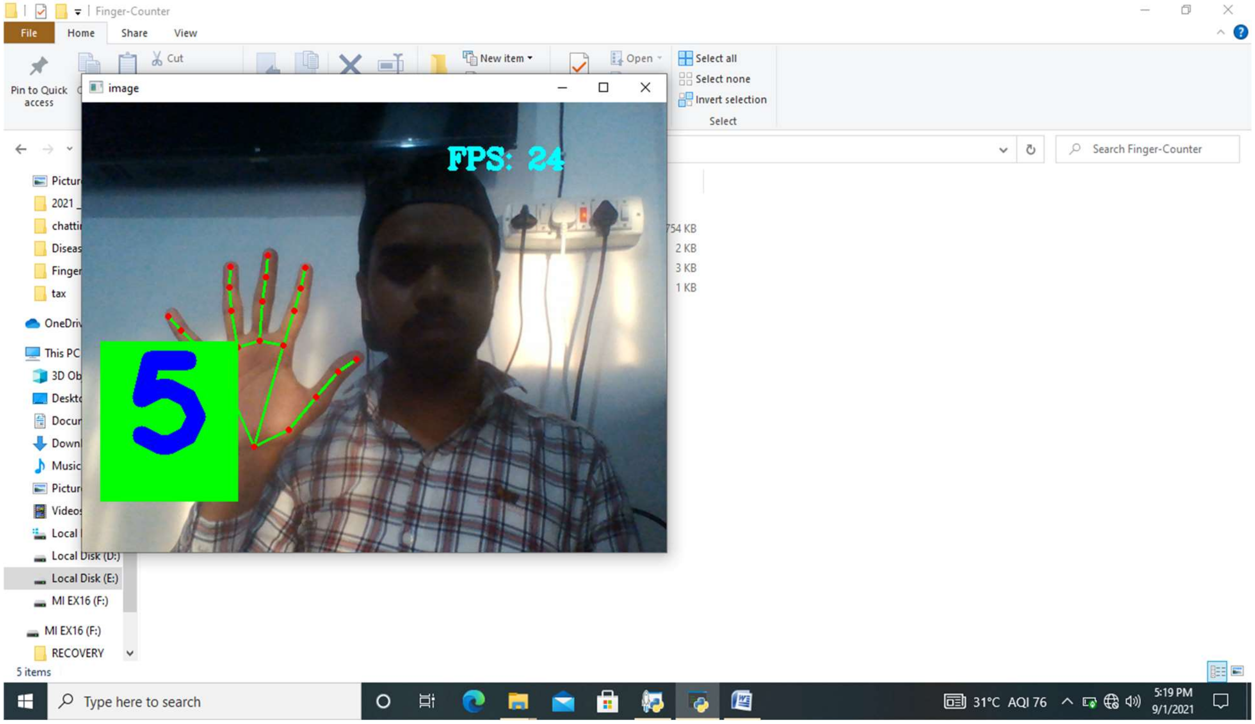


**Fig Interface and Framework Diagram**

- After the User Interface it consist of the framework in which the system works accordingly using all the technologies, algorithms and various tools in which the project works accordingly.
- The framework consists of all the modules starting from the data preparation, data building and assessment stage.
- All these three factors are then going into the data collection phase, where the data is classified accordingly using the appropriate models and algorithms such as decision tree, naïve bayes, random forest.
- Then all those algorithms use the datasets and it forms the sets where all the previous data is stored, then using that data it compares with the new data and result is generated.
- Then pre-processing work will happen to reduce and analyze the data that is present in the system.
- Then with the help of UI the data is transferred into the main screen.
- Then later all those data are analyzed and validated then the final result is generated.
- Finally after user enters the symptoms, all backend mechanisms works and the predicted result is displayed in the User Interface.



```
# -*- coding: utf-8 -*-

"""

Created on Mon Sep  6 23:09:19 2021

"""

import cv2

import os

import time

import handTrackingModule as htm
```

```
def getNumber(ar):

   s=""

   for i in ar:

     s+=str(ar[i]);

     if(s=="00000"):

      return (0)

   elif(s=="01000"):

      return(1)

   elif(s=="01100"):

      return(2)

   elif(s=="01110"):

      return(3)

   elif(s=="01111"):

      return(4)

   elif(s=="11111"):

      return(5)

   elif(s=="01001"):

      return(6)
```

```
    elif(s=="01011"):

        return(7)

 wcam,hcam=640,480

cap=cv2.VideoCapture(0)

cap.set(3,wcam)

cap.set(4,hcam)

pTime=0

detector = htm.handDetector(detectionCon=0.75)

while True:

    success,img=cap.read()

    img = detector.findHands(img, draw=True )

    lmList=detector.findPosition(img,draw=False)

    #print(lmList)

    tipId=[4,8,12,16,20]

    if(len(lmList)!=0):

        fingers=[]

        #thumb

        if(lmList[tipId[0]][1]>lmList[tipId[0]-1][1]):
               fingers.append(1)

        else :

               fingers.append(0)

        #4 fingers

        for id in range(1,len(tipId)):

            if(lmList[tipId[id]][2]<lmList[tipId[id]-2][2]):

               fingers.append(1)

            else :

               fingers.append(0)

        cv2.rectangle(img,(20,255),(170,425),(0,255,0),cv2.FILLED)

        cv2.putText(img,str(getNumber(fingers)),(45,375),cv2.FONT_HERSHEY_PLAIN,

                                   10,(255,0,0),20)

         cTime=time.time()

    fps=1/(cTime-pTime)

    pTime=cTime

    cv2.putText(img, f'FPS: {int(fps)}',(400,70),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,0),3)

    cv2.imshow("image",img)

    if(cv2.waitKey(1) & 0xFF== ord('q')):
        break# -*- coding: utf-8 -*-
"""
Created on Mon Sep  6 23:09:19 2021
"""
import cv2
import os
import time
import handTrackingModule as htm
def getNumber(ar):
    s=""
    for i in ar:
       s+=str(ar[i]);

    if(s=="00000"):
        return (0)
    elif(s=="01000"):
        return(1)
    elif(s=="01100"):
        return(2)

    elif(s=="01110"):

        return(3)

    elif(s=="01111"):

        return(4)

    elif(s=="11111"):

        return(5)

    elif(s=="01001"):

        return(6)

    elif(s=="01011"):

        return(7)

 wcam,hcam=640,480

cap=cv2.VideoCapture(0)

cap.set(3,wcam)

cap.set(4,hcam)

pTime=0

detector = htm.handDetector(detectionCon=0.75)

while True:
```

```
success,img=cap.read()

img = detector.findHands(img, draw=True )

lmList=detector.findPosition(img,draw=False)

#print(lmList)

tipId=[4,8,12,16,20]

if(len(lmList)!=0):

    fingers=[]

    #thumb

    if(lmList[tipId[0]][1]>lmList[tipId[0]-1][1]):

            fingers.append(1)

    else :

            fingers.append(0)

    #4 fingers

    for id in range(1,len(tipId)):



        if(lmList[tipId[id]][2]<lmList[tipId[id]-2][2]):

            fingers.append(1)
            else :

                fingers.append(0)

         cv2.rectangle(img,(20,255),(170,425),(0,255,0),cv2.FILLED)

        cv2.putText(img,str(getNumber(fingers)),(45,375),cv2.FONT_HERSHEY_PLAIN,

                                    10,(255,0,0),20)

         cTime=time.time()

    fps=1/(cTime-pTime)

    pTime=cTime

    cv2.putText(img, f'FPS: {int(fps)}',(400,70),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,0),3)

    cv2.imshow("image",img)

    if(cv2.waitKey(1) & 0xFF== ord('q')):

        break

import cv2

import mediapipe as mp

import time

 class handDetector():

    def __init__(self, mode=False, maxHands=2, detectionCon=0.5, trackCon=0.5):

        self.mode = mode
        self.maxHands = maxHands
        self.detectionCon = detectionCon
        self.trackCon = trackCon
        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands,
                                        self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils
    def findHands(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.hands.process(imgRGB)
        #print(results.multi_hand_landmarks)

        if self.results.multi_hand_landmarks:
            for handLms in self.results.multi_hand_landmarks:
                if draw:
                    self.mpDraw.draw_landmarks(img, handLms,
                                               self.mpHands.HAND_CONNECTIONS)
        return img
```

```
def findPosition(self, img, handNo=0, draw=True):
 lmList = []
       if self.results.multi_hand_landmarks:
           myHand = self.results.multi_hand_landmarks[handNo]
           for id, lm in enumerate(myHand.landmark):
               # print(id, lm)
               h, w, c = img.shape
               cx, cy = int(lm.x * w), int(lm.y * h)
               # print(id, cx, cy)
               lmList.append([id, cx, cy])
               if draw:
                   cv2.circle(img, (cx, cy), 15, (255, 0, 255), cv2.FILLED)
     return lmList
def main():
   pTime = 0
   cTime = 0
   cap = cv2.VideoCapture(1)
   detector = handDetector()
```

```
    while True:
        success, img = cap.read()
        img = detector.findHands(img)
        lmList = detector.findPosition(img)
        if len(lmList) != 0:
            print(lmList[4])
        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime
        cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 255), 3)
        cv2.imshow("Image", img)
        cv2.waitKey(1)
if __name__ == "__main__":
    main()
```

# TESTING

In the ideal circumstances a software engineer design an computer , a system, or a product with "Testability" in mind. This enables the individuals charged with testing to design effective test cases more easily. But what is "Testability"? James Bach describes testability in the following manner:-

Software testability is simply how easily a computer program can be tested. Since testing is so profoundly difficult, it pays to know what can be done to it.

Sometimes programmers are willing to do things that will help the testing process and a checklist of possible design points, features etc. can be useful in negotiation with them. There are certainly metrices that could be used to measures testability in most of its aspects. Sometimes, testability is used to measure testability in most of aspects. Sometimes, testability is used to mean how adequately a particular set of tests will cover the product. Its also used by the military to mean how easily a tool xan be checked and repaired in the field. These two meanings are not the same as software testability. The checklist that follows provides a set of characteristics that lead to testable software.

## 1. BLACK BOX TESTING

Black box testing, also called behavioral testing, focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of inputs conditions that fully exercise all functional requirements for the program. Black box testing is not an alternative to white box techniques. Rather, it is a complementry approach that is likely to uncover a different class of errors then white box methods.

Black box testing attempts to find errors in the following categories:

a) Incorrect or missing function

b) Interface errors

c) Errors in data structures or external database access

d) Behavior or performance errors, and

e) Initialization or terminaton errors

When computer software is considered, black box testing alludes to tests that are conducted at the software interface. Although they are designed to uncover errors, black box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that the integrity of external information is maintained. A black box test examines some fundamental aspect of a system with regard for the internal logical structure of the software.

We have tested each and every function in the project and come to the conclusion that they all have executed correctly. At the time when the project was in progress a number if functions where deleted and then added due to which errors where generated. This errors where removed time to time in the testing phase where I had to go in the depth study of the project. Its each and every parts where executed at least 100 times to see wheter the working was anticipating the expected results or not.

Tests where Designed to answer the following questions?

- How is functional validity tested?
- How is system behavior and performance tested?
- What classes of inputs will make good test cases?
- Is the system particularly sensitive to certain input values?
- How are the boundaries of a data volume can the system tolerate?
- What effects will specific combinations of data have on system operations?

By applying Black Box technique, we derive a set of answers for the questions:

- The functional validity is checked by the execution of the functions one by one and fulfilling the requirements of the functions as per the demand of the software. The functional validity is checked by using suitable technique which lies on the code optimization, efficiency in executions.

- System behavior is tested by certain tools and performance tests lies on the dumping of large bulk of data into the application software database. The application database consists of data's with their appropriate width. The validation of the data's linking of one data's with the other's everything are tested before inputting them into the application software database.

- Interactive classes of input are generally better for the good test classes .

- The system is sensitive for the system data because the whole application software is a time domain software. Any error on the time parameter may crash the whole application. Time is taken as an important parameter for all the calculations. This is why this application software is sensitive to system time.

- Boundary of data classs are isolated by giving the range of the arrays. Its totally a static memory allocated. Hence, any error on the static array may crash the whole applications.

- The specific can tolerate 50MB of the memory of the data without any problem.

- The specific combinations of the data of center id and time. The application program has no control over time and id. Both are generated automatically.

## ALPHA AND BETA TESTING

It is virtually impossible for a software developer to foresee how the user will really use a program. Instructions for use may be misinteperated, strange combinations of data may be regularly used; output that seemed clear to the tester may be unintelligible to the user in the filed.

When software is built for one person, a series of acceptance tests are conducted to enable the user to validate all requirements. Conducted by the end-user rather than software engineers, an acceptance test can range from an informal “test-drive” to a planned and systematically executed series of tests. In fact, acceptance testing can be conducted over a period of time thereby uncovering errors that might degrade the system overtime.

If software is developed as a product to be used by many peoples, it is impractical to perform acceptance tests with one another. Most software builders use a process called alpha and beta testing to uncover error that only the end user seems able to find. The alpha test is conducted at the developer site by a patients. The software is used in a natural setting with a developer “looking over the shoulder” of the users and recording errors and usage problems. Alpha tests are conducted in an controlled environment. The beta test is conducted at one or more patients site by the end user of software. Unlike alpha testing, the developer is generally not present. Therefore the beta test is a “live” application of the software in an environment that cannot encountered in beta testing a reports these to the developer at regular interval. As a result of problems reported during beta tests. Software engineers make modifications and then prepare for the release of the software product to the entire employee base. I this stage alpha and beta testing where also done to determine the working of the software i.e.to check the validation of the software, whether the results which it is generating is valid or not.

Alpha Testing

When user software is build for one employee a serial of dependent tests ate conducted to enable the users to validate the requirement conducted by an user rather than software engineers. And acceptance test can from an informal to applied systematically. This test is conducted at developer site by a user the software is used in a natural setting with the developers as we can say-“lookin over the soldier” of the user and recording errors alpha test are conducted then a control.

Beta Testing

The beta testing is conducted at one or more users’ side by the user of the software. Unlike alpha testing the developer generally is not present therefore Beta testing is live application of the software that controlled by the developer at the result the software engineerings may modify and then prepare for release of the software.

## SECURITY MECHANISMS

Any user, who is authorized by the administrator to operate the software, receives a unique user ID and a password with the help of which he can log into the system. To login the user goes through the system menu. He is then presented with the login box where he has to enter the username and password to login.

In this software, the login Id and password which is created for the login for the operation is strictly maintained in the database by the automatic automation. This way, we can easily find out at what time which user login to handle the process which is done by the specified work. If there will be any problem by the handling of the money and the transaction, we can check is there any user is doing anything wrong or not. The database maintained when the user login for the transaction or for any other purpose, it shows time and date in which period he/she was working at that time. Same user can log on many times after taking logout from the software.

In this security features user may belongs to any of the two categories:-

1. Authorized User 2. Administration

First when the software is installed in the organisation the software developer provide the user Id and password for the different work for different people. Like we provide the user Id and password to staff of the hospital to process some work but not all, same way the administrator get all the features for the processes. He can do anything as he wants to do. Administrator provides the work for the staff of the hospital as per their requirement.

The user logging in may access only certain features within the software. A user will only be allowed to view the records and opening different types of features and maintaining the database, providing the requirement of the user as they demanded. Every user get the unique and password. Every new user will be assigned a unique name along with a password by the administrator. The list of all users is maintained by the administrator of the software.

Once the administrator logs into the system, he has access to all the features within the system other than all essential details, the administrator has a separate responsibility which includes managing users and their details. So if  a new user has to be added, this can be done by the administrator. Therefore we have seen that a person who is logging in as a user who does not exists in the user logging in is not a valid user that is the user is not available in the table.

When sometimes the user logging in gives a incorrect password then message box is displayed which informs the user that the password enterd is not correct and may try again. Any user, who is authorized by the administrator to operate the software, receives a unique user Id and a password with the help of which he can log into the system. To login the user goes through the SYSTEM menu. He’s then presented with the login box where he has to enter the username and password to login.

Cost Estimation of the Project

- Predicting the resources required for a software development process

Topics covered

- Feasibility Analysis
- Productivity measures
- Estimation techniques
- Algorithmic cost modelling
- Project duration

Feasibility Analysis

Feasibility: – the measure of how beneficial or practical an information system will be to an organization.

Feasibility analysis: – the process by which feasibility is measured.

Three Tests for Feasibility

Technical feasibility: – a measure of the practicality of a technical solution and the availability of technical resources and expertise.

Economic feasibility: - a measure of the cost-effectiveness of a project or solution.

Operational feasibility – a measure of how well a solution will work or be accepted in an organization.

Economic Feasibility = Cost-Benefit Analysis

Costs:

- Development costs are one time costs that will not recur after the project has been completed.
- Operating costs are costs that tend to recur throughout the lifetime of the system. Such costs can be classified as:
- Fixed costs — occur at regular intervals but at relatively fixed rates.
- Variable costs — occur in proportion to some usage factor.

Benefits:

- Tangible benefits are those that can be easily quantified.
- Intangible benefits are those benefits believed to be difficult or impossible to quantify.

Three Popular Techniques to Assess Economic Feasibility

- Payback Analysis
- Return On Investment
- Net Present Value

The Time Value of Money is a concept that should be applied to each technique. The time value of money recognizes that a dollar today is worth more than a dollar one year from now.

Software cost components

- Hardware and software costs
- Travel and training costs
- Personnel costs (the dominant factor in most projects)
- salaries of engineers involved in the project
- Social and insurance costs
- Must also take project overhead into account
- costs of building, heating, lighting
- costs of networking and communications
- Costs of shared facilities (e.g. library, staff restaurant, etc.)

Fundamental estimation questions

- How much effort is required to complete an activity?
- How much calendar time is needed to complete an activity?
- What is the total cost of an activity?
- Project estimation and scheduling are interleaved management activities

Costing and pricing

- Estimates are made to discover the cost, to the developer, of producing a software system
- There is not a simple relationship between the development cost and the price charged
- Broader organisational, economic, political and business considerations influence the price charged

Productivity measures

- Size related measures based on some output from the software process. This may be lines of delivered source code, object code instructions, etc.
- Function-related measures based on an estimate of the functionality of the delivered software. Function-points are the best known of this type of measure

Measurement problems

- Estimating the size of the measure
- Estimating the total number of programmer months which have elapsed
- Estimating contractor productivity (e.g. documentation team) and incorporating this estimate in overall estimate

Lines of code

- What is a line of code?
- Productivity measures will vary from language to language – consider difference between lines of code in assembler versus Java
- Relationship to functionality must be based on past efforts in the same language

Productivity estimates

| System Category | LOC/person-month |
|---|---|
| Real-time embedded systems | 40-160 |
| Systems programs | 150-400 |
| Commercial applications | 200-800 |

Function points

- Based on a combination of program characteristics
- external inputs and outputs
- user interactions
- external interfaces
- files used by the system
- A weight is associated with each of these
- The function point count is computed by multiplying each raw count by the weight and summing all values

Function points

- Function point count modified by complexity of the project
- FPs can be used to estimate LOC depending on the average number of LOC per FP for a given language
- LOC = AVC * number of function points
- AVC is a language-dependent factor varying from 200-300 for assemble language to 2-40 for a 4GL
- FPs are very subjective. They depend on the estimator.
- Automatic function-point counting is impossible

## 4GL Object points

- Object points are an alternative function-related measure to function points when 4Gls or similar languages are used for development
- Object points are NOT the same as object classes
- The number of object points in a program is a weighted estimate of

a)The number of separate screens that are displayed

b)The number of reports that are produced by the system

c)The number of 3GL modules that must be developed to supplement the 4GL code

## Object Point Estimation

- Object points are easier to estimate from a specification than function points as they are simply concerned with screens, reports and 3GL modules
- They can therefore be estimated at an early point in the development process. At this stage, it is very difficult to estimate the number of lines of code in a system

## Quality and productivity

- All metrics based on volume/unit time are flawed because they do not take quality into account
- Productivity may generally be increased at the cost of quality
- It is not clear how productivity/quality metrics are related
- If change is constant then an approach based on counting lines of code is not as meaningful

Estimation techniques

- There is no simple way to make an accurate estimate of the effort required to develop a software system
- Initial estimates are based on inadequate information in a user requirements definition
- The software may run on unfamiliar computers or use new technology
- The skills of people working on the project may be unknown
- Project cost estimates may be self-fulfilling
- The estimate defines the budget and the product is adjusted to meet the budget
- Expert judgement
- Estimation by analogy
- Parkinson's Law
- Pricing to win
- Algorithmic cost modelling

Expert judgement

- One or more experts in both software development and the application domain use their experience to predict software costs. Process iterates until some consensus is reached.
- Advantages: Relatively cheap estimation method. Can be accurate if experts have direct experience of similar systems
- Disadvantages: Very inaccurate if there are no experts!

Estimation by analogy

- The cost of a project is computed by comparing the project to a similar project in the same application domain
- Advantages: Accurate if project data available
- Disadvantages: Impossible if no comparable project has been tackled. Needs systematically maintained cost database

Parkinson's Law

- The project costs whatever resources are available (typically used within an organization)
- Advantages: No overspend
- Disadvantages: System is usually left unfinished

Pricing to win

- The project costs whatever the organisation has to spend on it
- Advantages: You get the contract
- Disadvantages: Costs do not accurately reflect the work required. Either: (1) the management does not get the desired system or (2) the management overpays.
- This approach may seem unethical and unbusiness-like
- However, when detailed information is lacking it may be the only appropriate strategy

- The most ethical approach:
    - The project cost is agreed on the basis of an outline proposal and the development is constrained by that cost
    - A detailed specification may be negotiated or an evolutionary approach used for system development

Top-down and bottom-up estimation

- Any of these approaches may be used top-down or bottom-up
- Top-down
- Start at the system level and assess the overall system functionality and how this is delivered through sub-systems
- Bottom-up
- Start at the component level and estimate the effort required for each component. Add these efforts to reach a final estimate
- Usable without knowledge of the system architecture and the components that might be part of the system
- Takes into account costs such as integration, configuration management and documentation
- Can underestimate the cost of solving difficult low-level technical problems

Bottom-up estimation

- Usable when the architecture of the system is known and components identified
- Accurate method if the system has been designed in detail
- May underestimate costs of system level activities such as integration and documentation

Estimation methods

- Each method has strengths and weaknesses
- Estimation should be based on several methods
- If these do not return approximately the same result and the differences cannot be reconciled, there is insufficient information available
- Some action should be taken to find out more in order to make more accurate estimates

Experience-based estimates

- Estimating is primarily experience-based
- However, new methods and technologies may make estimating based on experience inaccurate
- Object-oriented rather than function-oriented development
- Client-server systems rather than mainframe systems
- Many off the shelf components
- Component-based software engineering
- Use of new CASE tools and program generators

Algorithmic cost modelling

- A formulaic approach based on historical cost information and which is generally based on the size of the software
- Cost is estimated as a mathematical function of product, project and process attributes whose values are estimated by project managers

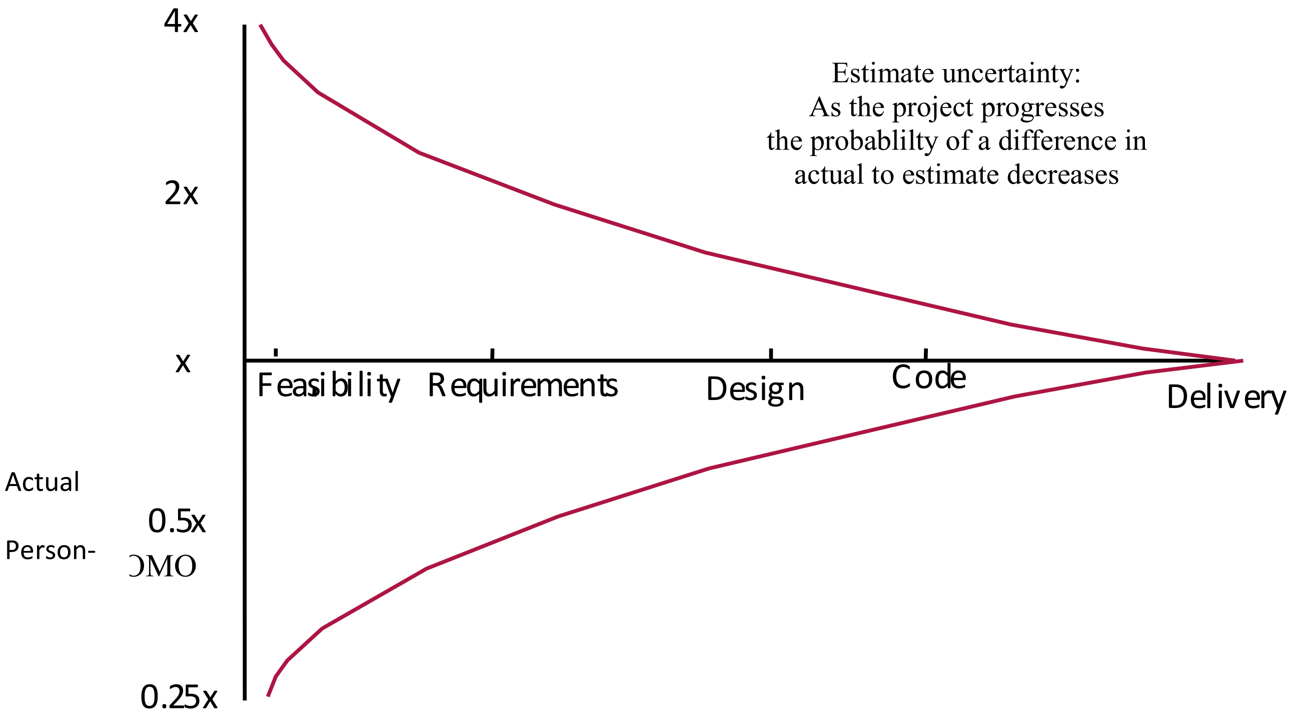
Algorithmic cost modelling

- Effort in PM = A ´ SizeB ´ M
- A is depends in on the type of software that is being developed (simple, moderate, embedded) [will vary 2.4-3.5]
- Size is an estimate of the code size or other functional assessmemt [thousands of lines of code, ie. 5,400 LOC→ 5.4]
- B reflects the disproportionate effort for large projects over small projects [typically 1.0-1.5]
- M is a multiplier reflecting a combination of product, process and people attributes (e.g. desired reliability, reuse required, personnel capability and expereince, support facilities) [will vary up from 1.0]

Estimation accuracy

- The size of a software system can only be known accurately when it is finished
- Several factors influence the final size
    - Use of “off the shelf” components
    - Programming language
    - Distribution of system
- As the development process progresses then the size estimate becomes more accurate

Estimate uncertainty



Estimate uncertainty:
As the project progresses
the probablilty of a difference in
actual to estimate decreases

x = estimated person-months

- Constructive Cost Model
- An empirical model based on project experience
- Well-documented, ‘independent’ model which is not tied to a specific software vendor
- Long history from initial version published in 1981 (COCOMO-81) through various instantiations to COCOMO 2
- COCOMO 2 takes into account different approaches to software development, reuse, etc.
- Can be used as a sanity check

Estimate Cost and Duration Very Early in Project

1. Use the function point method to estimate lines of code

2. Use Boehm's formulas to estimate labor required

3. Use the labor estimate and Boehm's formula to estimate duration

Basic COCOMO Formulae (Boehm)

Effort in Person-months = a × KLOC b

Duration in Months = c × Effort d

Where c = labour estimate, d = complexity of project type

These values are selected from a table such as the one below.

| Software Project | a | b | c | d |
|---|---|---|---|---|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semidetached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

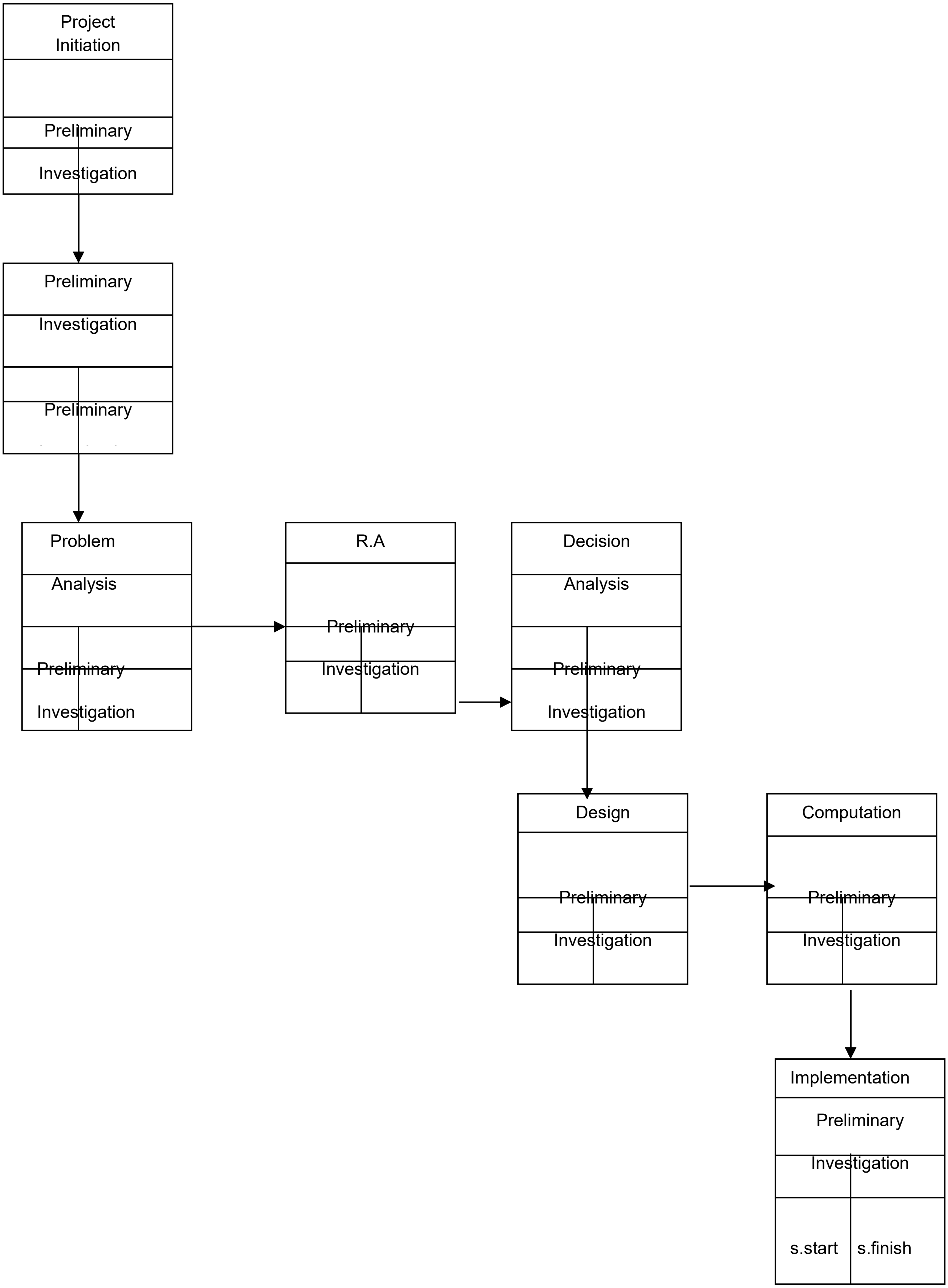
# PERT CHART

PERT CHART stand for Program Evaluation Review Techniques, unlike bar chart PERT can be both a cost and a time management system. PERT is organized by events and activity or task. One advantage of the PERT chart is that it is a schedule device that also shows graphically which task must be completed before others are begun.

Advantage of the pert chart is as follows:

- It force the manager to the plan
- It shows the interrelationship among the tasks in the project and in particularly it identified the critical path of the project, thus helping to focus on it.
- It exposes all the particularly all-possible parallelism in the activities and thus help in allocating the resources.
- It allows scheduling and simulation of alternative schedules.
- It enables the manager to monitor and control the project.

Program Evaluation Review Techniques (P.E.R.T) Chart



## Future Analysis

A new method for hand gesture recognition is introduced in this paper. The hand region is detected from the background by the background subtraction method. Then, the palm and fingers are segmented. On the basis of the segmentation, the fingers in the hand image are discovered and recognized. The recognition of hand gestures is accomplished by a simple rule classifier. The performance of our method is evaluated on a data set of 1300 hand images. The experimental results show that our approach performs well and is fit for the real-time applications. Moreover, the proposed method outperforms the state-of-art FEMD on an image collection of hand gestures.

## Conclusion

In this article, a novel motion-intend-based finger gesture recognition system based on mechanomyogram (MMG) signal is presented to recognize each finger tapping. This is the first system capable of detecting every finger tapping. The architect and the design of the system are presented. A full system characterization is also evaluated and the recognition accuracies of the system are compared under different experimental settings. Our system is able to achieve up to 94% accuracy. How different factors such as PG and BMI of the participants affect the system performances are also discussed, and the obtained results suggest that the proposed FGR system users should take more exercises to strengthen their muscles and reduce the hypodermic fat.

## References

1. Ibraheem NA, Khan R. Survey on various gesture recognition technologies and techniques. *Int J Comput Appl* (2012) 50(7):38–44. doi:10.5120/7786-0883

CrossRef Full Text | Google Scholar

2. Brunelli D, Tadesse AM, Vodermayer B, Nowak M, Castellini C. Low-cost wearable multichannel surface EMG acquisition for prosthetic hand control. *Proceedings of the 6th International Workshop on Advances in Sensors and Interfaces*. Gallipoli: IEEE (2015). p. 94–9.

Google Scholar

3. Beisteiner R, Windischberger C, Lanzenberger R, Edward V, Cunnington R, Erdler M, et al. Finger somatotopy in human motor cortex. *Neuroimage* (2001) 13(6):1016–26. doi:10.1006/nimg.2000.0737

PubMed Abstract | CrossRef Full Text | Google Scholar

4. Liao K, Xiao R, Gonzalez J, Ding L. Decoding individual finger movements from one hand using human EEG signals. *PLoS One* (2014) 9(1):e85192. doi:10.1371/journal.pone.0085192

PubMed Abstract | CrossRef Full Text | Google Scholar

5. Khan MJ, Hong MJ, Hong K-S. Decoding of four movement directions using hybrid NIRS-EEG brain-computer interface. *Front Hum Neurosci* (2014) 8:244. doi:10.3389/fnhum.2014.00244

PubMed Abstract | CrossRef Full Text | Google Scholar

6. Hotson G, McMullen DP, Fifer MS, Johannes MS, Katyal KD, Para MP, et al. Individual finger control of a modular prosthetic limb using high-density electrocorticography in a human subject. *J Neural Eng* (2016) 13(2):026017. doi:10.1088/1741-2560/13/2/026017

PubMed Abstract | CrossRef Full Text | Google Scholar

7. Nguyen H-D, Hong K-S, Shin Y-I. Bundled-optode method in functional near-infrared spectroscopy. *PLoS One* (2016) 11(10):e0165146. doi:10.1371/journal.pone.0165146

PubMed Abstract | CrossRef Full Text | Google Scholar

8. Hong K-S, Bhutta MR, Liu X, Shin Y-I. Classification of somatosensory cortex activities using fNIRS. *Behav Brain Res* (2017) 333:225–34. doi:10.1016/j.bbr.2017.06.034

PubMed Abstract | CrossRef Full Text | Google Scholar

9. Olman CA, Pickett KA, Schallmo M-P, Kimberley TJ. Selective bold responses to individual finger movement measured with FMRI at 3t. *Hum Brain Mapp* (2012) 33(7):1594–606. doi:10.1002/hbm.21310

PubMed Abstract | CrossRef Full Text | Google Scholar

10. Habermehl C, Holtze S, Steinbrink J, Koch SP, Obrig H, Mehnert J, et al. Somatosensory activation of two fingers can be