Anthony Pacheco | Andy Le | Capstone498 (Spring ’16)

ScanMeIn

Automated Lock System & Companion App (prototype)



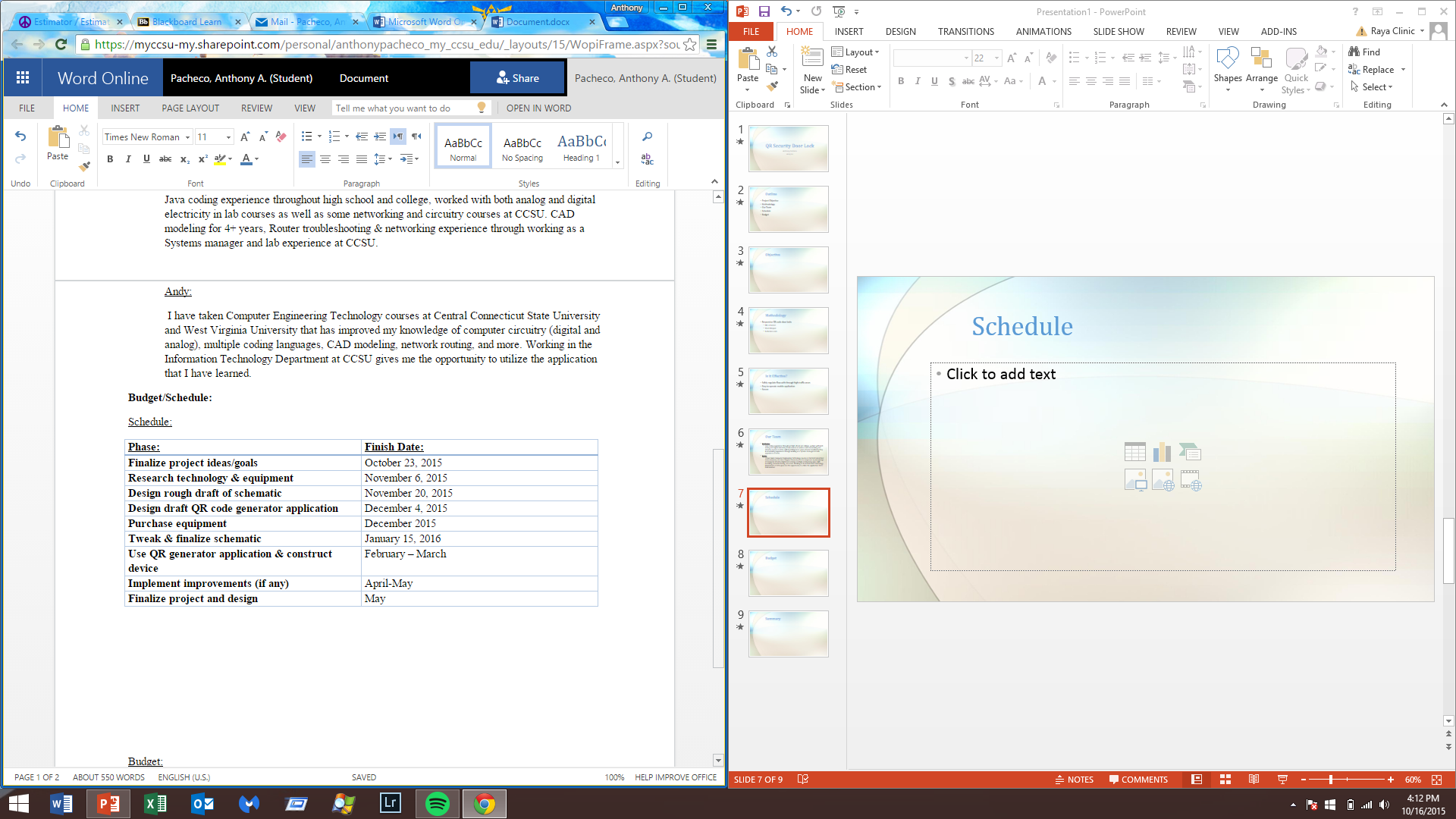
# Intro

Security is always an issue for people in the workplace and at home. Our idea for a product is a new lock system that responds to a scanned barcode. Nowadays, everyone has the convenience a smartphone, so who wants to go digging through their pockets looking for which key opens to door, when they can just use an app on their phone and scan in.

# Our Team

* Anthony:
  + Java and C coding experience throughout high school and college, worked with both analog and digital electricity in lab courses as well as some php and Arduino coding (C++). Experience in CAD modeling for 4+ years, Router troubleshooting & networking experience through working as a Systems manager and lab experience at CCSU.
* Andy:
  + Computer Engineering Technology courses at Central Connecticut State University that has improved my knowledge of computer circuitry (digital and analog), multiple coding languages, CAD modeling, network routing, and more. Working in the Information Technology Department at CCSU gives me the opportunity to utilize the concepts that I have learned.

# Schedule



# Budget

# How It Works

* User Opens the app and is presented with a login screen. If the user does not have an account, they will press the link “Register Here” and will be transferred to the registration form.
* The registration form requires the user to enter their first name, last name, user name and password which are all of type VARCHAR (Variable Character) meaning it can be numbers, letters or symbols. This is to give the users a chance to create a very secure password.
* Upon a successful registration the user will be taken back to the login screen where they can enter the user name and password of their new account.
* When a user has successfully logged in they will view the user area which displays a welcome message and a unique barcode that dynamically changes upon every login.
* Now that the user has logged in, they move the phone screen over to face the scanner and the scanner will read the barcode.
* The scanner is connected to an Arduino controller which executes a query on the database to return a user that matches the barcode value just scanned. If there are any results (there should only be one) the Arduino checks to see if the access value for that user is ‘1’, if it is the door opens. If not, nothing happens.

# Flowcharts

Android

User

Opens

App

no

no

Can the user login?

no

yes

Did the user successfully register?

Did the user successfully login?

yes

yes

User Area

with Barcode

Arduino

Does the user have access?

yes

Is the barcode value associated with a valid user?

no

yes

Open the door

Scan Barcode

no

# Components and Technology

* Parts
  + Arduino Uno R3
  + Arduino USB Shield
  + ESP8266MOD (Wi-Fi Shield)
  + Barcode Scanner
  + FET
  + Breadboard
  + Solenoid 12V, 1A
  + Android mobile device
  + Arduino IDE
  + MySQL Workbench
  + Amazon Web Services (AWS)
  + Android Studio IDE
  + WinSCP(file transmission)

# Benefits - Major convenience via companion app

* + User-Friendly -> very simple screen, not much clutter to confuse the user. The fonts are big and easy to read, no complex designs.
  + No key required

# Peace of mind – Security

* + AmazonWebServices is secure and safe
  + Valid barcode is required to gain access and the value is randomly changed every time the user logs in.

# Obstacles

During the process of developing this product we encountered numerous road blocks, alas we were able to overcome them.

-Database Issues: Trying to use our first database and trying to get it up and running, and then trimming the return statement from executing the query.

Solution: Changed to AmazonWebServices RDS database

-Arduino Issues: Trying to get the Arduino ESP to connect to the database when the MySQL library caused problems with our code, and trying to get it to connect without using php scripts.

Solution: Used an Arduino Developers' library containing methods to connect to the database, mixing implementations around the MySQL library for the ESP Wi-Fi shield.

# The Future

Our goal – Implement what we developed into a system in a school setting where professors will grant students access into a classroom, the implementation will come as a partner of the BlackBoard Learn system where students and professors will have an online User Interface(UI). This will present the professors with a schedule and messaging capability to communicate with students outside of the class room. The professors can view the open timeslots of labs around campus and when a student requests permission to use a lab, the professor can grant access remotely by filling out a timeslot.

# Code

Android

/\*LoginActivity\*/

package com.CCSU.anthony.capstoneapp3;

import android.app.AlertDialog;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import com.android.volley.AuthFailureError;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import java.util.HashMap;

import java.util.Map;

public class LoginActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

final RequestQueue queue = Volley.newRequestQueue(LoginActivity.this);

final String loginUrl = "http://ec2-54-174-119-3.compute-1.amazonaws.com/loginUser.php";

final EditText etUsername = (EditText) findViewById(R.id.etUsername);

final EditText etPassword = (EditText) findViewById(R.id.etPassword);

final Button bLogin = (Button) findViewById(R.id.bLogin);

final TextView registerLink = (TextView) findViewById(R.id.tvRegisterHere);

registerLink.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent registerIntent = new Intent(LoginActivity.this, RegisterActivity.class);

LoginActivity.this.startActivity(registerIntent);

}

});

bLogin.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

final AlertDialog.Builder builder = new AlertDialog.Builder(LoginActivity.this);

final Intent intent = new Intent(LoginActivity.this, UserAreaActivity.class);

final StringRequest lrequest = new StringRequest(Request.Method.POST, loginUrl, new Response.Listener<String>()

{

@Override

public void onResponse(String response)

{

Log.v(response, "Server Response");

if((response.contains("Successful Login")) || (response.contains("SL")) ||(response.contains("SL:Successful Login")))

{

builder.setMessage("Successful Login! ")

.create()

.show();

LoginActivity.this.startActivity(intent);

}else{

Log.v(response, "Server Response");

builder.setMessage("Login Failed: \n Invalid Credentials....\t Please check to make sure all fields have been filled out correctly. ")

.setNegativeButton("Retry", null)

.create()

.show();

}

}

},

new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError response) {

}

})

{

@Override

public Map<String, String> getParams() throws AuthFailureError {

Map<String, String> params = new HashMap<String, String>();

params.put("Username", etUsername.getText().toString());

params.put("Password", etPassword.getText().toString());

return params;

}

};

queue.add(lrequest);

}

});

}

}

/\*RegisterActivity\*/

package com.CCSU.anthony.capstoneapp3;

import android.app.AlertDialog;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import com.android.volley.AuthFailureError;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import java.util.HashMap;

import java.util.Map;

public class RegisterActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_register);

final RequestQueue queue = Volley.newRequestQueue(RegisterActivity.this);

final String registerUrl = "http://ec2-54-174-119-3.compute-1.amazonaws.com/registerUser.php";

/\* creates fields for the user to input information \*/

final EditText etFName = (EditText) findViewById(R.id.etFName);

final EditText etLName = (EditText) findViewById(R.id.etLName);

final EditText etUsername = (EditText) findViewById(R.id.etUsername);

final EditText etPassword = (EditText) findViewById(R.id.etPassword);

/\* creates a button to send the information to the database \*/

final Button bRegister = (Button) findViewById(R.id.bRegister);

bRegister.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

final Intent intent = new Intent(RegisterActivity.this, LoginActivity.class);

final AlertDialog.Builder builder = new AlertDialog.Builder(RegisterActivity.this);

StringRequest rrequest = new StringRequest(Request.Method.POST, registerUrl, new Response.Listener<String>(){

@Override

public void onResponse(String response) {

Log.v(response, "Registration Response: ");

if((response.contains("Successful Registration")) || (response.contains("SR")) ||(response.contains("SR:Successful Registration")))

{

builder.setMessage("Successful Registration! ")

.create()

.show();

RegisterActivity.this.startActivity(intent);

}else{

if((response.contains("User Already Exists")) || (response.contains("UAE")) ||(response.contains("UAE:User Already Exists")))

{

builder.setMessage("User Already Exists... Try Different Credientals ")

.setNegativeButton("Retry", null)

.create()

.show();

}else {

builder.setMessage("Registration Failed: \n Please check to make sure all fields have been filled out correctly. ")

.setNegativeButton("Retry", null)

.create()

.show();

}

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError response) {

}

}){

@Override

public Map<String, String> getParams() throws AuthFailureError {

Map<String, String> params = new HashMap<String, String>();

params.put("FirstName", etFName.getText().toString());

params.put("LastName", etLName.getText().toString());

params.put("Username", etUsername.getText().toString());

params.put("Password", etPassword.getText().toString());

Log.v(params.toString(), "VERBOSE params.put...");

return params;

}

};

queue.add(rrequest);

}

});

}

}

/\*UserAreaActivity\*/

package com.CCSU.anthony.capstoneapp3;

import android.content.Intent;

import android.graphics.drawable.Drawable;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import java.util.Random;

public class UserAreaActivity extends AppCompatActivity {

/\*images\*/

int[] pics = {R.drawable.a0, R.drawable.a1, R.drawable.a2, R.drawable.a3, R.drawable.a4,

R.drawable.a5, R.drawable.a6, R.drawable.a7, R.drawable.a8, R.drawable.a9, R.drawable.a10,

R.drawable.a11, R.drawable.a12, R.drawable.a13, R.drawable.a14, R.drawable.a15};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_user\_area);

final TextView welcomeMessage = (TextView) findViewById(R.id.tvWelcomeMsg);

final ImageView iVpic = (ImageView) findViewById(R.id.iVpic);

Button bLogout = (Button) findViewById(R.id.bLogout);

final Intent intent = new Intent(UserAreaActivity.this, LoginActivity.class);

bLogout.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

UserAreaActivity.this.startActivity(intent);

}

});

Random r = new Random();

int i = r.nextInt((pics.length) - 1);

if (i != pics.length) {

Drawable d = getResources().getDrawable(pics[i], getApplicationContext().getTheme());

/\* if .setImageDrawable(d) doesn't work. try iVpic.setImageResource(pics[i])

\* and get rid of Drawable d.

\*

\*

\* -->or try using a Bitmap with the code below.<--

\*

\*

\* Bitmap bPic = BitmapFactory.decodeResource(this.getResources(), pics[i]);

\* iVpic.setImageBitmap(bpic); <-- this would go in the case statement above "i++;"

\*

\*

\* \*/

switch (pics[i]) {

case R.drawable.a0:

iVpic.setImageDrawable(d);

break;

case R.drawable.a1:

iVpic.setImageDrawable(d);

break;

case R.drawable.a2:

iVpic.setImageDrawable(d);

break;

case R.drawable.a3:

iVpic.setImageDrawable(d);

break;

case R.drawable.a4:

iVpic.setImageDrawable(d);

break;

case R.drawable.a5:

iVpic.setImageDrawable(d);

break;

case R.drawable.a6:

iVpic.setImageDrawable(d);

break;

case R.drawable.a7:

iVpic.setImageDrawable(d);

break;

case R.drawable.a8:

iVpic.setImageDrawable(d);

break;

case R.drawable.a9:

iVpic.setImageDrawable(d);

break;

case R.drawable.a10:

iVpic.setImageDrawable(d);

break;

case R.drawable.a11:

iVpic.setImageDrawable(d);

break;

case R.drawable.a12:

iVpic.setImageDrawable(d);

break;

case R.drawable.a13:

iVpic.setImageDrawable(d);

break;

case R.drawable.a14:

iVpic.setImageDrawable(d);

break;

case R.drawable.a15:

iVpic.setImageDrawable(d);

break;

}

}

}

}

/\*Android Manifest\*/

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.CCSU.anthony.capstoneapp3">

<uses-permission android:name="android.permission.INTERNET"/>

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name="com.CCSU.anthony.capstoneapp3.LoginActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name="com.CCSU.anthony.capstoneapp3.RegisterActivity" />

<activity android:name="com.CCSU.anthony.capstoneapp3.UserAreaActivity" ></activity>

</application>

</manifest>

/\*Login Layout (xml) \*/

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="@drawable/bdl"

tools:context="com.CCSU.anthony.capstoneapp3.LoginActivity">

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/etUsername"

android:layout\_alignParentTop="true"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true"

android:hint="Username" />

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:inputType="textPassword"

android:ems="10"

android:id="@+id/etPassword"

android:layout\_below="@+id/etUsername"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true"

android:hint="Password" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Login"

android:id="@+id/bLogin"

android:layout\_below="@+id/etPassword"

android:layout\_centerHorizontal="true" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceMedium"

android:text="Register Here"

android:id="@+id/tvRegisterHere"

android:layout\_marginTop="10dp"

android:layout\_below="@+id/bLogin"

android:layout\_centerHorizontal="true" />

</RelativeLayout>

/\*Resigter Layout (xml)\*/

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="@drawable/bdl"

tools:context="com.CCSU.anthony.capstoneapp3.RegisterActivity">

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/etFName"

android:layout\_alignParentTop="true"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true"

android:hint="First Name" />

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/etLName"

android:layout\_below="@+id/etFName"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignRight="@+id/etFName"

android:layout\_alignEnd="@+id/etFName"

android:hint="Last Name" />

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/etUsername"

android:layout\_below="@+id/etLName"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignRight="@+id/etLName"

android:layout\_alignEnd="@+id/etLName"

android:hint="Username" />

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:inputType="textPassword"

android:ems="10"

android:id="@+id/etPassword"

android:layout\_below="@+id/etUsername"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignRight="@+id/etUsername"

android:layout\_alignEnd="@+id/etUsername"

android:hint="Password" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Register"

android:id="@+id/bRegister"

android:layout\_centerHorizontal="true"

android:layout\_below="@+id/etPassword" />

</RelativeLayout>

/\*User Area Lyout (xml) \*/

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.CCSU.anthony.capstoneapp3.UserAreaActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge"

android:text="Welcome"

android:id="@+id/tvWelcomeMsg"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="30dp"

android:textSize="30dp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Logout"

android:id="@+id/bLogout"

android:layout\_above="@+id/iVpic"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="10dp"/>

<ImageView

android:layout\_width="250dp"

android:layout\_height="250dp"

android:id="@+id/iVpic"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="50dp" />

</RelativeLayout>

/\*Build Gradle \*/

apply plugin: 'com.android.application'

android {

compileSdkVersion 23

buildToolsVersion "23.0.2"

defaultConfig {

applicationId "com.CCSU.anthony.capstoneapp3"

minSdkVersion 21 /\*previously 15, I changed it to 21 to support getDrawable() on 4/6/2016 @ 7:27pm\*/

targetSdkVersion 23

versionCode 2

versionName "1.01"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

compile fileTree(dir: 'libs', include: ['\*.jar'])

testCompile 'junit:junit:4.12'

compile 'com.android.support:appcompat-v7:23.1.1'

compile 'com.android.volley:volley:1.0.0'

}

/\*\*\*\*\*\*\*PHP CODE\*\*\*\*\*\*\*/

/\*Login User\*/

<?php

header('Content-Type: application/json');

$response = array();

$mysqli = new mysqli("thisinstance.cy3jxhjvzmqz.us-east-1.rds.amazonaws.com:3306", "Muser", "mpassword", "MyDB");

/\* check connection \*/

if (mysqli\_connect\_errno())

{

$response["DCE"] = mysqli\_connect\_error();

echo json\_encode($response);

exit();

}

if(isset($\_POST["Username"]) && isset($\_POST["Password"]))

{

$Username = (string)$\_POST["Username"];

$Uname = $Username;

$Password = (string)$\_POST["Password"];

if ($Username && $Password)

{

if ($stmt = $mysqli->prepare("SELECT \* FROM users WHERE Username=? AND Password=?"))

{

if ($stmt->bind\_param("ss", $Username, $Password))

{

if ($stmt->execute())

{

$result = $stmt->get\_result();

if ($row = $result->fetch\_assoc())

{

$BV = $row["BarcodeValue"];

$response["SL"] = "Successful Login";

$BarcodeValue = (int) rand(48,10048);

randvalcheck($BarcodeValue);

if ($stmtBV = $mysqli->prepare("UPDATE users SET BarcodeValue=? WHERE Username='$Uname'"))

{

if ($stmtBV->bind\_param("i", $BarcodeValue))

{

if ($stmtBV->execute())

{

$result->free();

}

}else{$response["PNB"] = "Parameters Could Not Be Bound";}

}else{ $response["BVE"] = "Barcode Value Error";}

} else $response["BUP"] = "Bad username or password";

} else $response["QE"] = "Query did not execute properly";

} else $response["QE"] = "Query parameters could not be bound";

/\* close statement \*/

$stmt->close();

} else $response["QE"] = "Query could not be prepared";

} else $response["BR"] = "Bad POST request parameters";

} else $response["BR"] = "Bad POST request parameters";

/\* close connection \*/

$mysqli->close();

echo json\_encode($response);

?>

/\*Register User\*/

<?php

header('Content-Type: application/json');

$response = array();

$mysqli = new mysqli("thisinstance.cy3jxhjvzmqz.us-east-1.rds.amazonaws.com:3306", "Muser", "mpassword", "MyDB");

/\* check connection \*/

if (mysqli\_connect\_errno()) {

$response["DCE"] = mysqli\_connect\_error();

echo json\_encode($response);

exit();

}

if(isset($\_POST["FirstName"]) && isset($\_POST["LastName"]) && isset($\_POST["Username"]) && isset($\_POST["Password"]))

{

$FirstName = (string)$\_POST["FirstName"];

$LastName = (string)$\_POST["LastName"];

$Username = (string)$\_POST["Username"];

$Password = (string)$\_POST["Password"];

$Access = (int) 1;

$BarcodeValue = (int) rand(48,10048);

if ($FirstName && $LastName && $Username && $Password) {

/\* create a prepared statement \*/

if ($stmt = $mysqli->prepare("SELECT Username FROM users WHERE Username=?")) {

/\* bind parameters for markers \*/

if ($stmt->bind\_param("s", $Username)) {

/\* execute query \*/

if ($stmt->execute()) {

$result = $stmt->get\_result();

if ($row = $result->fetch\_assoc()) {

$response["UAE"] = "User already exists";

$result->free();

$stmt->close();

} else {

$result->free();

$stmt->close();

randvalcheck($BarcodeValue);

if ($stmt = $mysqli->prepare("INSERT INTO users (Username,Password,FirstName,LastName,BarcodeValue,Access) values (?,?,?,?,?,?)")) {

if ($stmt->bind\_param("ssssii", $Username, $Password, $FirstName, $LastName, $BarcodeValue, $Access)) {

if ($stmt->execute()) $response["SR"] = "Successful registration";

else $response["QE"] = "Query 2 did not execute properly";

} else $response["QE"] = "Query 2 parameters could not be bound";

} else $response["QE"] = "Query 2 could not be prepared";

}

} else $response["QE"] = "Query 1 did not execute properly";

} else $response["QE"] = "Query 1 parameters could not be bound";

} else $response["QE"] = "Query 1 could not be prepared";

} else $response["BR"] = "Bad POST request parameters";

} else $response["BR"] = "Bad POST request parameters";

/\* close connection \*/

$mysqli->close();

echo json\_encode($response);

?>

/\*\*\*\*\*\*\*Arduino Code\*\*\*\*\*\*\*/

/\*Barcode Scanner\*/

/\*

Central Connecticut State University - CET Senior Capstone 2016

This code uses an Arduino Uno R3 with a 2D barcode scanner and

USB 2.0 Shield Hub. When a barcode is scanned, it is then converted

from ASCII to decimal and sent serially to the ESP8266.

Created by: Andy Le & Anthony Pacecho

\*/

// calls libraries

#include <hidboot.h>

#include <HID.h>

#include <usbhub.h>

#ifdef dobogusinclude

#include <spi4teensy3.h>

#include <SPI.h>

#endif

char BarcodeBuffer[20] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; // Barcode scanner buffer

int BarcodeBufferIndex = 0 ; // Pointer barcode scanner buffer

boolean BarcodeComplete = false ; // Scan complete?

class KbdRptParser : public KeyboardReportParser

{

protected:

void OnKeyDown (uint8\_t mod, uint8\_t key);

void OnKeyUp (uint8\_t mod, uint8\_t key);

void OnKeyPressed(uint8\_t key);

};

void KbdRptParser::OnKeyDown(uint8\_t mod, uint8\_t key)

{

uint8\_t c = OemToAscii(mod, key); // ASCII conversion

if (c)

if (mod != 1) OnKeyPressed(c); // if control pressed then cancel character

}

void KbdRptParser::OnKeyUp(uint8\_t mod, uint8\_t key)

{

if (key==40)

{ // check for "enter" new line this is end of barcode

BarcodeComplete = true ;

}

}

void KbdRptParser::OnKeyPressed(uint8\_t key)

{

BarcodeBuffer[BarcodeBufferIndex] = key;

BarcodeBufferIndex ++;

}

USB Usb;

HIDBoot<HID\_PROTOCOL\_KEYBOARD> Keyboard(&Usb);

uint32\_t next\_time;

KbdRptParser Prs;

void setup()

{

Serial.begin(115200);

Serial.println("Ready to scan");

#if !defined(\_\_MIPSEL\_\_)

while (!Serial); // Wait for serial port to connect

#endif

if (Usb.Init() == -1)

Serial.println("OSC did not start.");

delay( 200 );

next\_time = millis() + 5000;

Keyboard.SetReportParser(0, &Prs);

}

void loop()

{

BarcodeBufferIndex = 0;

while (!BarcodeComplete) // waiting to scan barcode

{

Usb.Task();

}

BarcodeComplete = false ; // when barcode is scanned

BarcodeBufferIndex --;

Serial.write(BarcodeBuffer,BarcodeBufferIndex); // send barcode value serially

//Serial.write('\n');

}

/\*MySQL Connection\*/

/\*

Central Connecticut State University - CET Senior Capstone 2016

This code uses an ESP8266 Wi-Fi Shield to connect to a MySQL server

to retrieve data to validate if a user has access to unlock a door.

INSTRUCTIONS FOR USE

1) Change the address of the server to the IP address of the MySQL server

2) Change the user and password to a valid MySQL user and password

3) Change the SSID and pass to match your WiFi network

4) Connect a USB cable to your Arduino

5) Select the correct board and port

6) Compile and upload the sketch to your Arduino

7) Once uploaded, open Serial Monitor (use 115200 speed) and observe

Note: The MAC address can be anything so long as it is unique on your network.

Created by: Andy Le & Anthony Pacecho

Using ESP8266WiFi library (created by: and Ivan Grokhotkov)

mysql\_connector library (created by: Dr. Charles A. Bell)

\*/

// calling libraries

#include "ESP8266WiFi.h"

#include "sha1.h"

#include "mysql.h"

#include "Ethernet.h"

byte mac\_addr[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress server\_addr(52,202,174,47); // IP of the MySQL \*server\* here

// MySQL login info

char user[] = "Muser"; // MySQL user login username

char password[] = "mpassword"; // MySQL user login password

// WiFi login info

const char\* ssid = "AndysPhone"; // your SSID

const char\* pass = "Central123"; // your SSID Password

int IncomingBarcode = 0; // Barcode value from scanner

int UserID = 0;

int UnlockDoor = D6; // sets pin 6 to unlock door

WiFiClient client; // Use this for WiFi instead of EthernetClient

Connector my\_conn; // The Connector/Arduino reference

void setup() {

Serial.begin(115200);

pinMode(UnlockDoor, OUTPUT); // initialize UnlockDoor as output

// Begin WiFi section

WiFi.begin(ssid, pass);

while ( WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

delay(50);

// End WiFi section

// connect to MySQL server

Serial.println("Connecting to DB...");

if (my\_conn.mysql\_connect(server\_addr, 3306, user, password)){

}

else

Serial.println("Connection failed.");

}

void loop() {

digitalWrite(UnlockDoor, LOW); // door is usually locked

delay(100);

if (Serial.available() > 0) // checks for serial input from scanner

{

IncomingBarcode = Serial.read(); // IncomingBarcode is the scanned barcode walue

Serial.print("Barcode scanned: ");

Serial.println(IncomingBarcode);

const char QUERY\_DATA[] = "SELECT \* FROM MyDB.users WHERE BarcodeValue = '%lu' && Access ='1';"; //Query statement to be executed

char query[128]; //destination array

sprintf(query, QUERY\_DATA, IncomingBarcode); // method used to insert IncomingBarcode value

// dynamically into query statement

my\_conn.cmd\_query(query); // execute query statement

Serial.println("Execute: ");

Serial.println(query);

my\_conn.get\_columns();

row\_values \*row = NULL;

do {

row = my\_conn.get\_next\_row();

if(row != NULL) {

int UserID = atol(row->values[0]); // grab UserID from DB

if (UserID > 0) // if UserID is found with vaild BarcodeValue && Access = 1

{ // then opens door for user!!

Serial.print("Access granted for UserID: ");

Serial.println(UserID);

digitalWrite(UnlockDoor, HIGH); // unlocks door

delay(3000);

}

} // if not valid, do nothing

my\_conn.free\_row\_buffer(); // clears the buffers to save memory

} while (row != NULL);

my\_conn.free\_columns\_buffer();

}

}