SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be **University** *u/s 3 of UGC Act, 1956)*

SOFTWARE ENGINEERING &

PROJECT MANAGEMENT

(18CSC206J)

B.TECH, II Year, IV Semester

Submitted to: Mr.Karthick Subramani

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

Course Code & Name: 18CSC206J -Software Engineering and Project Management

Team Members Tanay Shukla Saatvik Srivastava Shashank Sharma Utkarsh Srivastava	
Register Numbers RA2011026030003 RA2011026030033 RA2011026030041 RA2011026030052	
Date of Exam	

Internal Examiner-1 **Internal Examiner-2**

INDEX

Sr No.	Name of Experiment	Page No.	Date of Exp.	Date of Submission	Signature
1	Identify a software project, Create business case, and arise at a problem statement.				
2	Stake holders and user descriptions with identify appropriate process model, comparative study with agile model.				
3	Identify the general requirements, system requirements, functional and non-functional requirements.				
4	Prepare Project Plan based on scope, Find Job roles and responsibilities, calculate project effort based on resources.				
5	Prepare the work, Breakdown Structure based on timelines, Risk Identification and plan.				
6	Design a system architecture, use case diagram, Er Diagram, dataflow diagram up to level-1, class diagram, collaboration diagram.				
7	State and sequence diagram, Deployment diagram, sample frontend design (UI/UX)				
8	Coding and Result				

Lab-01

Aim: Identify a software project, Create business case, and arise at a problem statement.

Software project: AI virtual mouse **Business**

case description:

1. AI virtual mouse is a mouse simulation system that performs all the functions performedby your mouse corresponding to your hand movements and gestures.

2. A camera captures your video and depending on your hand gestures, you can move thecursor and perform a left-click, right-click, drag, select and scroll up and down.

3. The approach can make it not only easier to carry out many existing chores but also takeon trickier tasks such as creating 3-D models, browsing medical imagery during surgery without touching anything.

What is the importance of AI virtual mouse?

It's a computer mouse that's for people who are unable to move and have a disability. The device is programmed to move a cursor on the screen.

It's not able to get to specific places, but it can help people with disabilities to use a computer. It's simpler to use than a regular mouse and doesn't require the users to have the strength to move a real mouse or know how to use a keyboard.

What are the beneficiaries of the virtual mouse system?

A virtual mouse is used when someone is unable to control the real mouse. For example, a person with Parkinson's disease or extreme weakness may have uncontrollable arm and hand tremors. The person may not be able to control the real mouse. With a virtual mouse, the person with Parkinson's disease will be able to use the computer.

Problem Statement :

1. To design virtual mouse which detects hand gestures patterns instead of physical mouse.

2. dThe camera is positioned such that it recognizes the moment of finger tips and performs the operations of mouse.

3. The utilization of virtual mouse appears in space saving situations or in movement situation.

LAB-02

Aim: Stake holders and user descriptions with identify appropriate process model, comparative study with agile model.

Stakeholders and user description of our project are as follow :-

Customers/Client

The client of our software/system will be a broad group of people who wanted to enhance their user experience. They could be higher company officials, gamers, tech-innovators and research people.

Project manager

The project is lead by Utkarsh Srivastava.

Project team members

The project team members are Utkarsh Srivastava, Tanay Shukla, Saatvik Shrivastava and Shashank Sharma.

Steering committee

- Stakeholders - The stakes are shared among Utkarsh Srivastava, Shashank Sharma, Tanay Shukla and Saatvik Srivastava.

- Documentation - The documentation is headed by Shashank Sharma and Saatvik Srivastava.

- Tech team- The tech team of our project include Tanay Shukla and Utkarsh Srivastava.

Executives

The top management or the ones who planned on our project strategies are the team members of our project. Significant tech decisions of our project are taken by Tanay Shukla and Utkarsh Srivastava. Documentation and other procedural decisions are made by Shashank sharma and Saatvik Srivastava.

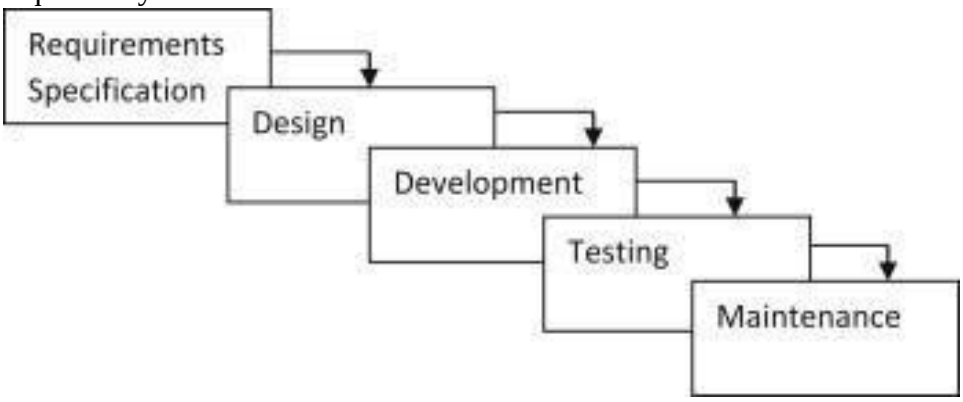
Resource managers

The resources needed to develop the project are the particular team members of our project.

Appropriate SDLC (Software Development Life Cycle) model implemented: **Waterfall Model.**

Waterfall Model

In Waterfall Model, the whole process of software development is divided into separate phases. Here typically, the outcome of one phase acts as the input for the next phase sequentially.



We adopted this model in our project development process.

1. First of all, we evaluated our requirements and planned an overview of our resources.

2. Next, we designed a prototype of our software and how it will function.

3. Thereafter, development was on it to develop based on the requirements.

4. After completing, testing was initialized and we recorded the errors.

5. We formed a proper maintenance system for the software to be maintained successfully.

We would also like to note here that the waterfall model is preferred because in this project requirements are stable and not changed frequently.
For straightforward and testable requirements, Waterfall model will yield the best results.

Comparative study against Agile model/Agility

- Agile is an incremental and iterative approach; Waterfall is a linear and sequential approach.
- Agile separates a project into sprints; Waterfall divides a project into phases.

- Agile helps complete many small projects; Waterfall helps complete one single project.

- Agile introduces a product mindset with a focus on customer satisfaction; Waterfall focuses on successful project delivery.
- Requirements are prepared everyday in Agile, while requirements are prepared once at the start in Waterfall.
- Agile allows requirement changes at any time; Waterfall avoids scope changes once the project starts.
- Testing is performed concurrently with development in Agile; testing phase comes only after the build phase in a Waterfall project.
- Test teams in Agile can take part in requirements change; test teams in Waterfall do not get involved in requirements change
- Agile enables the project team to operate without a dedicated project manager; Waterfall prefers presence of a project manager in every phase.

LAB-3

AIM: Identify the Requirements, System requirements, Functional requirements, Non-functional requirements

REQUIREMENT:

Window XP,10,11

Internet

MS office

Python 3.7

Python IDEs

Purpose

This Virtual Mouse Hand Recognition application uses a simple camera without the additional requirement of the hardware for the controlling of the cursor using simple gestures and hand control. This is done using vision based hand gesture recognition with inputs from a webcam.

Scope:

The application will simplify the process of using a computer, by reducing the need for using the mouse or typing in certain cases and since it's a desktop application it makes the process a little more acceptable

SYSTEM REQUIREMENTS

• Desktop PC-

Some minimum requirements for computers often are:

1. 1Ghz-2Ghz processor
2. Windows operating system - Which version varies but keep in mind that newer isn't always better; some companies are slow to adapt to changes.

3. 1GB of RAM
4. Webcam
5. Python and its IDE
6. 15" to 17" monitor
7. Broadband internet connection- A stable internet connection is required to run the software.
8. Web browser – Latest version of the web browser has to be used.

Functional Requirement

1. The code is written on Python3.7. It uses the cross-platform image processing module OpenCV and implements the mouse actions using Python-specific library PyAutoGUI. Video captures by the webcam are processed and only the three coloured fingertips are extracted. Their centres are calculated using the method of moments and depending upon their relative positions it is decided that what action is to be performed.

2. The first thing that we do is convert the captured video into HSV format

3. Remove Noise & Define Functions in the Video Feed

4. Final Steps (Set Position, Choose & Perform Actions)

Non-Functional Requirements

• Maintainability

• Utility

• Availability

• Reliability

The objective of this study is to develop a system:

The purpose of the implemented solution is to control the mouse cursor by user hand gestures captured through a webcam. For improving the gesture recognition based on the fluctuation of illuminance levels.

LAB SESSION-4

Aim:- To prepare project plan based on scope, Identify job roles and responsibilities and calculate project effort based on resources.

To prepare project plan based on scope

TASK	START DATE	END DATE	DURATION
GATHER REQUIREMENTS	29/03/22	01/04/22	3
UI DESIGN	02/04/22	09/04/22	2
CODING	10/04/22	30/04/22	5
UNIT TESTS	01/05/22	07/05/22	3
SYSTEM TESTS	08/05/22	12/05/22	3
BUG FIXES	13/05/22	16/05/22	2
IMPROVEMENTS	17/05/22	20/05/22	2
FINAL TESTING	21/05/22	22/05/22	3

RELEASE VERSION 23/05/22

GANTT CHART OF THE PROJECT:-

FIND JOB ROLES AND RESPONSIBILITIES:-

NAME	ROLES
Tanay shukla and Utkarsh Srivastava	Team Leader and head coder
Shashank sharma	UI/ UX Designer/documentation
Saatvik srivastava	Coder and Tester

Calculate project effort based on resources

Task	Duration (in hrs.) Extras(in hrs.)
Gather requirements	**12 3**
UI Design	**16 0**
Coding	**105 6**
Unit Tests	**21 0**

System Test	**15 0**
Bug Fixes	**8 2**
Improvement	**8 0**
Final Testing	**6 0**
Release Version	**4 0**
Total Time	**195 11**

Assumption ->working hours->

1. Total time to complete the project= 195 hrs
2. Total time after adding extra time to complete the project= 195+11 = 206 hrs
3. Total time after adding project management time to complete the project=206+15%of 206=236.9
4. Total time taken after adding contingency hours=236.9+45%of 236.9=236.9+106.605=343.505

5. Total time taken after review and adjustments=343.505-

Result:- This lab has been successfully completed.

LAB SESSION -5

Aim:- Prepare the Work Breakdown Structure based on timeliness, Risk Identification and Plan.

- **Work Breakdown Structure:**

- It includes dividing a large and complex project into simpler, manageable and independent tasks.
- For constructing a work breakdown structure, each node is recursively decomposed into smaller sub-activities, until at the leaf level, the activities becomes undividable and independent. ● It follows a Top-Down approach.

Steps:

•Step-1: Identify the major activities of the project.
•Step-2: Identify the sub-activities of the major activities.
•Step-3: Repeat till undividable, simple and independent activities are created.

Uses:

•It allows to do a precise cost estimation of each activity.
•It allows to estimate the time that each activity will take more precisely.
•It allows easy management of the project.
•It helps in proper organization of the project by the top management.

- **Risk Management:**

Risk management is the process of identifying, assessing, and prioritizing the risks to minimize, monitor, and control the probability of unfortunate events.

•Schedule Risk :

Schedule related risks refers to time related risks or project delivery related planning risks. The wrong schedule affects the project development and delivery.
•Some reasons for Schedule risks –
•Time is not estimated perfectly
•Improper resource allocation
•Tracking of resources like system, skill, staff etc •Frequent project scope expansion
•Failure in function identification and its' completion

• **Budget Risk :**
Budget related risks refers to the monetary risks mainly it occurs due to budget overruns. Always the financial aspect for the project should be managed as per decided but if financial aspect of project mismanaged then there budget concerns will arise by giving rise to budget risks. So proper finance distribution and management are required for the success of project otherwise it may lead to project failure. Some reasons for Budget risks –
• Wrong/Improper budget estimation
• Unexpected Project Scope expansion
• Mismanagement in budget handling• Cost overruns Improper tracking of Budget

• **Operational Risks :**

Operational risk refers to the procedural risks means these are the risks which happen in day-to-day operational activities during project development due to improper process implementation or some external operational risks. Some reasons for Operational risks –
• Insufficient resources
• Conflict between tasks and employees
• Improper management of tasks
• No proper planning about project
• Less number of skilled people
• Lack of communication and cooperation
• Lack of clarity in roles and responsibilities

- Insufficient training

Technical Risks :

Technical risks refers to the functional risk or performance risk which means this technical risk mainly associated with functionality of product or performance part of the software product. Some reasons for Technical risks –

1. Frequent changes in requirement
2. Less use of future technologies
3. Less number of skilled employee
4. High complexity in implementation
5. Improper integration of modules

•Programmatic Risks :

Programmatic risks refers to the external risk or other unavoidable risks. These are the external risks which are unavoidable in nature. These risks come from outside and it is out of control of programs. Some reasons for Programmatic risks –

•Rapid development of market

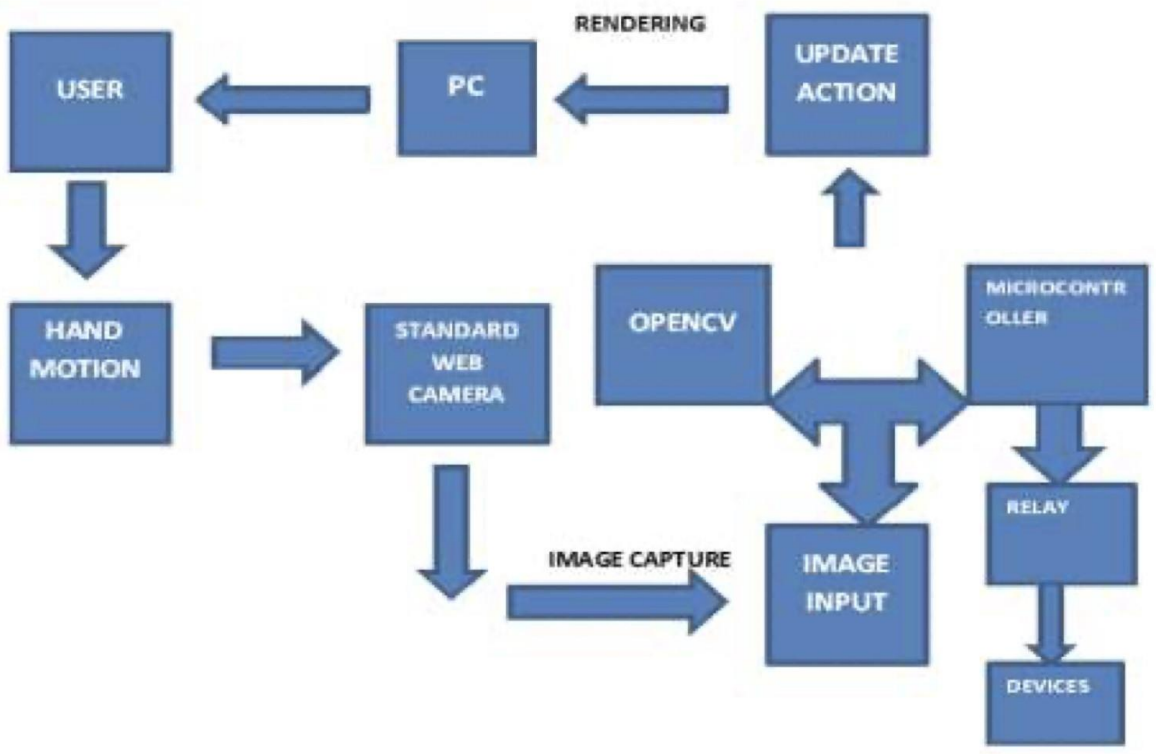
•Running out of fund / Limited fund for project development •Changes in Government rules/policy

- Loss of contracts due to any reason

LAB SESSION 6: Design a System Architecture, Use Case Diagram, ER Diagram, DFD Diagram, Class Diagram and Collaboration Diagram.

- **Design System Architecture:**

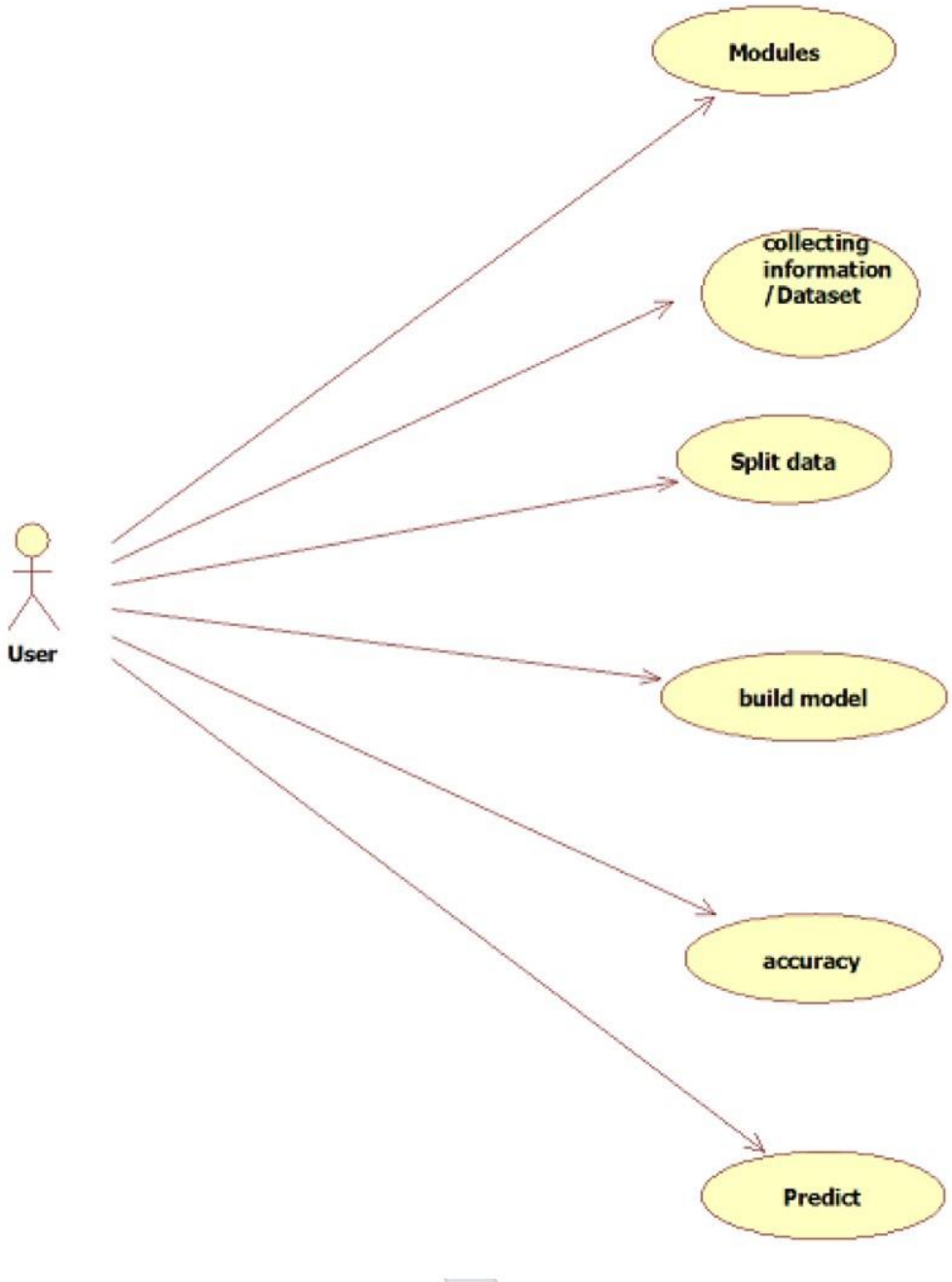
A system architecture or systems architecture is the conceptual model that defines structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure of the system which comprises system components, the externally visible properties of those components and provides a plan from which products can be procured and systems developed, that will work together to implement the overall system.



- **Use Case Diagram:**

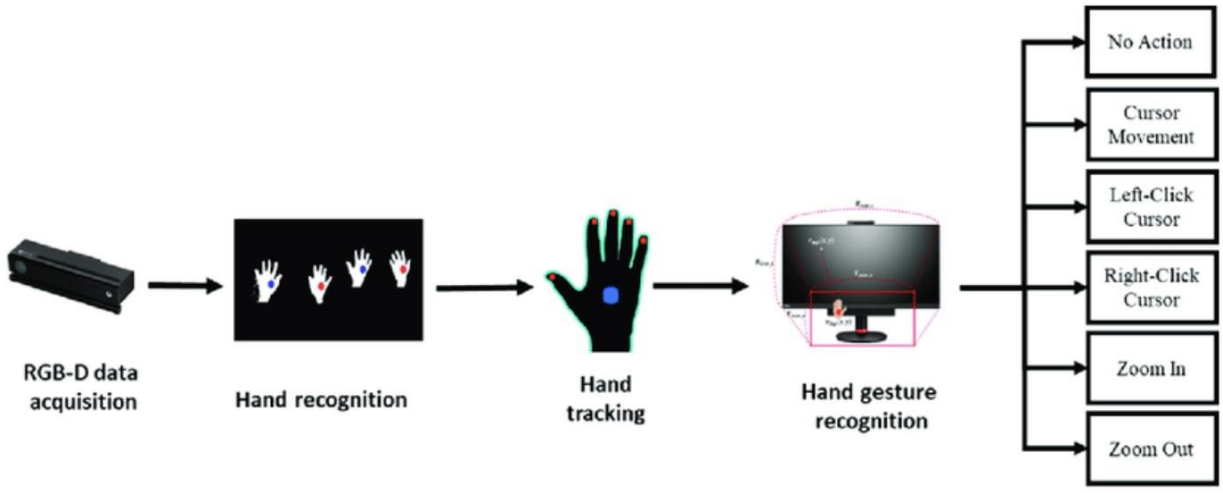
Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in

collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. The User looks into the camera. The camera captures frames of the User and according to User head movements determines the motion of the pointer to the left, right, up and down.



- **Data Flow Diagram:**

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing.



- **Class Diagram:**

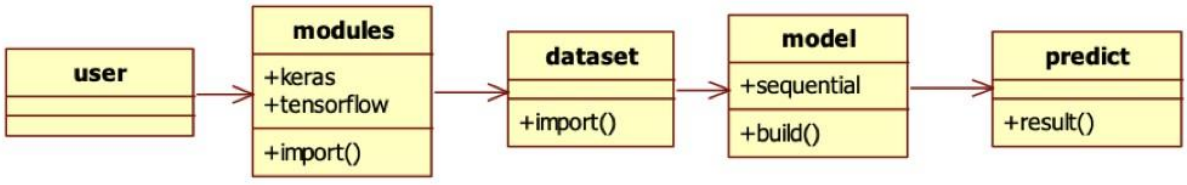
A class diagram in Unified Modeling Language(UML) is a type of static structure diagram that describe the structure of a system by showing the system‟s classes, their attributes, operations(or) methods and relationship between the classes. The class diagram is the main building block in object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both main objects and or interaction in the application and the objects to be programmed.

We define 3 classes as User, the Camera that acts as a medium to capture frames and Mode that determines the actions to be done according to the result of the algorithm for comparing frames and selecting the action.

User

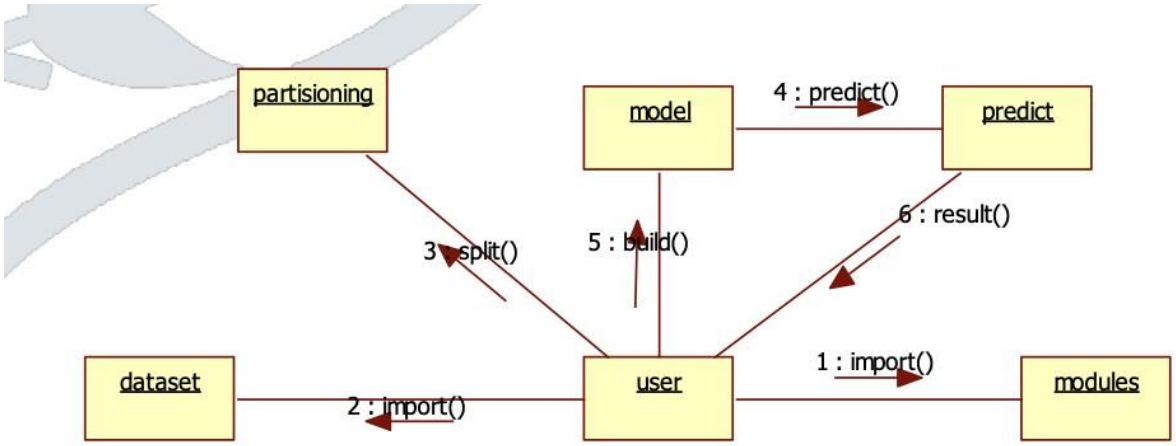
The User moves the head and chooses the direction for the mouse pointer to be moved on the screen. The User also closes the eyes to perform clicking actions.
Camera
Camera grabs pictures and according to algorithms detects the head and eyes. It draws a rectangle on the eyes and head to show detection.

Movement
There are two movements mapped to the head and eyes. Head is used for moving of the pointer on the screen whereas eyes are used to select clicking options like left click, right click, scroll up and scroll down.



- **Collaboration Diagram:**

A collaborating diagram also called a communication diagram or interaction diagram is an illustration of the relationships and interaction among software objects in the Unified Modeling Language (UML).
User looks into the camera. The camera grabs frames and displays image to the user. User selects the movements to be done by the pointer which are accordingly performed by the system and displayed to the User. User also selects Mouse actions to be performed which are then displayed back to the User.


partisioning
model
4 : predict()
predict
6 : result()
3 : split()
5 : build()
dataset
user
1 : import()
modules
2 : import()

LAB SESSION -7

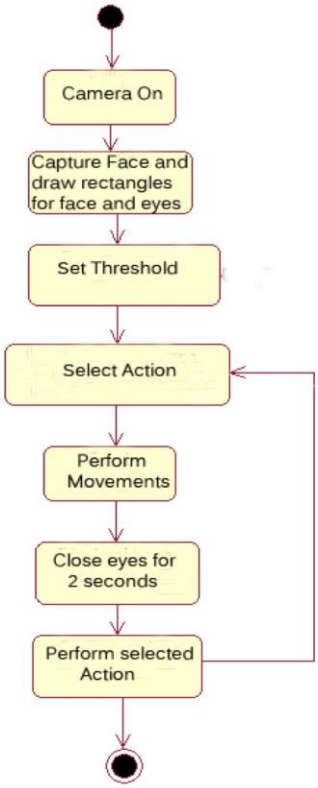
Aim:- State and Sequence Diagram, Deployment Diagram, Sample Frontend Design (UI/UX)

- **State Chart Diagram:**

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented in series of events that could occur in one or more possible states.

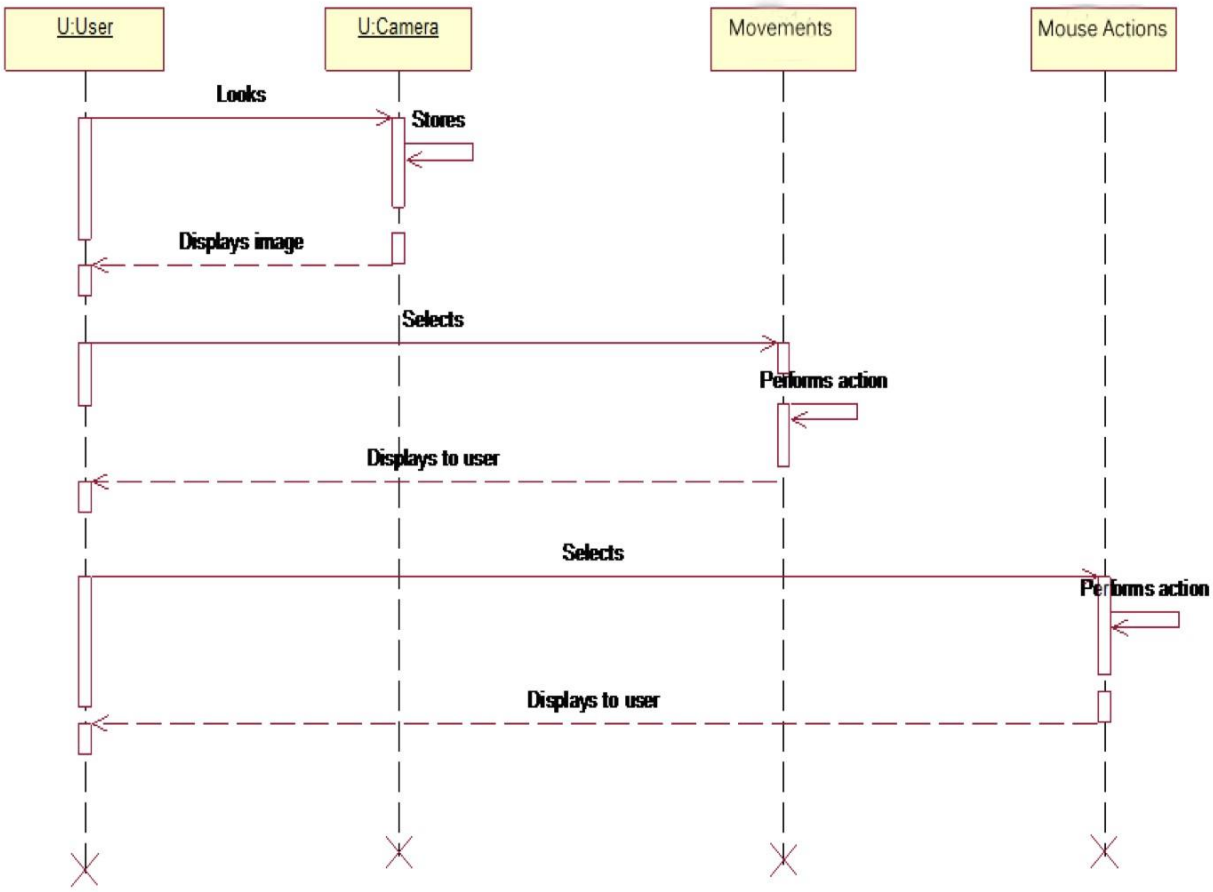
Following are the main purposes of using State chart diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object



- **Sequence Diagram:**

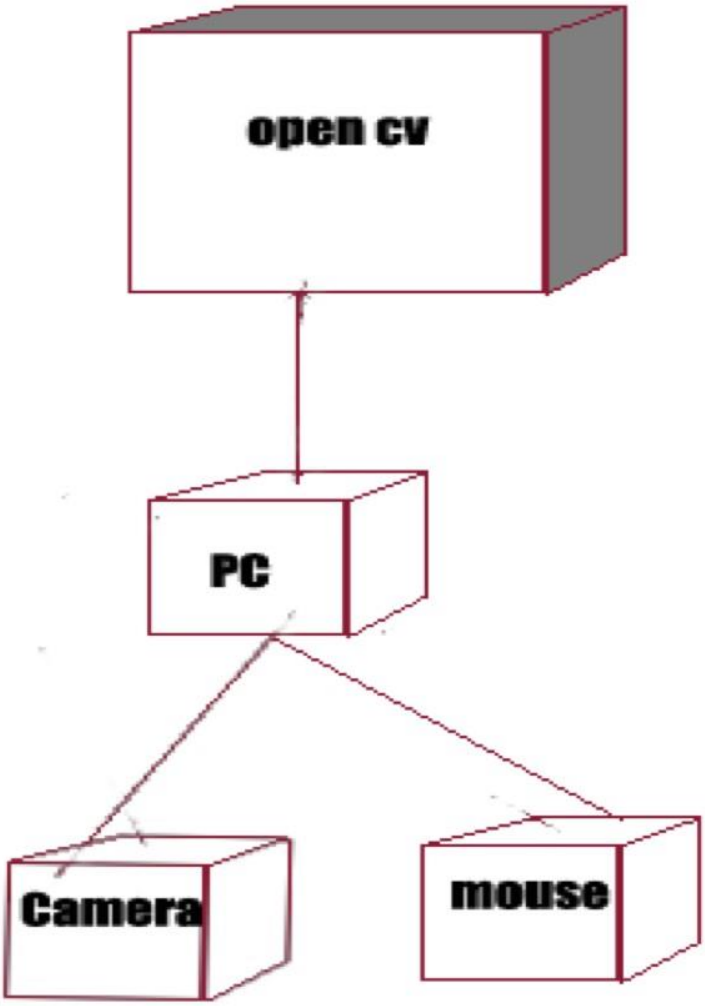
A sequence diagram in a Unified Modeling Language(UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.
The User looks into the Camera. The camera stores the frames and displays the image to the User. The User selects the movements to be performed. If movements are detected, the pointer is accordingly moved on the screen and displayed to the user. If Mouse action is selected by the User, the needed selection mode is shown and displayed back to the User. When the application is closed, all the objects are stopped.



- **Deployment Diagram:**

A deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for your system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another.

Deployment diagrams can also be created to explore the architecture of embedded systems, showing how the hardware and software components work together.



LAB SESSION -8

Aim:- Building codes according to our project and user requirement.

Capture Video from Camera

Program:-

```
# All packages needed for the program are imported ahead


import cv2 cap =
cv2.VideoCapture(0)
while(1):
```

```
    # Capture frame-by-frame
    _, frameinv = cap.read()

    # flip horizontaly to get mirror image in camera
    frame = cv2.flip( frameinv, 1)

    # Our operations on the frame come here hsv =
    cv2.cvtColor( frame, cv2.COLOR_BGR2HSV)

    # Display the resulting frame
    cv2.imshow('Frame', hsv)

    k = cv2.waitKey(10) & 0xFF
    if k == 27:
       break cap.release()
cv2.destroyAllWindows()
```

Calibrate the colour ranges

```
import cv2 import
numpy as np

def nothing(x):
    pass

# Create a black image, a window
kernel = np.zeros((300,512,3), np.uint8)
name = 'Calibrate'
cv2.namedWindow(name)

# create trackbars for color change
cv2.createTrackbar('Hue', name, 0, 255, nothing)
```

```
cv2.createTrackbar('Sat', name, 0, 255, nothing)
cv2.createTrackbar('Val', name, 0, 255, nothing)

# create switch for ON/OFF functionality
switch = '0 : OFF \n 1 : ON'

cv2.createTrackbar(switch, name,0,1,nothing)

while(1):
    cv2.imshow(name,kernel) k
    = cv2.waitKey(1) & 0xFF
    if k == 27:
        break

    # get current positions of four trackbars
    hue = cv2.getTrackbarPos('Hue', name) sat
    = cv2.getTrackbarPos('Sat', name) val =
    cv2.getTrackbarPos('Val', name) s =
    cv2.getTrackbarPos(switch,name)

    if s == 0:
        kernel[:] = 0 else:
    kernel[:] = [hue,sat,val]

cv2.destroyAllWindows()
```

Remove Noise & Define Functions in the Video Feed

```
# cv2.inRange function is used to filter out a particular color from the frame
# The result then undergoes morphosis i.e. erosion and dilation
# Resultant frame is returned as mask
```

```
def makeMask(hsv_frame, color_Range):

        mask = cv2.inRange( hsv_frame, color_Range[0], color_Range[1])
        # Morphosis next ...
        eroded = cv2.erode( mask, kernel, iterations=1)

        dilated = cv2.dilate( eroded, kernel, iterations=1)

        return dilated
```

Find Contours & Draw Centroids

```
# Contours on the mask are detected.. Only those lying in the previously set area #
range are filtered out and the centroid of the largest of these is drawn and returned
def drawCentroid(vid, color_area, mask, showCentroid):

        contour, _ = cv2.findContours( mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
        l=len(contour)

        area = np.zeros(l)

    # filtering contours on the basis of area rane specified globally
        for i in range(l):
                if cv2.contourArea(contour[i])>color_area[0] and cv2.contourArea(contour[i]):
            area[i] = cv2.contourArea(contour[i])
          else:
            area[i] = 0 a =
    sorted(area, reverse = True)

    # bringing contours with largest valid area to the top
        for i in range(l):
                for j in range(1):
                        if area[i] == a[j]:
```

```
                    swap( contour, i, j)
    if l > 0 :
        # finding centroid using method of
        'moments' M = cv2.moments(contour[0]) if
        M['m00'] != 0:
            cx = int(M['m10']/M['m00']) cy =
            int(M['m01']/M['m00']) center = (cx,cy) if
            showCentroid: cv2.circle( vid, center, 5,
            (0,0,255), -1) return center
    else:
        # return error handling values
        return (-1,-1)
```

Final Steps (Set Position, Choose & Perform Actions)

```
'''
This function takes as input the center of yellow region (yc) and the
previous cursor position (pyp). The new cursor position is calculated in
such a way that the mean deviation for desired steady state is reduced.
'''
def setCursorPos( yc, pyp):

    yp = np.zeros(2)

    if abs(yc[0]-pyp[0])<5 and abs(yc[1]-pyp[1])<5:
        yp[0] = yc[0] + .7*(pyp[0]-yc[0])
    yp[1] = yc[1] + .7*(pyp[1]-yc[1]) else:
        yp[0] = yc[0] + .1*(pyp[0]-yc[0])
        yp[1] = yc[1] + .1*(pyp[1]-yc[1])

    return yp
```

PyAutoGUI library:

- free cursor movement
- left-click
- right-click
- drag/select
- scroll up
- scroll down

```
# Depending upon the relative positions of the three centroids, this function chooses whether
# the user desires free movement of cursor, left click, right click or dragging
def chooseAction(yp, rc, bc):
        out = np.array(['move', 'false'])
        if rc[0]!=-1 and bc[0]!=-1:

                if distance(yp,rc)<50 and distance(yp,bc)<50 and distance(rc,bc)<50 :
                        out[0] = 'drag'
                out[1] = 'true' return
                out elif
                distance(rc,bc)<40:
                        out[0] = 'right'
                return out elif
                distance(yp,rc)<40:
                        out[0] = 'left' return out elif
                distance(yp,rc)>40 and rc[1]-bc[1]>120:
                        out[0] =
                'down' return out elif
                bc[1]-rc[1]>110:
                        out[0] = 'up'
                return out else:
                        return out
```

```
        else:
            out[0] = -1
            return out
def performAction( yp, rc, bc, action, drag, perform):
        if perform:
            cursor[0] = 4*(yp[0]-110)
            cursor[1] = 4*(yp[1]-120)
            if action == 'move':
                if yp[0]>110 and yp[0]<590 and yp[1]>120 and yp[1]<390:
                    pyautogui.moveTo(cursor[0],cursor[1]
                ) elif yp[0]<110 and yp[1]>120 and
                yp[1]<390: pyautogui.moveTo( 8 , cursor[1])
                elif yp[0]>590 and yp[1]>120 and yp[1]<390:
                pyautogui.moveTo(1912, cursor[1]) elif
                yp[0]>110 and yp[0]<590 and yp[1]<120:
                pyautogui.moveTo(cursor[0] , 8) elif
                yp[0]>110 and yp[0]<590 and yp[1]>390:
                    pyautogui.moveTo(cursor[0] ,
                1072) elif yp[0]<110 and yp[1]<120:
                pyautogui.moveTo(8, 8) elif yp[0]<110 and
                yp[1]>390: pyautogui.moveTo(8, 1072) elif
                yp[0]>590 and yp[1]>390:
                    pyautogui.moveTo(1912, 1072)
                else: pyautogui.moveTo(1912, 8)
            elif action == 'left':
            pyautogui.click(button = 'left') elif action
            == 'right':
                pyautogui.click(button = 'right')
                time.sleep(0.3)
        elif action == 'up':
```

```
                    pyautogui.scroll(5)
#                   time.sleep(0.3)
                elif action == 'down':
                    pyautogui.scroll(-5)
#                   time.sleep(0.3)
                elif action == 'drag' and drag == 'true':
                    global y_pos
                    drag = 'false'
                    pyautogui.mouseDown()

                    while(1):
                        k = cv2.waitKey(10) & 0xFF
                        changeStatus(k)
                        _, frameinv = cap.read()
                        # flip horizontaly to get mirror image in camera frame =
                        cv2.flip( frameinv, 1) hsv = cv2.cvtColor( frame,
                        cv2.COLOR_BGR2HSV) b_mask = makeMask( hsv,
                        blue_range) r_mask = makeMask( hsv, red_range) y_mask =
                        makeMask( hsv, yellow_range) py_pos = y_pos b_cen =
                        drawCentroid( frame, b_area, b_mask, showCentroid) r_cen =
                        drawCentroid( frame, r_area, r_mask, showCentroid) y_cen =
                        drawCentroid( frame, y_area, y_mask, showCentroid)

                        if      py_pos[0]!=-1 and y_cen[0]!=-1:
                            y_pos = setCursorPos(y_cen, py_pos)
                        performAction(y_pos, r_cen, b_cen, 'move', drag, perform)
                        cv2.imshow('Frame', frame) if distance(y_pos,r_cen)>60 or
                        distance(y_pos,b_cen)>60 or
distance(r_cen,b_cen)>60:
                            break
```

```
pyautogui.mouseUp()
```