

# A.I VIRTUAL MOUSE

A Project Report

Submitted in partial fulfilment of the  
Requirements for the award of the Degree of  
**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

By

**Shalikh Anayatullah Rizvi**

**SEAT NO: 2010027**

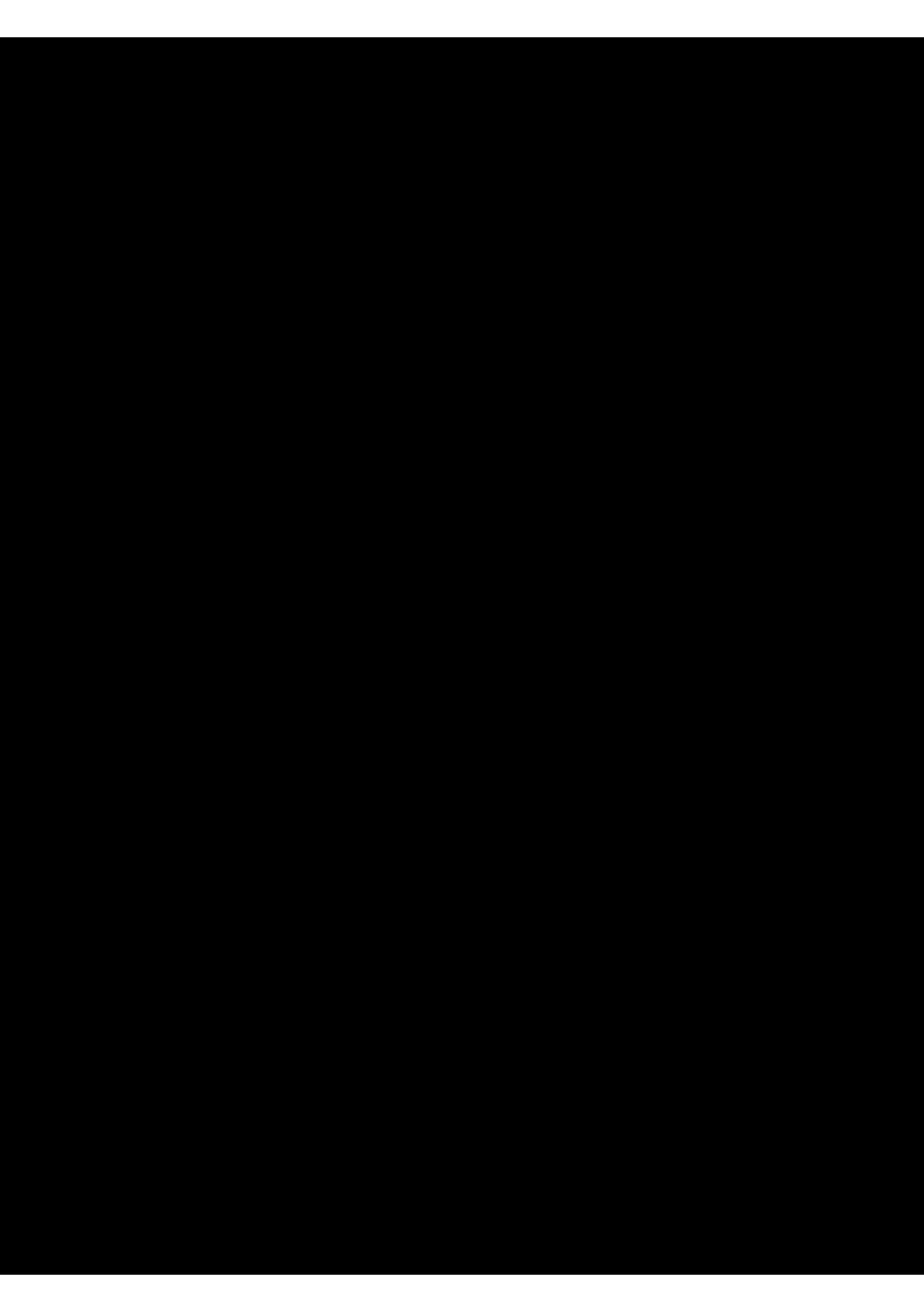
Under the esteemed guidance of  
**Mrs. Arogya Selvi**

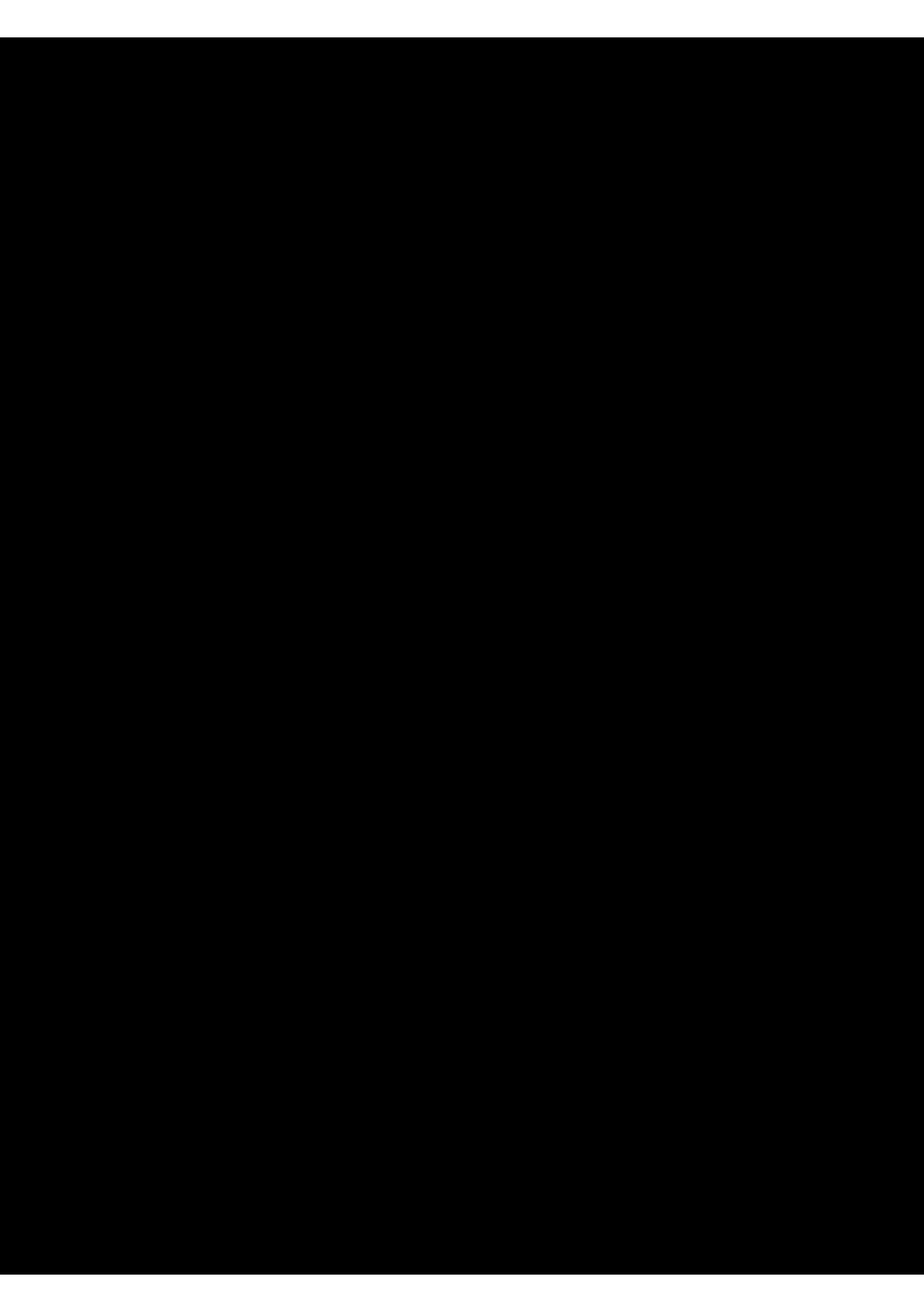


**DEPARTMENT OF INFORMATION TECHNOLOGY  
S.I.W.S  
N.P. SWAMIY COLLEGE OF COMMERCE AND ECONOMICS  
AND**

**S.M.T. THIRUMALAI COLLEGE OF SCIENCE  
(Affiliated to University of Mumbai)  
337 SEMPREE - WADALA ESTATE, MUMBAI - 400031  
MAHARASHTRA**

**2021 - 2022**





2021 - 2022  
MAHARASHTRA  
337 SEWRE - WADALA ESTATE, MUMBAI - 400031  
(Affiliated to University of Mumbai)

SMT. THIRUMALAI COLLEGE OF SCIENCE

AND

N.R. SWAMY COLLEGE OF COMMERCE AND ECONOMICS  
S.I.W.S  
DEPARTMENT OF INFORMATION TECHNOLOGY



Designation

Mrs. Arogyya Selevi  
Under the esteemed guidance of

19030

Shakikh Anayatullah Rizwan

By

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

Requirements for the award of the Degree of

Submitted in partial fulfillment of the

A Project Report

A.I. VIRTUAL MOUSE



External Examiner

College Seal

Date: \_\_\_\_\_

Co-Ordinator

Internal Guide

R. A. Ghadikar

INFORMATION TECHNOLOGY from University of Mumbai.

This is to certify that the project entitled, "A.I. VIRTUAL MOUSE" is bonafide work of  
"SHAIKH ANAYATULLAH RIZWAN" bearing Seat No: 2010027 submitted in partial  
fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in

Class: T.Y.BSc.IT Roll No: 19030 Examination Seat No: 2010027

### CERTIFICATE



DEPARTMENT OF INFORMATION TECHNOLOGY

337 SEWEE - WADALA STATE, MUMBAI MAHARASHTRA - 400031  
(Affiliated to University of Mumbai)

SMT. THIRUMALAI COLLEGE OF SCIENCE

AND

N.R. SWAMY COLLEGE OF COMMERCE AND ECONOMICS

S.I.W.S



I am developing software named „A.I. VIRTUAL MOUSE“ this software will help to use mouse through hand that is define landmark (fingertip). For this we will use camera to capture continuous movement and we will use color technique for better understanding of detection. The software will allow the user to move mouse through one finger up module and it will go in the left click mode by detecting distance between two finger module.

## ABSTRACT

## **ACKNOWLEDGEMENT**

I am greatly indebted to my project supervisor Mrs. Arogya Selvi for her guidance and constant supervision as well as providing us with all the information regarding the project titled-  
**A.I. Virtual Mouse.**

I am Also thankful to Mumbai University for providing me with all the latest technologies, thus, comforting me with project.

I take this opportunity to express my deep gratitude towards all the members of the Information Technology Department, for helping me in the completion of the project. Finally, I am thanking to my all Friends for their encouragement & support throughout the period of completion.

Again, I owe my profound gratitude to my project guide, for not only helping me with the project but also developing a keen interest in the same during its progress.

## **DECLARATION**

I hereby declare that the project entitled "A.I. VIRTUAL MOUSE" done at S.I.W.S. N.R. Swamy college of Commerce and Economics & Smt.Thirumalai College of Science, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student



Mr. Shaikh Anayatullah Rizwan

## TABLE OF CONTENT

Sr.No.	Index	Date
1	<b>INTRODUCTION</b>	
1.1	Background	
1.2	Objectives	
1.3	Purpose, Scope and Applicability	
1.3.1	Purpose	
1.3.2	Scope	
1.3.3	Applicability	
1.4	Achievements	
2	<b>SURVEY OF TECHNOLOGIES</b>	
3	<b>REQUIREMENTS AND ANALYSIS</b>	
3.1	Problem Definition	
3.2	Requirements Specification	
3.3	Planning and Scheduling	
3.4	Software and Hardware Requirements	
3.5	Preliminary Product Description	
3.6	Conceptual Models	
4	<b>SYSTEM DESIGN</b>	
4.1	Basic Modules	
4.2	Data Design	
4.3	Procedural Design	

4.4	User interface design	
4.5	Test Case Design	
<b>5</b>	<b>IMPLEMENTATION AND APPROACHES</b>	
5.1	Implementation approaches	
5.2	Coding Details and code efficiency	
5.3	Testing Approach	
<b>6</b>	<b>RESULTS AND IMPROVEMENTS</b>	
6.1	Test Reports	
<b>7</b>	<b>CONCLUSION</b>	
<b>8</b>	<b>REFERENCES</b>	

# Chapter 1:Introduction

## 1.1 Background:

It has been generations since we have been using hand gestures for communicating in human society. The shaking of hands, Thumbs up and Thumbs down signs have been ever existing in the environment. It is believed that gestures are the easiest way of interaction with anyone.

The main reason of these technologies becomes more popular is because it can be applied into many different fields easily and efficiently. Especially, gesture recognition easily can be applied to the web services, smart home systems, robot manipulation and games and in AI (Artificial intelligence) field.

## 1.2 Objectives:

The main objective is to develop this project or software is to increase its uses in robotics and upcoming technology which will increase Human-object relationship by using hand tracking module and different gesture recognition technique. The software will enable the user to use mouse of computer without using mousepad or mouse, it will detect the hand through camera of computer or laptop and will work according to hand tracking module by which user will able to move mouse and click just by showing finger in the camera.

## 1.3 Purpose ,scope and Applicability

### 1.3.1 Purpose:

The existing system consists of a mouse that can be either wireless or wired to control the cursor, know we can use hand gestures to monitoring the system. The existing virtual mouse control system consists of the simple mouse operation using the colored tips and X, Y coordinate for detection which are captured by web-cam, hence colored fingers acts as an object which the web-cam sense color like purple, green, blue color to monitor the system, whereas could perform basic mouse operation like minimize, drag, scroll up, scroll down, left-click right-click using hand gestures.

### **1.3.2 Scope:**

This software can easily replace the traditional mouse system that has been in existence for decades with the use of this algorithm the user can control the mouse without the use of any other hardware device this is done using a hand gestures recognition with inputs from a web-cam. It can also helpful for teacher and in robot industry.

### **1.3.3 Applicability:**

The project aim for improvement in mouse using system and it can be helpful for teachers who get annoyed while using wired mouse and such project can also increase interaction between computer and human and it can make user feel like real-world environment.

### **1.3.4 Achievements:**

The achievements and the knowledge that will be achieved is that it will make many activities easy for users like painting for kids and teachers teaching or explaining using hand instead mouse and can give thrilling experience to user and encounter problem of suddenly stop working of wired mouse with ease.

## CHAPTER 2: SURVEY OF TECHNOLOGY

Language	Python
Library	OpenCV, AutoPy, NumPy and MediaPipe
IDE	VsCode

### Python:

- Python is an interpreted high-level general-purpose programming language.
- Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects
- Python is dynamically-typed and garbage-collected
- It supports multiple programming paradigms including structured (particularly, procedural) object-oriented and functional programming

### OpenCV:

- OpenCV is a huge open-source library for computer vision, machine learning, and image processing.
- OpenCV supports a wide variety of programming languages like Python, C++, Java, etc.
- It can process images and videos to identify objects, faces, or even the handwriting of a human.

### **NumPy:**

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant.
- It is an open source project and you can use it freely.
- NumPy stands for Numerical Python.

### **AutoPy:**

- AutoPy is a cross-platform, simple GUI automation toolkit for Python. It includes functions for controlling the keyboard and mouse, finding colors and bitmaps on-screen, and displaying alerts — all in a cross-platform, efficient, and simple manner.

### **MediaPipe:**

- MediaPipe is Google's open-source framework, used for media processing. It is cross-platform or we can say its platform friendly. It is run on Android, iOS, web, and YouTube servers that's what Cross-platform means, to run everywhere
- Basically, MediaPipe is a framework for Computer Vision and Deep Learning that builds perception pipelines. For now, You just need to know, perception Pipelines are some sort of audio, video, or time-series data that catch the process in pipelining zone

### **VsCode:**

- Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS.
- Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.
- It is a source-code editor that can be used with a variety of programming languages such as Python, Java and c++ etc.

## CHAPTER 3: REQUIREMENT ANALYSIS

### 3.1 Problem Definition

Each technology has their pros and cons. The older one is wired technology, in which users need to tie up themselves with the help of wire in order to connect or interface with the computer system. In wired technology user cannot freely move in the room as they connected with the computer system via wire and limited with the length of wire. Later on some advanced techniques have been introduced like Image based techniques which require processing of image features like texture etc. If we work with these features of the image for hand gesture recognition the result may vary and could be different as skin tones and texture changes very rapidly from person to person from one continent to other. To overcome these challenges and promote real time application, gesture recognition technique based on color detection and their relative position with each other has been implemented. The color can also be varied and hence obviating the need of any particular color. The movement as well as mouse events of mouse are very smooth and user is able to select the small menu buttons and icons without any difficulty.

### 3.2 Requirements Specification

In order to implement this project I have used PyCharm IDE which helps in Coding and analysis, with code completion, syntax and error highlighting, linter integration, quick fixes, specialized project views, file structure views and quick jumping between files, classes, methods and usages and it also support for scientific tools like matplotlib, numpy. The framework I am going to use is Mediapipe it is created by google. They had created many amazing models that helps us to get started with fundamental AI problems such as facial detection ,hand detection ,hand landmarks detection and etc. So the model which I am going to use is handtracking it uses two main module at the backend and one is Palm detection and the other one is handlandmarks, palm detection which basically work on complete image and it provide crop image of the hand from there the hand landmarks model finds 21 different landmarks on the cropped image of the hand and I am using camera which will help in hand movement so it will work as mouse

### 3.3 Planning and scheduling:

Gantt chart:

Activities	June	July	August	September	October
Project Idea Finalization					
Survey of technology					
Requirement Detailing					
System Design					

### 3.4 Software and Hardware Requirements:

#### Hardware Requirements

- Processor: Intel core i3 and above or equivalent
- RAM: 2 GB and above
- Hard Disk: 256 GB and above
- Camera: inbuilt in laptop or external camera connected to computer

#### Software Requirements:

- Language: Python
- Operating System: Windows 8 or above
- Other software: Pycharm

### Product

#### AI Virtual Mouse

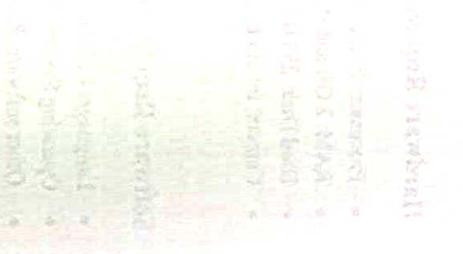
##### Project Overview

AI Virtual mouse is a modern upcoming technology which can replace wired mouse and make many activities easy for users like painting for kids and teachers teaching or explaining using hand instead of mouse and can give thrilling experience to user and with adding some more module we can implement this application in many other fields like Internet of things, so to implement this project I have used different library and frame work like OpenCV which is used to understand the content of the digital image. It extracts the description from the real-time image or digital image, which may be an object, a text description, and so on. Framework like MediaPipe is a an open-source framework from Google for building multimodal (e.g. video, audio, any time series data), cross platform (i.e Android, iOS, web, edge devices)

#### System Architecture



#### System Components



### 3.5 Preliminary Product Description

AI Virtual mouse is a modern upcoming technology which can replace wired mouse and make many activities easy for users like painting for kids and teachers teaching or explaining using hand instead of mouse and can give thrilling experience to user and with adding some more module we can implement this application in many other fields like Internet of things, so to implement this project I have used different library and frame work like OpenCV which is used to understand the content of the digital image. It extracts the description from the real-time image or digital image, which may be an object, a text description, and so on. Framework like MediaPipe is a an open-source framework from Google for building multimodal (e.g. video, audio, any time series data), cross platform (i.e Android, iOS, web, edge devices)

#### Conclusion

#### Future Work

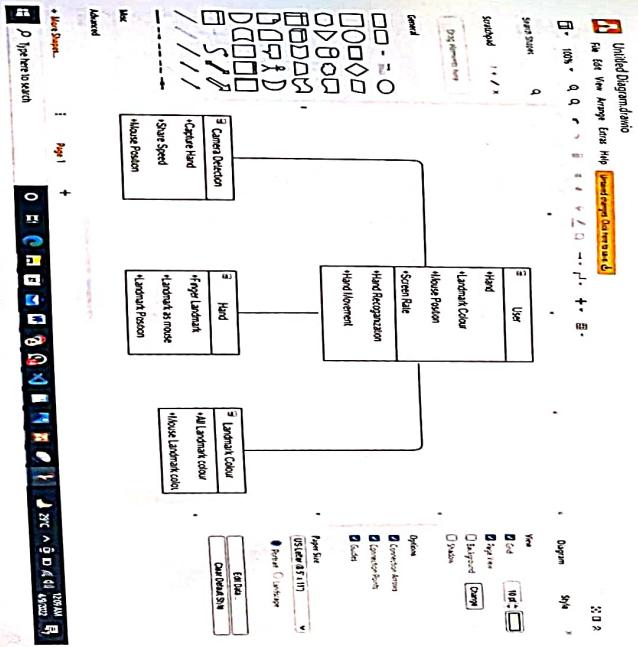
#### References

#### Appendix

## 3.6 Conceptual Models

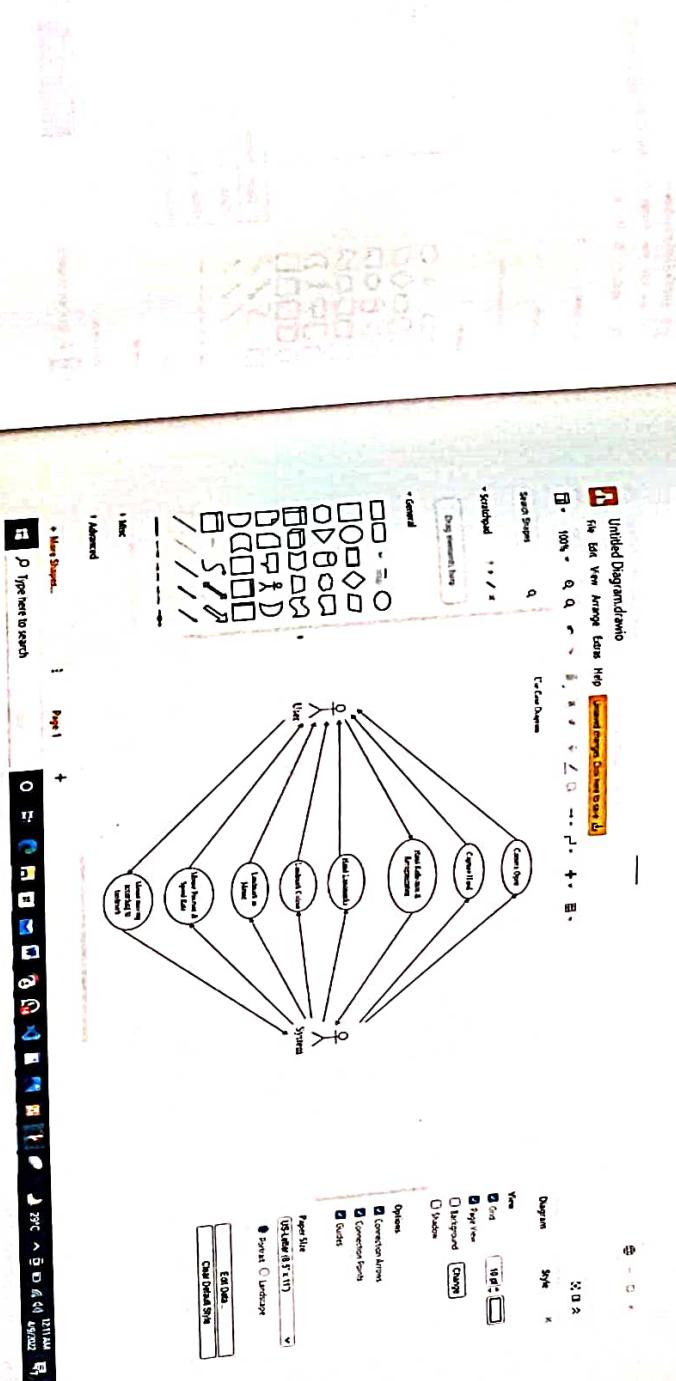
### Class Diagram:

Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among objects.



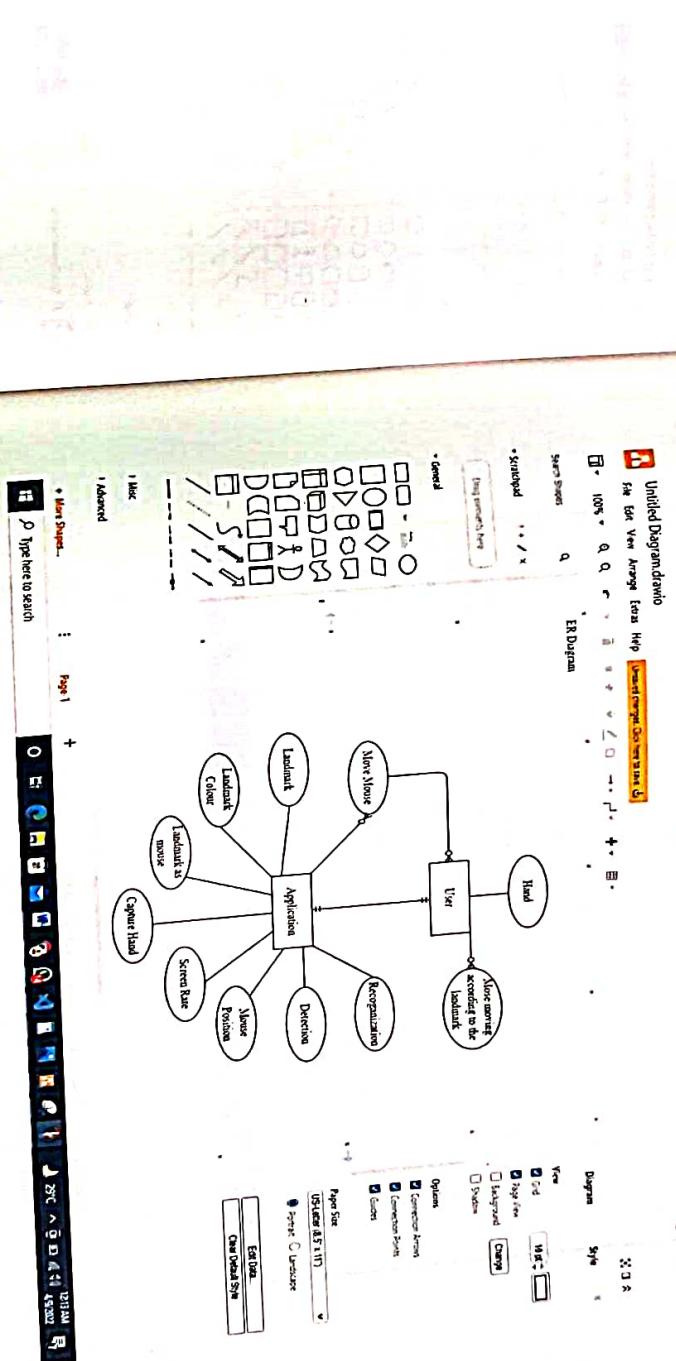
### Use case Diagram:

A Use Case Diagram is a graphical depiction of a user's possible interactions with a system. A Use Case Diagram shows various use cases and different types of users the system has.



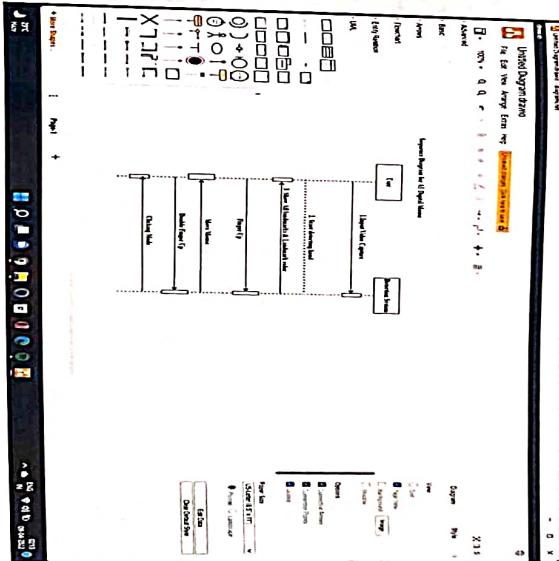
### ER Diagram:

All Entity-Relationship diagram describes interrelated things of interest in a specific domain of knowledge. A basic ER diagram is composed of entity types and specifies relationships that can exist between entities.



### Sequence Diagram:

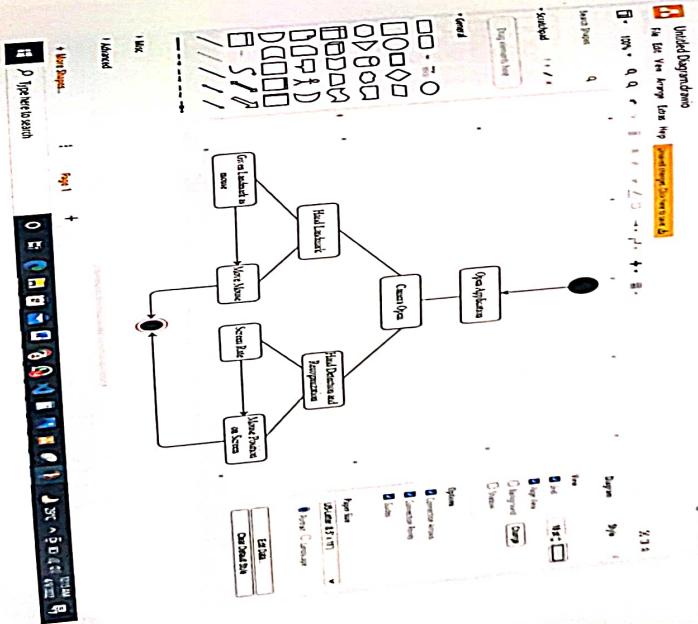
A Sequence Diagram simply depicts interaction between objects in a sequential order. It describes how-and-in what order a group of objects works together.



## State Transition:

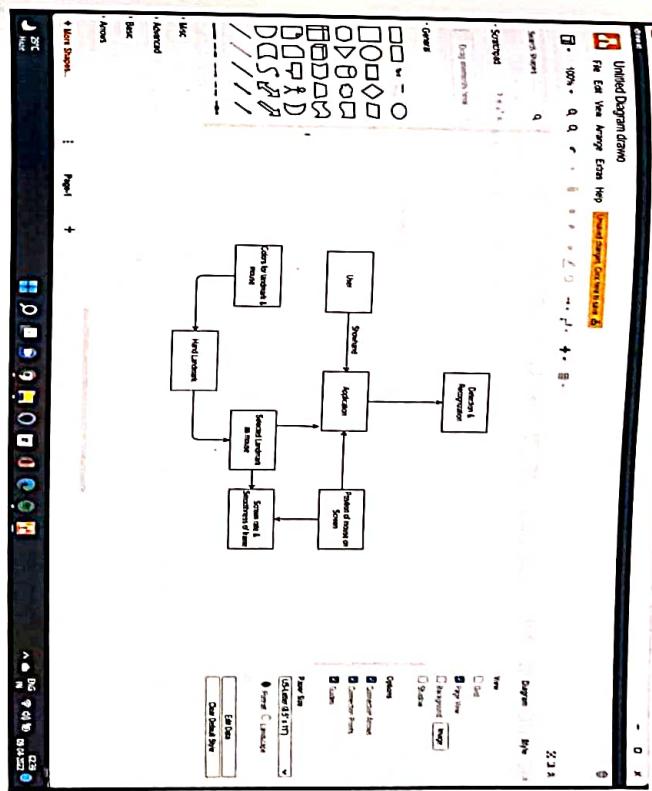
State Transition Diagrams describe all of the states that an object can have, the events under which an object changes state, the conditions that must be fulfilled before the transition occur and the activities undertaken during the life of an object

Object State Transition Diagram



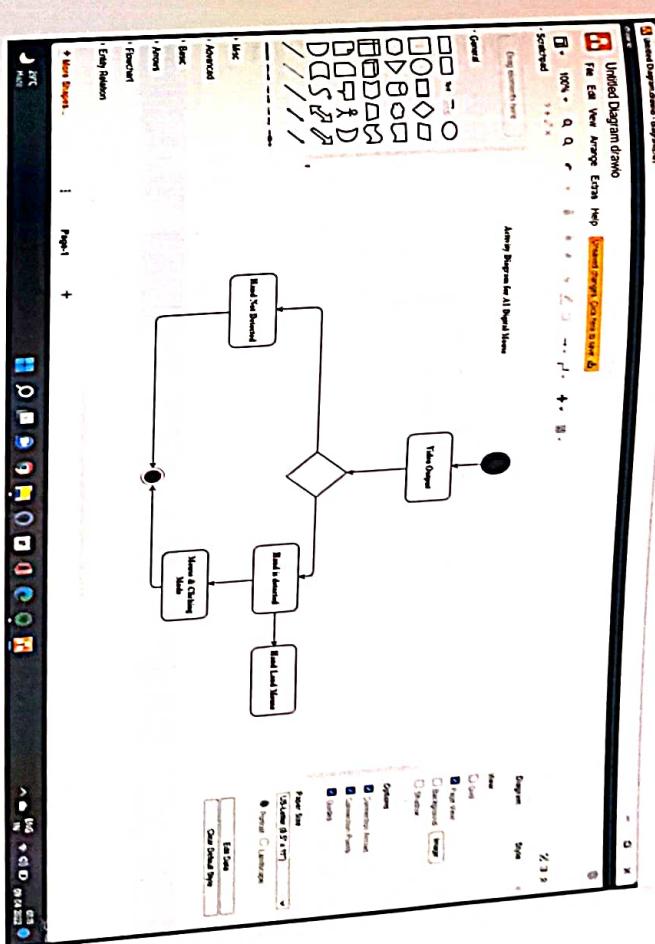
### Data Flow Diagram:

A data-flow diagram is a way of representing a flow of data through a process or a system.



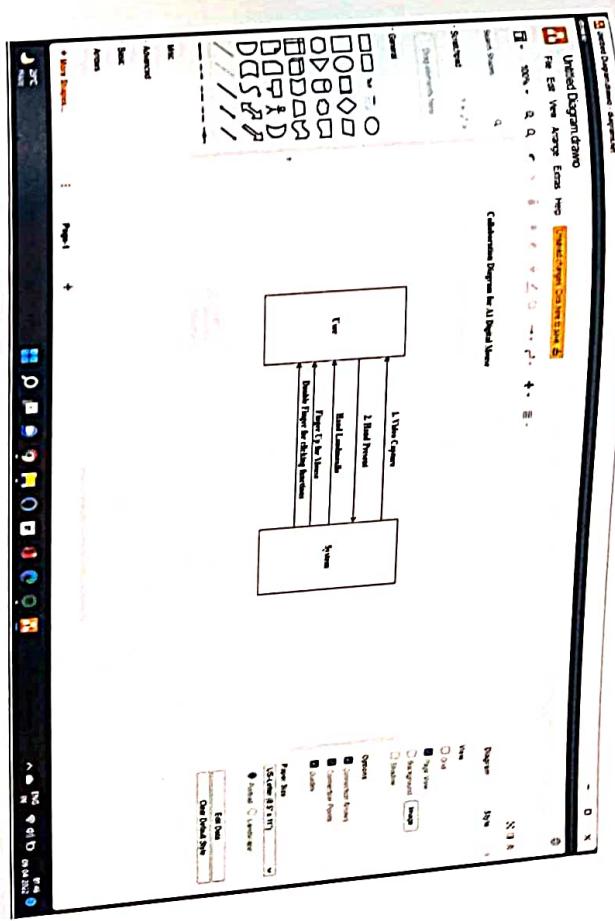
## Activity Diagram:

Activity Diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.



### Collaboration Diagram:

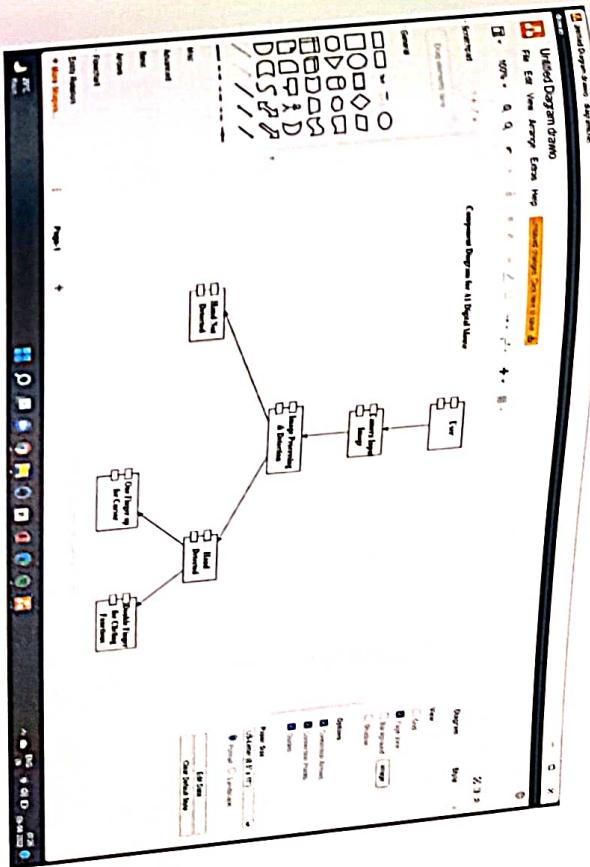
The collaboration diagram is used to show the relationship between the objects in a system. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time.



## Component Diagram

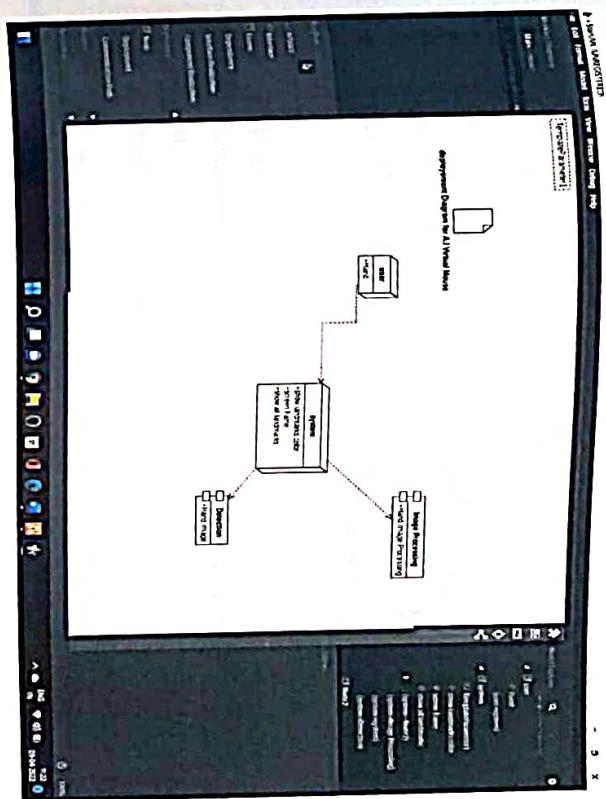
### Component Diagram:

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system.



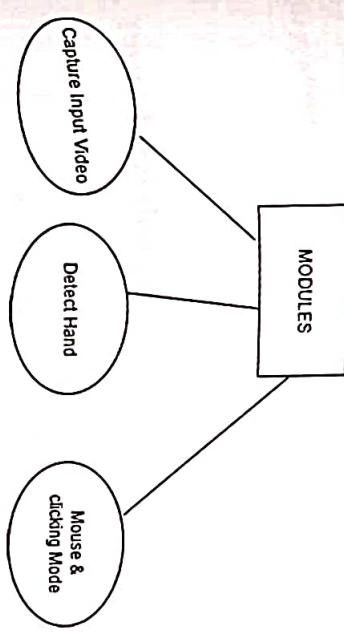
### **Deployment Diagram:**

A UML deployment diagram is a diagram that 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.



# CHAPTER 04: SYSTEM DESIGN

## 4.1 BASIC MODULE



- **MODULE:**

The Module will take input as continuous video capturing to detect any hands is present or not

- **Capture Video:**

It will capture video to detect if any hand is present or not if present it will do some time detection and some time recognition to work smoothly

- **Detect Hand:**

After detecting hand it will detect hand landmarks with the help of mediapipe library and selected landmarks for mouse will have pink circle

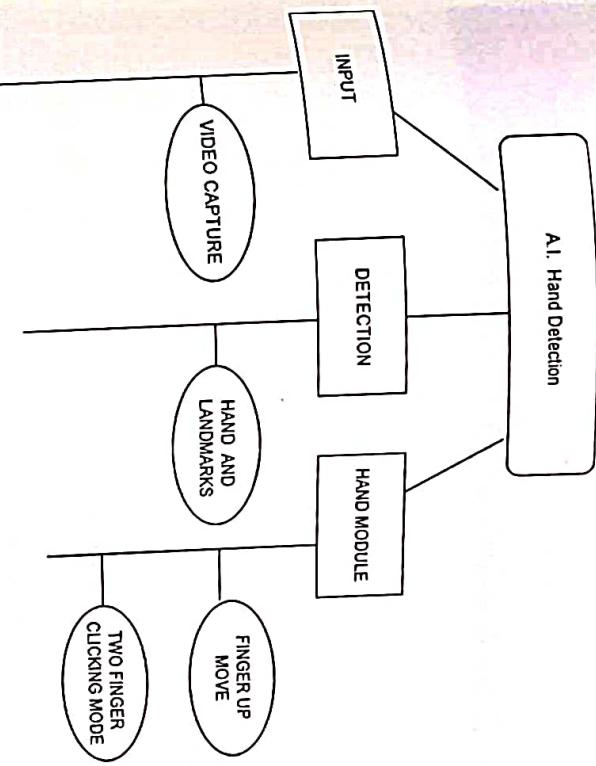
- **Mouse & Clicking Mode:**

After land mark is visible in frame raise the first finger and mouse module will get trigger and if we raise two finger clicking mode will get trigger

## 4.2 Data Design:

Data design is the first design activity, which results in less complex, modular and efficient program structure. The information domain model developed during analysis phase transformed into program structures needed for implementing the software. The data objects, attributes, and relationships depicted in entity relationship diagrams and the information stored in data dictionary provide a base for data design activity

## 4.3 Procedural design



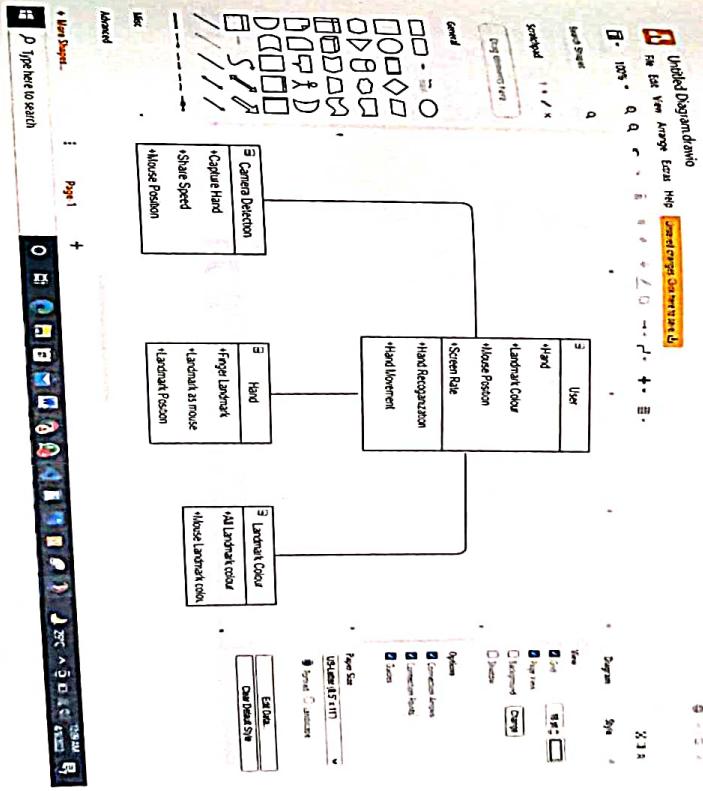
## 4.4 USER INTERFACE DESIGN

Designing User Interface



## 4.5 Test Case Design

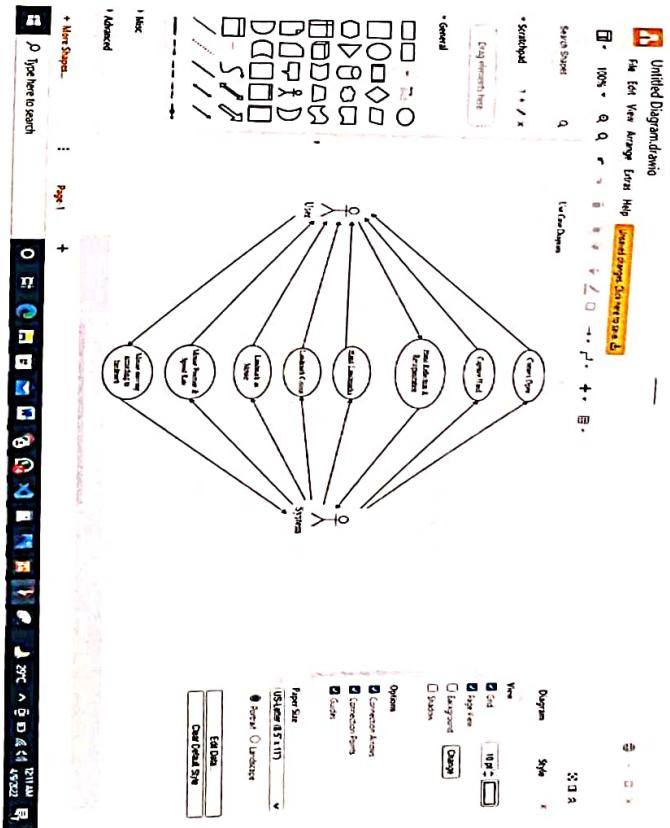
**Class Diagram:**  
Class Diagram is a static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among objects.



### Use case Diagram:

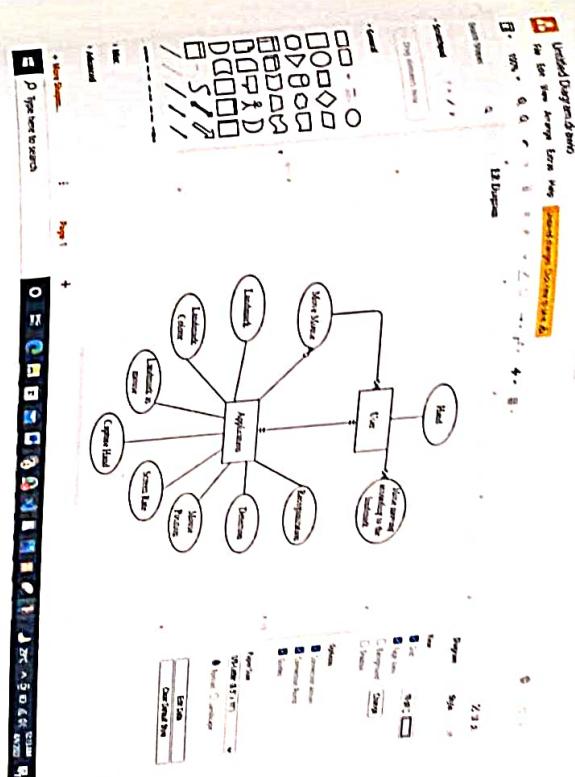
A Use Case Diagram is a graphical depiction of a user's possible interactions with a system.

A Use Case Diagram shows various use cases and different types of users the system has, and their interactions with a system.



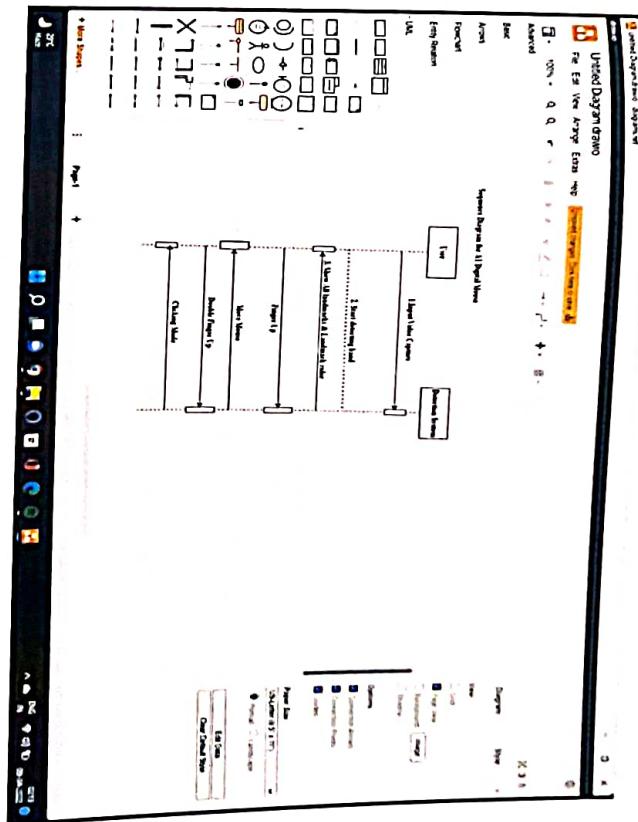
### ER Diagram:

An Entity-Relationship diagram describes interrelated things of interest in a specific domain of knowledge. A basic ER diagram is composed of entity types and specifies relationships that can exist between entities.



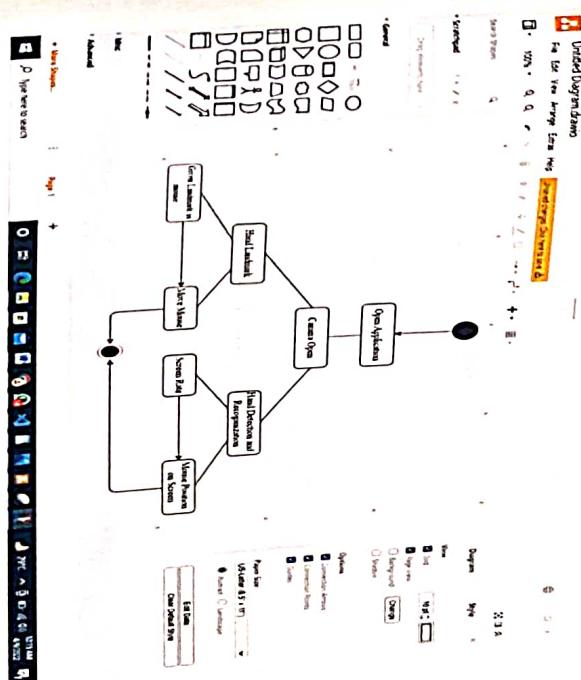
### Sequence Diagram:

A Sequence Diagram simply depicts interaction between objects in a sequential order. It describes how-and in what order a group of objects works together.



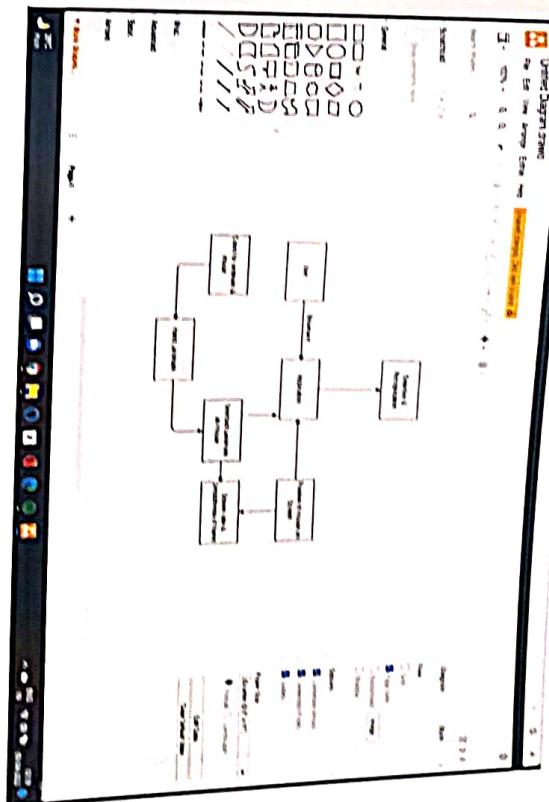
### State Transition:

State Transition Diagrams describe all of the states that an object can have, the events under which an object changes state, the conditions that must be fulfilled before the transition occur and the activities undertaken during the life of an object.



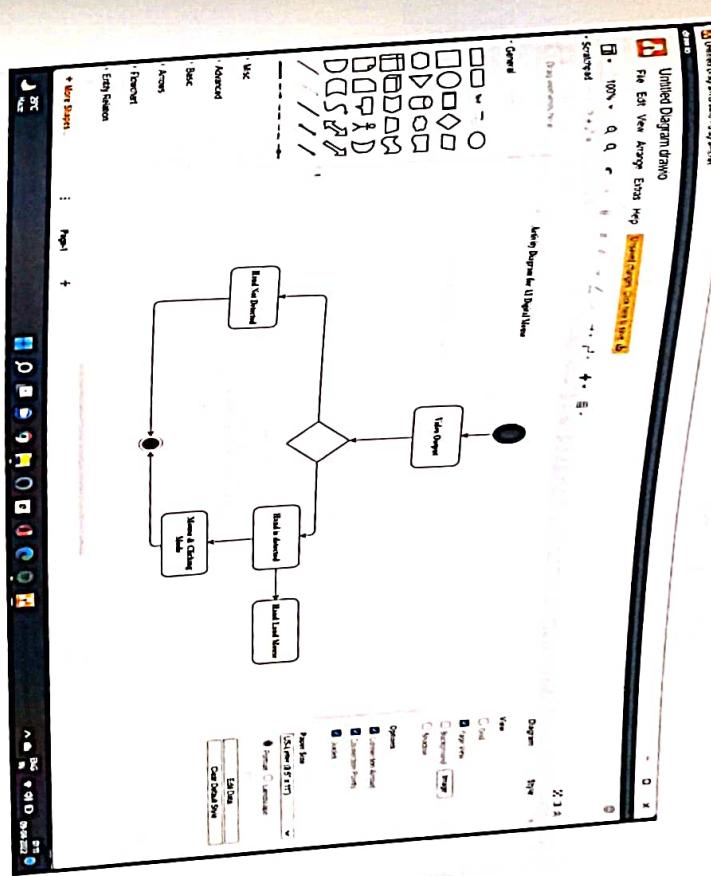
#### Data Flow Diagram:

A data flow diagram maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.



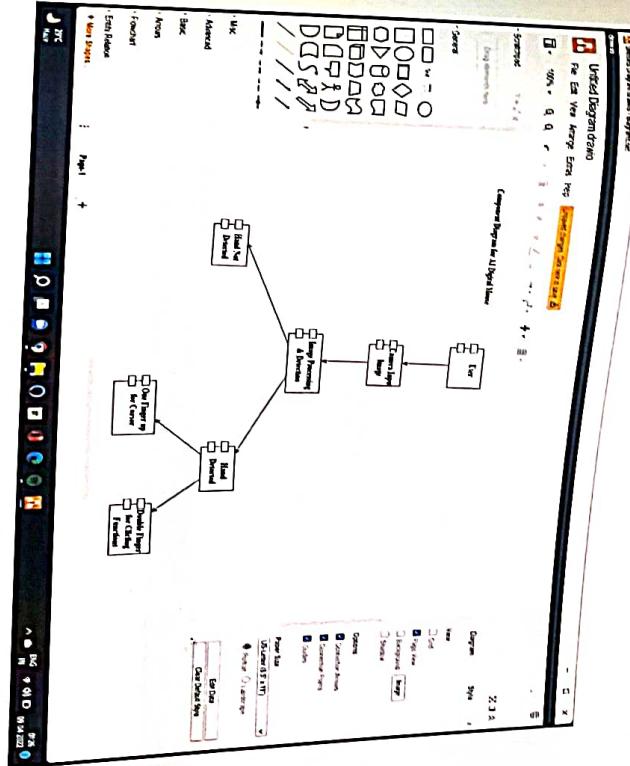
### Activity Diagram:

An activity diagram is a behavioural diagram i.e. it depicts the behaviour of a system.



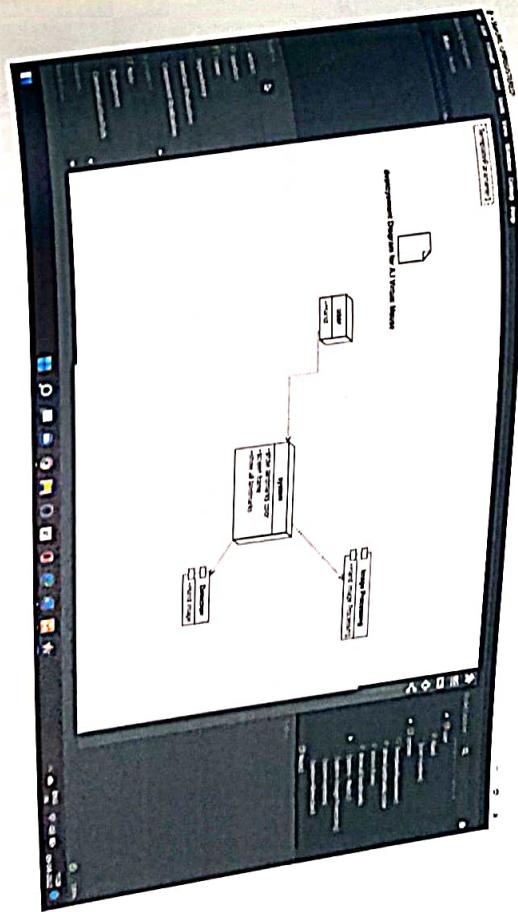
### Component Diagram:

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system.



### **Deployment Diagram:**

A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagram is a kind of structure diagram used in modelling the physical aspects of an object-oriented system.



## CHAPTER 5: IMPLEMENTATION AND TESTING

### 5.1 IMPLEMENTATION APPROCHES

#### VideoCapturing:

As application is open direct GUI window of video capturing will pop up for detection and recognition for Hand

#### HandLandMarks:

As Hand is detected it will first get recognized and then all landmarks will appear.

#### Module:

We had created two module that is when only one hand is up it will go into the cluster movement mode and if both as well as middle finger is up it will go into object tracking mode

Exit to quit off the application simply click q button on the keyboarded GUI window will get close

### 5.2 CODING DETAILS AND CODE EFFICIENCY

#### HandModuleCode:

## handM.py

```
import cv2
import mediapipe as mp
import time
import math

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.mpDraw = mp.solutions.drawing_utils
        self.tipids = [4, 8, 12, 16, 20]

    def findHands(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.mpHands.process(imgRGB)

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

    return img

    def findPosition(self, img, handNo=0, draw= True ):
        xList = []
        yList = []
        bbox = []
        self.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)
        xl1st.append(cx)
        yl1st.append(cy)
    # print(id, cx, cy)
    self.lmlist.append([id, cx, cy])
    if draw:
        cv2.circle(img, (cx, cy), 5, (255, 0, 255), cv2.FILLED)
        xmin, xmax = min(xList), max(xList)
        ymin, ymax = min(yList), max(yList)
        bbox = xmin, ymin, xmax, ymax

    if draw:
        cv2.rectangle(img, (bbox[0],bbox[1]),
                      (bbox[2]+20, bbox[3]+20), (0, 255, 0), 2)

    return self.lmlist, bbox

def findDistance(self, p1, p2, img, draw=True):
    x1, y1 = self.lmlist[p1][1], self.lmlist[p1][2]
    x2, y2 = self.lmlist[p2][1], self.lmlist[p2][2]
    cx, cy = (x1 + x2) // 2, (y1 + y2) // 2

    if draw:
        cv2.circle(img, (x1, y1), 15, (255, 0, 255), cv2.FILLED)
        cv2.circle(img, (x2, y2), 15, (255, 0, 255), cv2.FILLED)
        cv2.line(img, (x1, y1), (x2, y2), (255, 0, 255), 3)
        cv2.circle(img, (cx, cy), 15, (255, 0, 255), cv2.FILLED)

    length = math.hypot(x2 - x1, y2 - y1)
    return length, img, [x1, y1, x2, y2, cx, cy]

def fingersUp(self):
    fingers = []
    if Thumb
        if self.lmlist[self.tipIds[0]][1] > self.lmlist[self.tipIds[0] - 1][1]:
            fingers.append(1)
        else:
            fingers.append(0)

    # 4 fingers
    for id in range(1, 5):
        if self.lmlist[self.tipIds[id]][2] < self.lmlist[self.tipIds[id] - 2][2]:
            fingers.append(1)

```

१५४

```

    main():
        pipeline = []
        cline = []
        cap = cv2.VideoCapture(0)
        detector = handDetector()
        while True:
            success, img = cap.read()
            img = detector.findHands(img)
            lmList = detector.findPosition(img)
            if len(lmList) != 0:
                print(lmList[4])

```

```

cTime = tLine.time()
fps = 1 / (cTime - sTime)
sTime = cTime

```

CV-2-PATIENT (ing., str. int. (frz)), (12, 7), C-2-FONT REGENBORN 21-1-2  
(255, 0, 255), 31

```
cV2.imshow("Image", img)  
cV2.waitKey(1)
```

```
if __name__ == "__main__":  
    main()
```

```
import cv2  
import autody
```

AI Virtual Mouse code::

[AlvirtualMouse.pw](http://AlvirtualMouse.pw)

```

import time
import numpy as np
import handm as hm

# hCam = 640, 480
frameR = 50 # Frame Reduction
smoothening = 7

#-----#
piXme = 0
piYme = 0, 0
clockX, clockY = 0, 0

cap = cv2.VideoCapture(0)
cap.set(3, wCam)
cap.set(4, hCam)

detector = htm.handDetector(maxHands=1)
wScr, hScr = autopy.screen.size()

while True:
    success, img = cap.read()
    img = detector.findHands(img)
    lmList, bbox = detector.findPosition(img)

    if len(lmList) != 0:
        x1, y1 = lmList[8][1:]
        x2, y2 = lmList[12][1:]

        #print(x1, y1, x2, y2)

        fingers = detector.fingersUp()

        #print(fingers)

        cv2.rectangle(img, (frameR, frameR), (wCam - frameR, hCam - frameR),
                     (255, 0, 255), 2)
        # 4. Only index Finger : Moving Mode
        if fingers[1] == 1 and fingers[2] == 0:
            # 5. convert coordinates

            x3 = np.interp(x1, (frameR, wCam-frameR), (0, wScr))
            y3 = np.interp(y1, (frameR, hCam-frameR), (0, hScr))
            # 6. Smoothen Values
            clockX = piXme + (x3 - piXme) / smoothening
            clockY = piYme + (y3 - piYme) / smoothening

```

```
# 7. Move Mouse--  
autopy.mouse.move(x=clock, y=clock)  
cv2.circle(img, (x1, y1), 15, (255, 0, 255), cv2.FILLED)  
  
# 8. Both Index and Middle Finger are up : Clicking Node  
if fingers[1] == 1 and fingers[2] == 1:  
    length, img, lineInfo = detector.findDistance(s, 12, img)  
    if length < 39:  
        cv2.circle(img, (lineInfo[4], lineInfo[5]),  
                  15, (0, 255, 0), cv2.FILLED)  
  
# 9. Find Distance between Fingers  
cTime = time.time()  
fps = 1 / (cTime - pTime)  
pTime = cTime  
cv2.putText(img, str(int(fps)), (48, 50), cv2.FONT_HERSHEY_PLAIN, 3,  
           (255, 0, 0), 3)  
  
cv2.imshow("img", img)  
if cv2.waitKey(1) == ord('q'):  
    break  
  
cap.release()  
cv2.destroyAllWindows()
```

## **5.3 Testing approach**

### **5.3.1 Unit testing**

Unit test concentrates on the tiniest aspect of software development. This is where we measure a single unit or a group of interconnected units. In this capture video is opening properly and frame rate is also displaying.

### **5.3.2 Integrated testing**

The aim of integration testing is to take unit-tested components and put them together in a software framework that is determined by design. Integration testing entails combining a number of components to generate outputs. Here integrity testing I used to check in my software is that all landmarks was also coming on hands after detection and recognition & detection of hand all of this was unit testing. And user was able to move mouse with index finger.

## **5.4 Modifications and improvements:**

Some time screen become glitchy and not able to properly detect hand , closing window is not working properly so I have used q button as a ui closing command.

## CHAPTER 6: RESULTS AND DISCUSSION

### 6.1 TEST REPORTS:

Sr.No	Test	Expected Result	Result
1	Index finger up	Cursor should be in moving mode	Success
2	Index + Middle finger is up	Cursor should be in clicking mode	Success

### 6.2 USER DOCUMENTATION

A.I. Virtual Mouse is an application in which we can use mouse with hand without use of wired mouse all we need is a web cam for this and to quit this application press small q

## CHAPTER 7: CONCLUSION

A.I. VirtualMouse gives the best interaction between human and machine. It is also important for developing alternative human computer interaction modalities. It enables human to interface with machine in a more natural way. It can be used for many applications like sign language recognition for deaf and dumb people, robot control etc.

This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. Digital Canvas is an extension of our system which is gaining popularity among artists, by which the artist could create 2D or 3D images using the Virtual Mouse technology using the hand as brush and a Virtual Reality kit or a monitor as display set.

## CHAPTER 8: REFERENCES

1. <https://www.youtube.com>
2. <https://www.google.com>

YouTube Channel: Muratza WorkShop



