**REMOTE DESKTOP**

**SASHANK S 2016103060**

**RAHUL S 2016103568**

**CHAPTER NO. TITLE PAGE No**

**1. PROBLEM STATEMENT 3**

**2. INTRODUCTION 3**

**3. PROBLEM DESCRIPTION 3**

**4. WORKFLOW 4**

**5. IMPLEMENTATION 5**

**5.1 Preparing the Rotating Parts 5**

**5.1.2 Preparing Microcontroller Board 5**

**5.1.3 Preparing the Interface Module 6 5.1.4 Preparing the Proximity Sensor Module 6**

**5.1.4.1 Working Principle: 7**

**5.1.5 Preparing Servo Motor 8**

**6. CONCLUSION 9**

**7. CODE 9**

**1 PROBLEM STATEMENT:**

To establish a remote desktop connection between a client and a server.

**2.INTRODUCTION:**

Electronic devices have acquired an increasingly important role in our society and are integrated in our lives making both the users and their devices more accessible. Currently in the western world most families have at least one computer. This computer is generally equipped with multimedia accessories and an Internet connection. Portable devices, such as mobile phones and PDAs, are part of this technological and social environment. a. However, there is a problem when it comes to the interconnectivity between all these devices. Today users face many difficulties when attempting to use what should be the aggregated possibilities of their devices, rather than simply the functionality of each device. The hypothesis of this project is that the user's difficulties could be overcome if their devices could be internetworked.

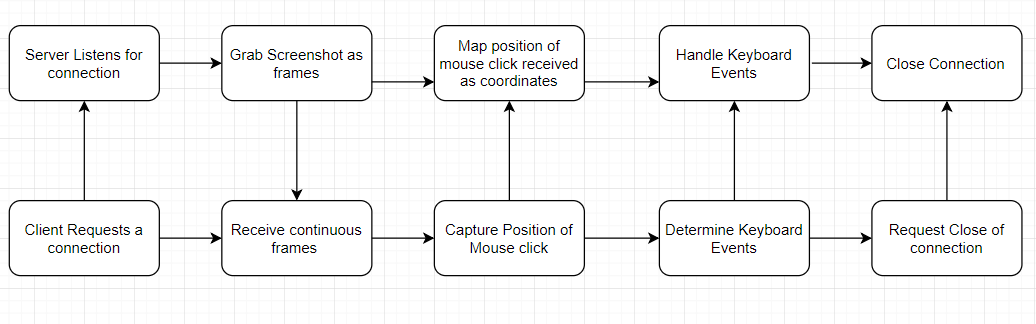
**3.PROBLEM DESCRIPTION:**

We can also use wireless links to connect multiple devices together. The advantages of this are that we do not need to use cables to connect the different components, but the disadvantages are that we now have to address the problems of (1) Device discovery and (2) Addressing.An additional disadvantage of wireless device aggregation is that the cost of developing a wireless version of a device can be expensive. One of the possibilities for device aggregation is to internetwork devices via the use of remote desktop applications. These applications give a use the possibility to connect to and use their devices remotely through the Internet or some other kind of network. Remote desktop applications provide remote control of applications that normally utilize the system’s graphical user interface (typically referred to as a desktop), virtually adding new capabilities and services to a device that is executing the remote desktop client application. Remote desktop applications protocols work on the application layer, generally over TCP/IP, and they are responsible for communication between the remote desktop server and the remote desktop client.

****

**3.1 TCP/IP stack on a typical remote desktop scenario**

**4.WORKFLOW:**



**4.1 GENERAL OUTLINE OF WORKFLOW**

**5 IMPLEMENTATION:**

**5.1.1 Establishing HTTP connection**

We use Flask framework to establish a connection between client and server.

1. First we imported the [**Flask**](http://flask.pocoo.org/docs/1.0/api/#flask.Flask) class. An instance of this class will be our WSGI application.
2. Next we create an instance of this class. The first argument is the name of the application’s module or package.
3. We then use the [**route()**](http://flask.pocoo.org/docs/1.0/api/#flask.Flask.route) decorator to tell Flask what URL should trigger our function.
4. The function is given a name which is also used to generate URLs for that particular function, and returns the message we want to display in the user’s browser.
5. By using URL of the server we can run the client on any platform.

**5.1.2 Capturing Frames**

1. We use Pillow library to capture the screen. The **ImageGrab** module can be used to copy the contents of the screen or the clipboard to a PIL image memory.
2. Using **pyautogui** we get the position of the mouse pointer. Using **ImageDraw** we then draw the position of the mouse pointer on the image
3. Convert the captured image to a **numpy** array.
4. Convert the color scheme from BGR to RGB using **OpenCV. Cv2.cvtColor** changes color space of the bytes in the array and stores it as an image frame.
5. Then using OpenCV encode the frame to jpeg image. **Cv2.imencode** does this encoding.
6. Return the jpeg image as a byte array.
7. This frame is sent to the html page as an image which is loaded in the img tag.

**5.1.3 Capturing Mouse events on client side**

We handle the following events

* MouseMove – mouse pointer movement
* Tap - leftclick
* Taphold ContextMenu - rightclick
* Doubletap DoubleClick – double click

These are handled on client side using **JavaScript’s DOM**. We get the position of the mouse using the properties pageX and pageY. These properties return the coordinates of the mouse pointer, relative to the document, when the mouse event was triggered. So we calculate the position with respect to the screen size. This will give the actual position of the mouse. Then we send the type of event, mouse coordinates and screen resolution to the server.

At the server side we map the position of mouse pointer by calculating a fixed ratio and multiplying it with the server side screen resolution. Using pyautogui we move to mouse pointer to the calculated position. We use the following **pyautogui** functions

* moveTo(x,y)
* click(x,y)
* click(x,y,button=right)
* doubleclick(x,y)

**5.1.4 Handling Keyboard Events**

We handle keyboard events on the clients side using **Javascript’s DOM.**

The **Keyup** attribute along with the **keycode** is used to determine which key has been pressed.

We handle Function keys, character keys(except alphabets) and alphabet keys using three different functions.

**5.1.4.1 Function Keys:**

1. The keys which are identified using key codes. Eg. Enter 13, backspace 8, left 37, right 38 etc.
2. On the server side these can be executed using **pyautogui.press(key)**.
   * + 1. **Char Keys:**
3. We try to determine if shift is pressed or not.
4. This can be determined by using the property of key event – **shiftKey.**
5. We send the key and if shift key is pressed or not.
6. On the client side if shift is pressed we use the following combinations.

* **Pyautogui.keydown(‘shift’)**
* **Pyautogui.keydown(character)**
* **Pyautogui.keyup(character)**
* **Pyautogui.keyup(‘shift’)**

1. If shift is not pressed we use **pyautogui.typewrite(character)**

**5.1.4.3 Alphabet Keys**

Here we check if the alphabet to be typed is in caps or not in client side.

We do this by checking the ascii value using **event.which** attribute as well as if the shift key is pressed or not using **event.shiftKey.** We then send the alphabet in either upper or lower case to the server.

At the server we use **pyautogui.typewrite(character)** to key in the text.

**5.2 Tools:**

Since we use different python modules in this project.

|  |  |  |
| --- | --- | --- |
| S.No | TITLE | DESCRIPTION |
| 1 | **Flask** | Flask is a web application framework written in Python. Web Application Framework or simply Web Framework represents a collection of libraries and modules that enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management etc. |
| 2 | **Pillow** | The **Python Imaging Library** adds image processing capabilities to your Python interpreter.  This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.  The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool. |
| 3 | **pyautogui** | Cross-platform GUI automation for human beings.  PyAutoGUI is a Python module for programmatically controlling the mouse and keyboard. |
| 4 | **OpenCV** | OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. An end-to-end solution for all your image and video needs. |
| 5 | **numpy** | NumPy is the fundamental package for scientific computing with Python. It contains among other things:   * a powerful N-dimensional array object * sophisticated (broadcasting) functions * tools for integrating C/C++ and Fortran code * useful linear algebra, Fourier transform, and random number capabilities |
| 6 | **JavaScript DOM** | The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:   * The HTML elements as **objects** * The **properties** of all HTML elements * The **methods** to access all HTML elements * The **events** for all HTML elements |

5.1. TABULATION OF TOOLS

**6. CONCLUSION :**

Our Remote Desktop Application is designed in such a way that it is cross platform. We can run it in any platform that we want to. We want to extend this idea by hosting it on a server and enabling access through not only a local network. We would also like to add multi user and multi tasking capabilities.

**7. CODE:**

**Server.py**

|  |
| --- |
|  |

from flask import Flask, Response, render\_template, request  
from PIL import ImageGrab, ImageDraw  
import cv2  
import sys  
import numpy as np  
import pyautogui  
  
  
app = Flask(\_\_name\_\_)  
  
  
def grabImg():  
 while True:  
 allframes = frames()  
 for frame in allframes:  
 yield (b'--frame\r\n' + b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')  
  
  
def frames():  
 while True:  
 img = ImageGrab.grab()  
 img = draw\_mouse(img)  
 # convert image to numpy array  
 img\_np = np.array(img)  
 # convert color space from BGR to RGB  
 frame = cv2.cvtColor(img\_np, cv2.COLOR\_BGR2RGB)  
 # convert image to jpg format  
 ret, jpeg = cv2.imencode('.jpg', frame)  
 yield jpeg.tobytes()  
  
  
def draw\_mouse(img):  
 *"""  
 utility function to draw mouse pointer  
 """* # generate Draw object for PIL image  
 draw = ImageDraw.Draw(img)  
 # find current position of mouse pointer  
 pos = pyautogui.position()  
 # coordinates of ellipse  
 ax, ay, bx, by = pos[0], pos[1], pos[0] + 10, pos[1] + 10  
 # draw ellipse on image  
 draw.ellipse((ax, ay, bx, by), fill="yellow")  
 return img  
  
  
@app.route('/sendImg')  
def sendImg():  
 return Response(grabImg(), mimetype='multipart/x-mixed-replace; boundary=frame')  
  
  
@app.route('/mouse', methods=['GET', 'POST'])  
def mouse():  
 type = request.form['type']  
 ex = request.form['x']  
 ey = request.form['y']  
 imX = request.form['X']  
 imY = request.form['Y']  
 dx, dy = pyautogui.size()  
 print(ex, ey, imX, imY, dx, dy, file=sys.stderr)  
 print(type)  
 x, y = dx \* (float(ex) / float(imX)), dy \* (float(ey) / float(imY))  
 if type == "mousemove":  
 pyautogui.moveTo(x, y)  
 elif type == "click":  
 pyautogui.click(x, y)  
 elif type == 'rightclick':  
 pyautogui.click(x, y, button='right')  
 elif type == 'dblclick':  
 pyautogui.doubleClick(x, y)  
  
 return ""  
  
  
@app.route('/keyboard', methods=['POST'])  
def keyboard\_event():  
 # keyoard event  
 event = request.form.get('type')  
 print(event)  
 pyautogui.press(event)  
 return ""  
  
  
@app.route('/text', methods=['POST'])  
def text\_event():  
 chars = request.form['type']  
  
 pyautogui.typewrite(chars)  
 return ""  
  
  
@app.route('/char', methods=['POST'])  
def char\_event():  
 shift = request.form['shift']  
 char = request.form['type']  
 if (shift == 'true'):  
 pyautogui.keyDown('shift')  
  
 pyautogui.keyDown(char)  
 pyautogui.keyUp('shift')  
 pyautogui.keyUp(char)  
 else:  
 pyautogui.typewrite(char)  
 # pyautogui.typewrite(chars)  
 return ""  
  
  
@app.route('/')  
def screen():  
 return render\_template('screen.html')  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(host='192.168.1.3', port=8003)  
 # app.run(debug=True)

**Client.html**

<!DOCTYPE html>  
<html lang="en">  
<style>  
  
 img {  
 max-width: 100%;  
 max-height: 100vh;  
 height: auto;  
}  
  
</style>  
<head>  
  
 <meta charset="UTF-8">  
 <title>Screen</title>  
  
 <script type="text/javascript" src="{{ url\_for('static', filename='js/jquery-3.3.1.min.js') }}"></script>  
 <script type="text/javascript" src="{{ url\_for('static', filename='js/jquery.finger.min.js') }}"></script>  
 <script type="text/javascript" src="{{ url\_for('static', filename='js/materialize.min.js') }}"></script>  
</head>  
<body>  
<div >  
 <img class="screen" id="screen" src="{{ url\_for('sendImg') }}">  
 <script>  
 function get\_xy(event, offset){  
 // function to get position of event on image  
 if(event.pageX == null){  
 // for mobile  
 var x = event.x - offset.left  
 var y = event.y - offset.top;  
 }  
 else{  
 // for pc  
 var x = event.pageX - offset.left  
 var y = event.pageY - offset.top;  
  
 }  
 return [x,y];  
 }  
 function mouse\_event(screen, event, type1) {  
 var offset = screen.offset();  
 var point = get\_xy(event, offset);  
 console.log(type1);  
  
 $.ajax({  
 type: 'POST',  
 url: "http://192.168.1.3:8003/mouse",  
 data: {  
 type: type1,  
 x: point[0],  
 y: point[1],  
 X: screen.width(),  
 Y: screen.height()  
 },  
 dataType:"json",  
 success: function(result) {  
 //alert("success")  
 }  
 });  
 }  
  
function keyboard\_event(type1) {  
 console.log(type1);  
 $.ajax({  
 type: 'POST',  
 url: "http://192.168.1.3:8003/keyboard",  
 data: {  
 type: type1  
 },  
 success: function(result) { }  
 });  
 }  
  
function text\_event(ch,ev){  
console.log(ev);  
var which = -1;  
 if (ev.which) {  
 which = ev.which;  
 } else if (ev.keyCode) {  
 which = ev.keyCode;  
 }  
 var shift\_status = false;  
 if (ev.shiftKey) {  
 shift\_status = ev.shiftKey;  
 } else if (ev.modifiers) {  
 shift\_status = !!(ev.modifiers & 4);  
 }  
 if (((which >= 65 && which <= 90) && !shift\_status) ||  
 ((which >= 97 && which <= 122) && shift\_status)) {  
 // uppercase, no shift key  
 $.ajax({  
 type: 'POST',  
 url: "http://192.168.1.3:8003/text",  
 data: {  
 type: ch  
 },  
 success: function(result) { }  
 });  
} else {  
 $.ajax({  
 type: 'POST',  
 url: "http://192.168.1.3:8003/text",  
 data: {  
 type: ch.toUpperCase()  
 },  
 success: function(result) { }  
 });  
}  
  
 }  
  
 function char\_event(ch,ev){  
console.log(ev);  
if(ev.shiftKey){  
 // uppercase, no shift key  
 $.ajax({  
 type: 'POST',  
 url: "http://192.168.1.3:8003/char",  
 data: {  
 shift : 'true',  
 type: ch  
 },  
 success: function(result) { }  
 });  
} else {  
 $.ajax({  
 type: 'POST',  
 url: "http://192.168.1.3:8003/char",  
 data: {  
 shift : 'false',  
 type :ch  
 },  
 success: function(result) { }  
 });  
}  
  
 }  
  
  
  
 $(document).ready(function() {  
 $.Finger.doubleTapInterval = 2000;  
 document.oncontextmenu = function() {return false;};  
  
  
 $('#screen').on('mousemove', function(event) {  
 var screen = $(this);  
 mouse\_event(screen, event, "mousemove");  
 });  
 $('#screen').on('tap', function(event) {  
 var screen = $(this);  
 mouse\_event(screen, event, "click");  
  
 });  
 $('#screen').on('taphold contextmenu', function(event) {  
 var screen = $(this);  
 mouse\_event(screen, event, "rightclick");  
 });  
 $('#screen').on('doubletap dblclick', function(event) {  
 var screen = $(this);  
 mouse\_event(screen, event, "dblclick");  
 });  
 $('html').on('keyup', function(event){  
 if(event.keyCode == 13){  
 keyboard\_event('enter')  
 }else if(event.keyCode == 8){  
 keyboard\_event('backspace')  
 }else if(event.keyCode == 37){  
 keyboard\_event('left')  
 }else if(event.keyCode == 38){  
 keyboard\_event('up')  
 }else if(event.keyCode == 39){  
 keyboard\_event('right')  
 }else if(event.keyCode == 40){  
 keyboard\_event('down')  
 }else if(event.keyCode == 40){  
 keyboard\_event('down')  
 }else if(event.keyCode == 9){  
 keyboard\_event('tab')  
 }else if(event.keyCode == 45){  
 keyboard\_event('insert')  
 }else if(event.keyCode == 27){  
 keyboard\_event('escape')  
 }else if(event.keyCode == 46){  
 keyboard\_event('delete')  
 }else if(event.keyCode == 48){  
 char\_event('0',event)  
 }else if(event.keyCode == 49){  
 char\_event('1',event)  
 }else if(event.keyCode == 50){  
 char\_event('2',event)  
 }else if(event.keyCode == 51){  
 char\_event('3',event)  
 }else if(event.keyCode == 52){  
 char\_event('4',event)  
 }else if(event.keyCode == 53){  
 char\_event('5',event)  
 }else if(event.keyCode == 54){  
 char\_event('6',event)  
 }else if(event.keyCode == 55){  
 char\_event('7',event)  
 }else if(event.keyCode == 56){  
 char\_event('8',event)  
 }else if(event.keyCode == 57){  
 char\_event('9',event)  
 }else if(event.keyCode == 192){  
 char\_event('`',event)  
 }else if(event.keyCode == 189){  
 char\_event('-',event)  
 }else if(event.keyCode == 187){  
 char\_event('=',event)  
 }else if(event.keyCode == 219){  
 char\_event('[',event)  
 }else if(event.keyCode == 221){  
 char\_event(']',event)  
 }else if(event.keyCode == 220){  
 char\_event('\\',event)  
 }else if(event.keyCode == 186){  
 char\_event(';',event)  
 }else if(event.keyCode == 222){  
 char\_event('\'',event)  
 }else if(event.keyCode == 188){  
 char\_event(',',event)  
 }else if(event.keyCode == 190){  
 char\_event('.',event)  
 }else if(event.keyCode == 191){  
 char\_event('/',event)  
 }else if(event.keyCode == 65){  
 text\_event('a',event)  
 }else if(event.keyCode == 66){  
 text\_event('b',event)  
 }else if(event.keyCode == 67){  
 text\_event('c',event)  
 }else if(event.keyCode == 68){  
 text\_event('d',event)  
 }else if(event.keyCode == 69){  
 text\_event('e',event)  
 }else if(event.keyCode == 70){  
 text\_event('f',event)  
 }else if(event.keyCode == 71){  
 text\_event('g',event)  
 }else if(event.keyCode == 72){  
 text\_event('h',event)  
 }else if(event.keyCode == 73){  
 text\_event('i',event)  
 }else if(event.keyCode == 74){  
 text\_event('j',event)  
 }else if(event.keyCode == 75){  
 text\_event('k',event)  
 }else if(event.keyCode == 76){  
 text\_event('l',event)  
 }else if(event.keyCode == 77){  
 text\_event('m',event)  
 }else if(event.keyCode == 78){  
 text\_event('n',event)  
 }else if(event.keyCode == 79){  
 text\_event('o',event)  
 }else if(event.keyCode == 80){  
 text\_event('p',event)  
 }else if(event.keyCode == 81){  
 text\_event('q',event)  
 }else if(event.keyCode == 82){  
 text\_event('r',event)  
 }else if(event.keyCode == 83){  
 text\_event('s',event)  
 }else if(event.keyCode == 84){  
 text\_event('t',event)  
 }else if(event.keyCode == 85){  
 text\_event('u',event)  
 }else if(event.keyCode == 86){  
 text\_event('v',event)  
 }else if(event.keyCode == 87){  
 text\_event('w',event)  
 }else if(event.keyCode == 88){  
 text\_event('x',event)  
 }else if(event.keyCode == 89){  
 text\_event('y',event)  
 }else if(event.keyCode == 90){  
 text\_event('z',event)  
 }  
  
 });  
  
  
  
 });  
 </script>  
</div>  
</body>  
</html>