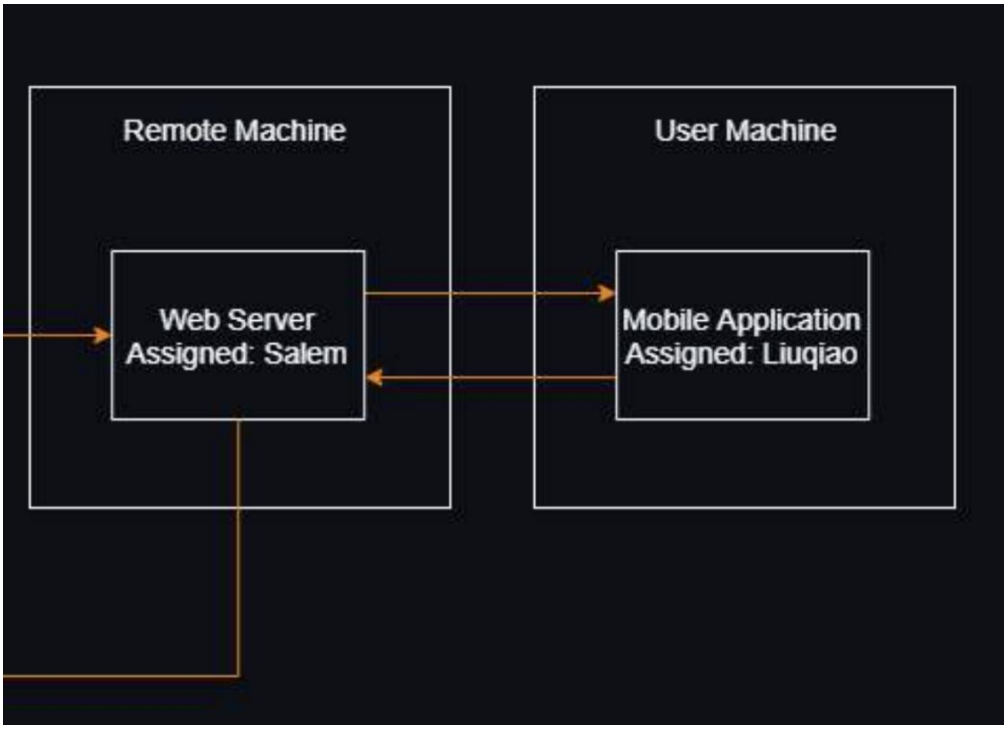
Liuqiao Song
ECE342

Pet Door App development

1 Introduction and block diagram



For the app development assignment, the app need to have some form of login so that you can associate the specific raspi to that app, the login is handled by the server you need to send api requests to the server, then you need to have bluetooth connection so that we can feed wifi password information to the raspi, and the raspi has to connect to the internet to grab real time data and communicate with the server.

The app has to be able to setup schedules for the raspi like when to open the door when to lock it and feed it to the server then the server will send it to raspi, In rhis way we want some basic functionality like the ability to open and close the door from the app and receive hooks from the server when raspi report motion sensing and opening door.

In this project. the server should provide apis to the app and raspi, we'll need authentication and user logins, then transmit app details to raspi and raspi to app

2 Block Interface:

Liuqiao Song
ECE342

Name	Type	Definition
BAT_PWR	DC Power	• Voltage: 5.1V • Current: 1.2A
USB_PWR	DC Power	• Voltage: 5.1V • Current: 1.2A
IR_INT	Analog Signal	• Voltage: DC 5V • Minimium Range: 1m
IR_EXT	Analog Signal	• Voltage: DC 5V • Minimium Range: 1m
USR_IN	Data	• Standards: 2.4GHz IEEE 802.11b/g/n • Protocol: RESTful API • Format: JSON Object

3 Bill of materials

SDK Manager.exe
Remote App.apk
Eclipse.exe

4 Design of the app

4.1 According to the task specified among the team members, mobile application should be guaranteed available on time. And based my knowledge and experience of the recognition of the system, final

decision was made to employ Eliplse, SDK, JDK to realize the predefined assignment.

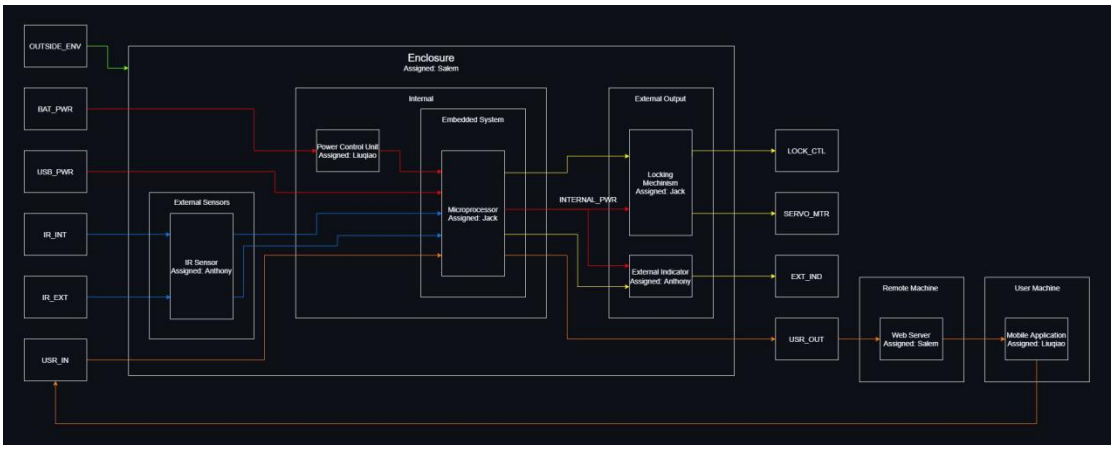


Figure 1. The scheme of the task specified for the team numbers

Which is demonstrated as Figure 1.

4.2 Platform construction

4.2.1 JDK (Java Development Kit) is installed the version of “jdk-8u291-windows-x64”, after is installation, some configuration should be provided to the environmental variable and system variable.

4.2.2 Eliplse and SDK are installed together as a compact bag. The folder is named as “eclipse-SDK_20200429_121741”. Eclipse is “Plug Ins” oriented development, its interface is friendly defined. SDK (Software Development Kit) is invoked by Eclipse and then to serve to our Remote APP design task.

Figure 2. The resulted operating situation

The resulted operating situation is demonstrated as Figure 2.

Theoretically, when we input the legal address of IP and the legal port number together with the API (Applicable Program Interface), we should access successfully to the Server of the other team member designed.

4.3 Mobile APP construction and simulation

4.3.1 Mobile APP construction

It is covered by two different parts. One is Loginactivity.java, and the other one is activity_login.xml, respectively.

The Loginactivity.java part is in charge of the upper side function of Figure 2, and the activity_login.xml plays its role of the lower side function of Figure 2.

The grammar and norms are so sensitive, it is tough to pull them to the right track. It took me long time to be familiar with them.

4.3.2

For my mobile phone is IOS, not Android. In case of convenience, a simulation software to emulate Android mobile phone is employed. Here, “nox_setup_v7.0.0.6_full” is installed to emulate the “RemoteAPP.apk”, and NOX works well.

5 The design of the app interface and the property

//LoginActivity.java

```
package com.example.remoteAPP;

import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.apache.http.protocol.HTTP;
import org.apache.http.util.EntityUtils;
```

Liuqiao Song
ECE342

```
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;


import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import com.example.dynamicttable.R;

import android.widget.Button;
import android.widget.EditText;

public class LoginActivity extends Activity {

        private EditText tv_IP;
        private EditText tv_PORT;
        private EditText tv_JSON;
        private EditText tv_API;
        private Button bt_send;
        private String IPAdd="";
        private final int IS_FINISH=1;
    private ProgressDialog dialog=null;

    private Handler handler =new Handler(){
        @Override
```

```
        public void handleMessage(android.os.Message msg){
                if(msg.what==IS_FINISH){
                        String str=(String) msg.obj;
                        try {
                                dialog.dismiss();

                                        Toast.makeText(LoginActivity.this,  "command  is  finished",
0).show();
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        }
    };
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_login);

                tv_IP=(EditText)findViewById(R.id.tv_IP);
                tv_PORT=(EditText)findViewById(R.id.tv_PORT);
                tv_JSON=(EditText)findViewById(R.id.tv_JSON);
                tv_API=(EditText)findViewById(R.id.tv_API);
                bt_send=(Button)findViewById(R.id.bt_send);
                String content="{'client_type':'app',\n" +
                                "'username':'<some-username>',\n" +
                                "'password':'<some-password>',\n" +
                                "'device_id':'<raspi-device-id>',\n" +
                                "'command':'<some-string>'}";
                tv_JSON.setText(content);
        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
                // Inflate the menu; this adds items to the action bar if it is present.
                getMenuInflater().inflate(R.menu.login, menu);
                return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item){
                switch(item.getItemId()){


                default:
```

```
                        break;
                }

                return super.onOptionsItemSelected(item);

        }

        public void loginButton(View v){
                try {

        IPAdd=tv_IP.getText().toString()+":"+tv_PORT.getText().toString()+"/"+tv_API.getText().toString();
                        dialog=new ProgressDialog(this);
                        dialog.setCancelable(true);
                        dialog.setCanceledOnTouchOutside(false);
                        dialog.setTitle("send command");
                        dialog.setMessage("the command is sending.....");
                        dialog.show();

                        MyThread mythread=new MyThread();
                        mythread.start();

                } catch (Exception e) {
                        // TODO: handle exception
                        e.printStackTrace();
                }
        }

        public class MyThread extends Thread{

        public MyThread(){
        }

        @Override
        public void run(){
                Message message=Message.obtain();
                try{
                        String path="http://"+IPAdd;
                        HttpClient httpClient=new DefaultHttpClient();
                        HttpParams params=httpClient.getParams();
                        HttpConnectionParams.setConnectionTimeout(params, 50000);
                        HttpConnectionParams.setSoTimeout(params, 50000);


                        HttpPost request=new HttpPost(path);
                        List<BasicNameValuePair> parameters=new ArrayList<BasicNameValuePair>();
```

```
                        parameters.add(new BasicNameValuePair("title",tv_JSON.getText().toString()));


                        HttpEntity entity=new UrlEncodedFormEntity(parameters,HTTP.UTF_8);
                        request.setEntity(entity);
                        HttpResponse response=httpClient.execute(request);
                        int statusCode=response.getStatusLine().getStatusCode();

                        if(statusCode==200){
                                entity=response.getEntity();
                                final String result=EntityUtils.toString(entity);
                                message.obj=result;
                                message.what=IS_FINISH;
                                handler.sendMessage(message);
                        }
                        httpClient.getConnectionManager().shutdown();
                }catch(Exception e){
                        e.printStackTrace();
                }
        }
  }

        /*
        public class MyThread implements Runnable{
                private String UserName;
                private String UserPass;
        public MyThread(String name,String pass){
                this.UserName=name;
                this.UserPass=pass;
        }


        @Override
        public void run(){
                Message message=Message.obtain();
                try{
                        String
path="http://"+IPAdd+"/api/padapp/applogin?user="+this.UserName+"&pass="+this.UserPass;
                        HttpClient httpClient=new DefaultHttpClient();
                        HttpParams params=httpClient.getParams();
                        HttpConnectionParams.setConnectionTimeout(params, 5000);
                        HttpConnectionParams.setSoTimeout(params, 5000);
                        HttpGet request=new HttpGet(path);
                        HttpResponse response=httpClient.execute(request);
                        int statusCode=response.getStatusLine().getStatusCode();
```

```
                        if(statusCode==200){
                                HttpEntity entity=response.getEntity();
                                final String result=EntityUtils.toString(entity);
                                message.obj=result;
                                message.what=IS_FINISH;
                                handler.sendMessage(message);
                        }else{
//                              final String result="error";
//                              message.obj=result;
//                              message.what=IS_FINISH;
//                              handler.sendMessage(message);
                        }
                        httpClient.getConnectionManager().shutdown();
                }catch(Exception e){
//                              final String result="error1";
//                              message.obj=result;
//                              message.what=IS_FINISH;
//                              handler.sendMessage(message);
                        e.printStackTrace();
                }
        }
   }*/

}
```

//activity_login.xml

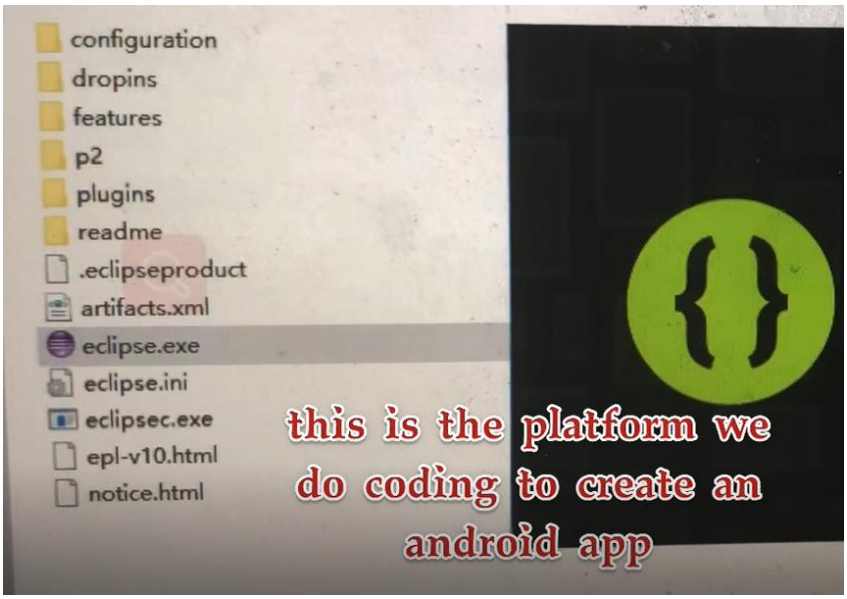
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >


        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:orientation="horizontal">
            <TextView
                  android:layout_width="100dp"
                  android:layout_height="wrap_content"
                  android:textSize="30dp"
                  android:gravity="right"
                  android:text="IP:"/>
```

```
        <EditText
            android:id="@+id/tv_IP"
            android:textSize="30dp"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="134.68.224.135"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:orientation="horizontal">
        <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:textSize="30dp"
            android:gravity="right"
            android:text="PORT:"/>
        <EditText
            android:id="@+id/tv_PORT"
            android:textSize="30dp"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="12345"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:orientation="horizontal">
        <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:textSize="30dp"
            android:gravity="right"
            android:text="api:"/>
        <EditText
            android:id="@+id/tv_API"
            android:textSize="30dp"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="api/name"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:orientation="horizontal">

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize="30dp"
            android:layout_gravity="right"
            android:text="JSON string for send:"/>

    </LinearLayout>
```

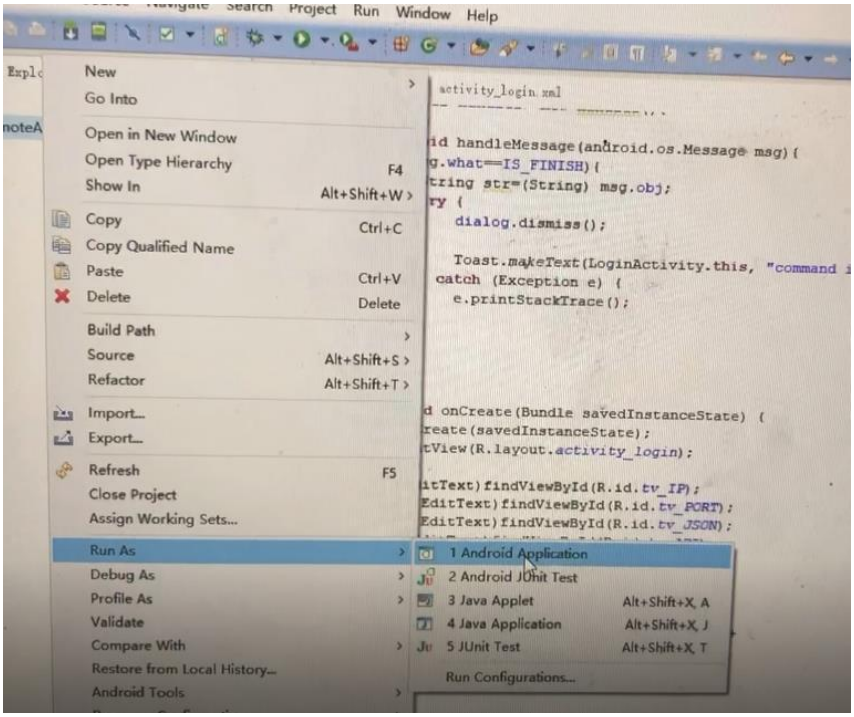
```
        <ScrollView
            android:id="@+id/scrollView1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical" >
                <EditText
                  android:id="@+id/tv_JSON"
                  android:textSize="30dp"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                        android:minLines="10"
                        android:gravity="top"
                  android:text="123456"/>

                <Button
            android:id="@+id/bt_send"
            android:gravity="center"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:paddingTop="10dp"
            android:paddingBottom="10dp"
            android:layout_marginTop="20dp"
              android:textSize="30dp"
            android:background="#004B97"
            android:textColor="#ffffff"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:onClick="loginButton"
            android:text="send"/>

            </LinearLayout>
        </ScrollView>

</LinearLayout>
```

6 Results and how to test

Liuqiao Song
ECE342

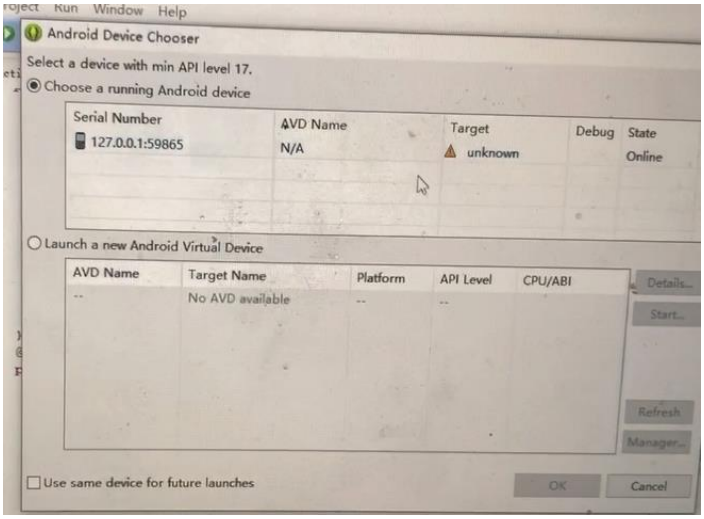


Step 1 open the API development



Step 2 Run the code

Liuqiao Song
ECE342



Successfully complied

Step 3 getting the app

Step 4 connect to the web server and test

For connecting to the server make a TCP connection to 134.68.224.135:12345.

To send from raspi to server:

Liuqiao Song
ECE342

If the raspi need to register itself with the server, it will send the following message:

{"client_type": "raspi",

"device_id" : "<raspi-device-id>"}

If it needs to send a response message to a specific app (username) it needs to send the following:

{"client_type": "raspi",

"device_id" : "<raspi-device-id>",

"username" : "<some-username>",

"response": "<some-string>"}

To send from app to the server:

If the app needs to verify a username/password, it needs to send the following message:

{"client_type": "app",

"username" : "<some-username>",

"password" : "<some-password>"}

If the app needs to send command to a specific raspi, it needs to send the following:

{"client_type": "app",

"username" : "<some-username>",

"password" : "<some-password>",

"device_id": "<raspi-device-id>",

"command": "<some-string>"}

And we could find they are connecting if the web server is working

7 Conclusion and my feelings

7.1 Conclusion

Finally, an artifact is available now, it is precious for me, though it is still so much needed to be improved**.**

7.2 My feelings

I was struggled to overcome the unpredictable difficulties and I insist on continuously. JDK itself installation is not so smooth, environmental variable and system variable configuration neglect prohibit me from forwarding a while. And awkward skills in code editing and program debugging worsen this even further. Anyhow, step by step and asking for help from the team members and learn from the network benefit me a lot**.**

7.3 Acknowledgements

Thank you for my team members in cooperation of this project, and yours patience, generosity and inclusiveness inspire me to go forward. And thank you too, our lecture and mentor in this class, thank you for your help in encouragement. I hope I can do better so soon.

8 Reference

https://developer.android.com/studio/run/emulator

https://developer.apple.com/forums/thread/54124

9 video Recoding

https://youtu.be/Ca8Q6kSc-_Y