



RFID СИСТЕМА ЗА КОНТРОЛ НА ДОСТЪПА

Йордан Живков Йорданов

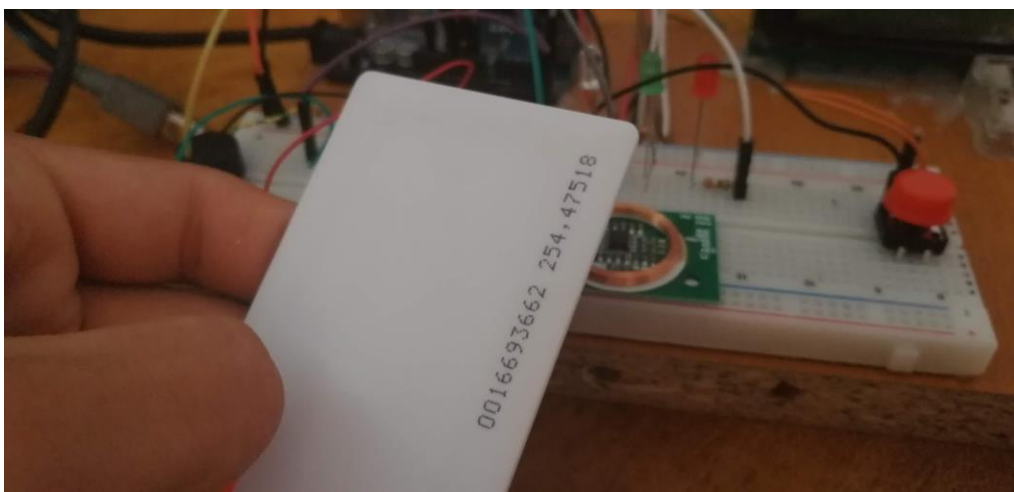


JULY 4, 2019

RFID – радиочестотна идентификация

Какво е RFID?

Технологията за радиочестотна идентификация е подготвена за някои значителни подобрения, стимулирани от това, което според анализаторите ще бъде по-бърз растеж



в здравеопазването, търговията на дребно, безопасността на храните и други пазари. Бъдещето на RFID се разраства и разширява, тъй като повече индустрии и компании инвестират в технологията. В резултат на това RFID става все по-рентабилен от всякога за решаване на реални бизнес предизвикателства.

Технологията за радиочестотна идентификация (RFID) използва електромагнитни полета за предаване на данни от RFID етикет към четец, като осигурява точни данни за проследяване в реално време. За разлика от баркод скенери, RFID четците не се насочват директно към етикет, за да получат сканиране.

Анализи и прогнози

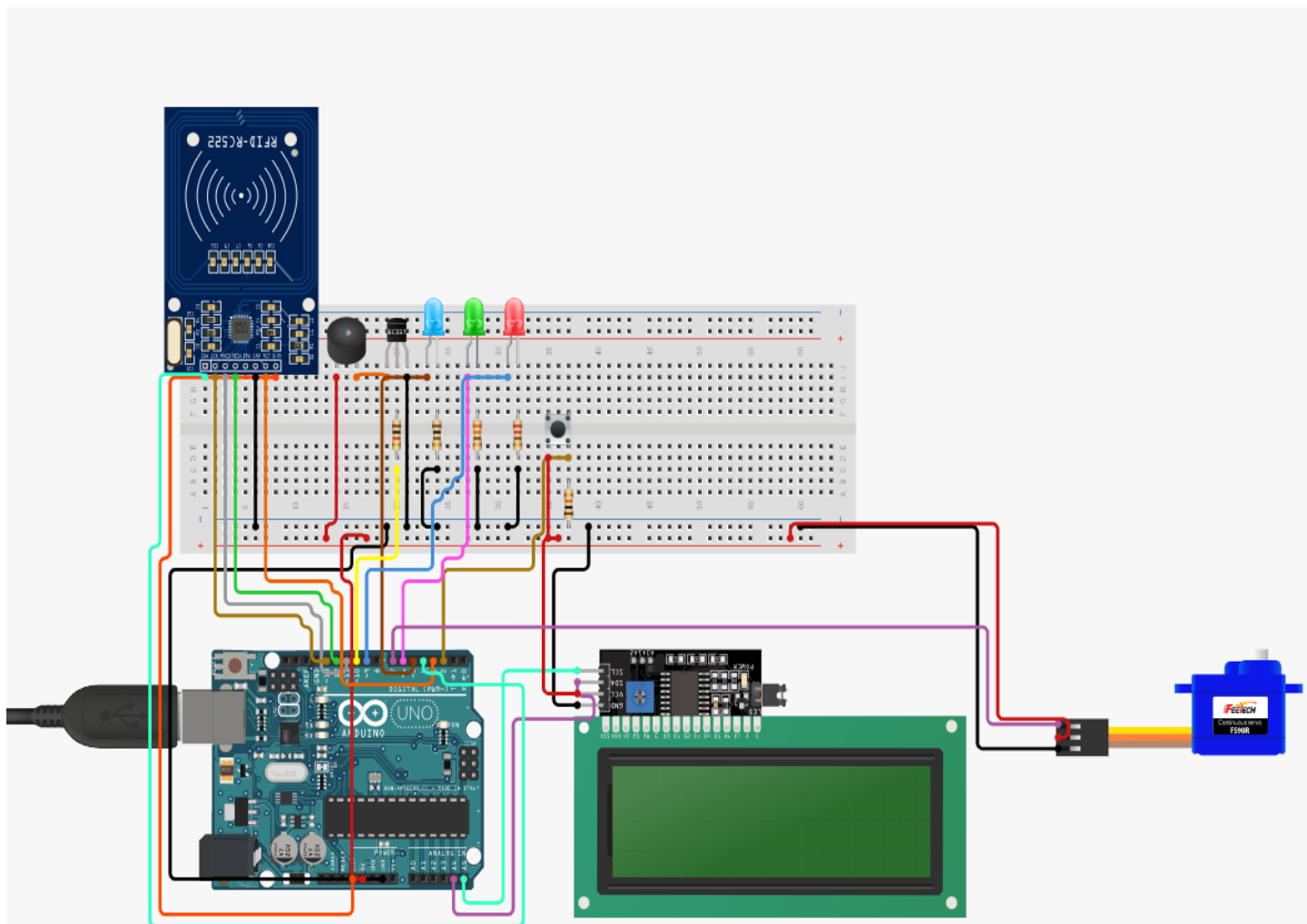
Анализаторите предвиждат експлозия на RFID за последното десетилетие; пазарът обаче нараства бавно и стабилно. Докладът за проучвания и пазари обаче предвижда глобално снабдяване с RFID тагове да има сложен годишен темп на растеж от 22,4% до 2018 г. Още един доклад поставя пазара на интелигентни етикети на 10 милиарда долара до 2020 година. Някои експерти смятат, че пазарът на пасивни RFID етикети само ще нарасне до почти 7 милиарда етикети през следващата година.

Бъдещи иновации за RFID

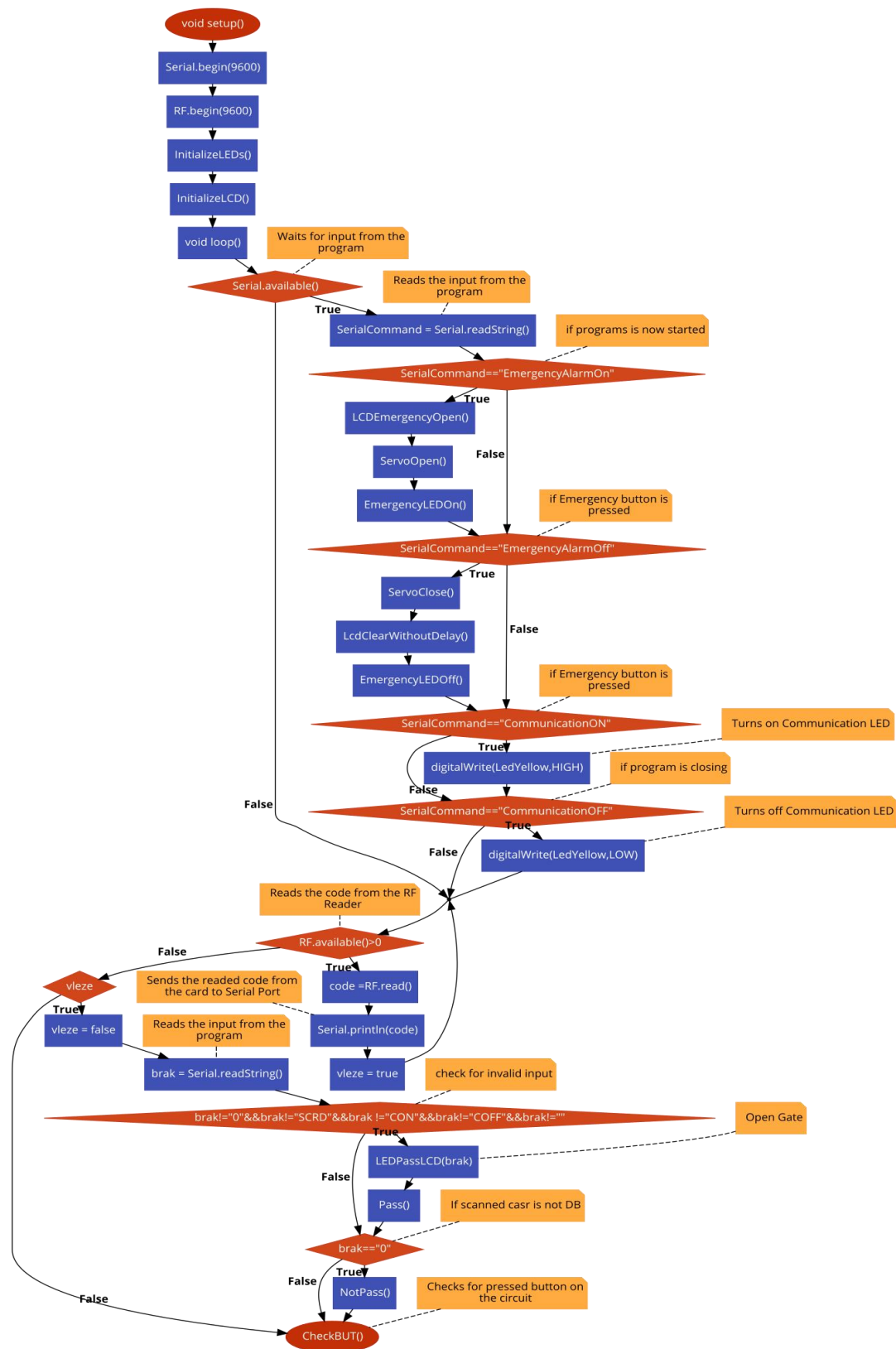
Наред с по-широкото внедряване, новите технологии ще направят RFID по-надежден и рентабилен за по-голям брой приложения.



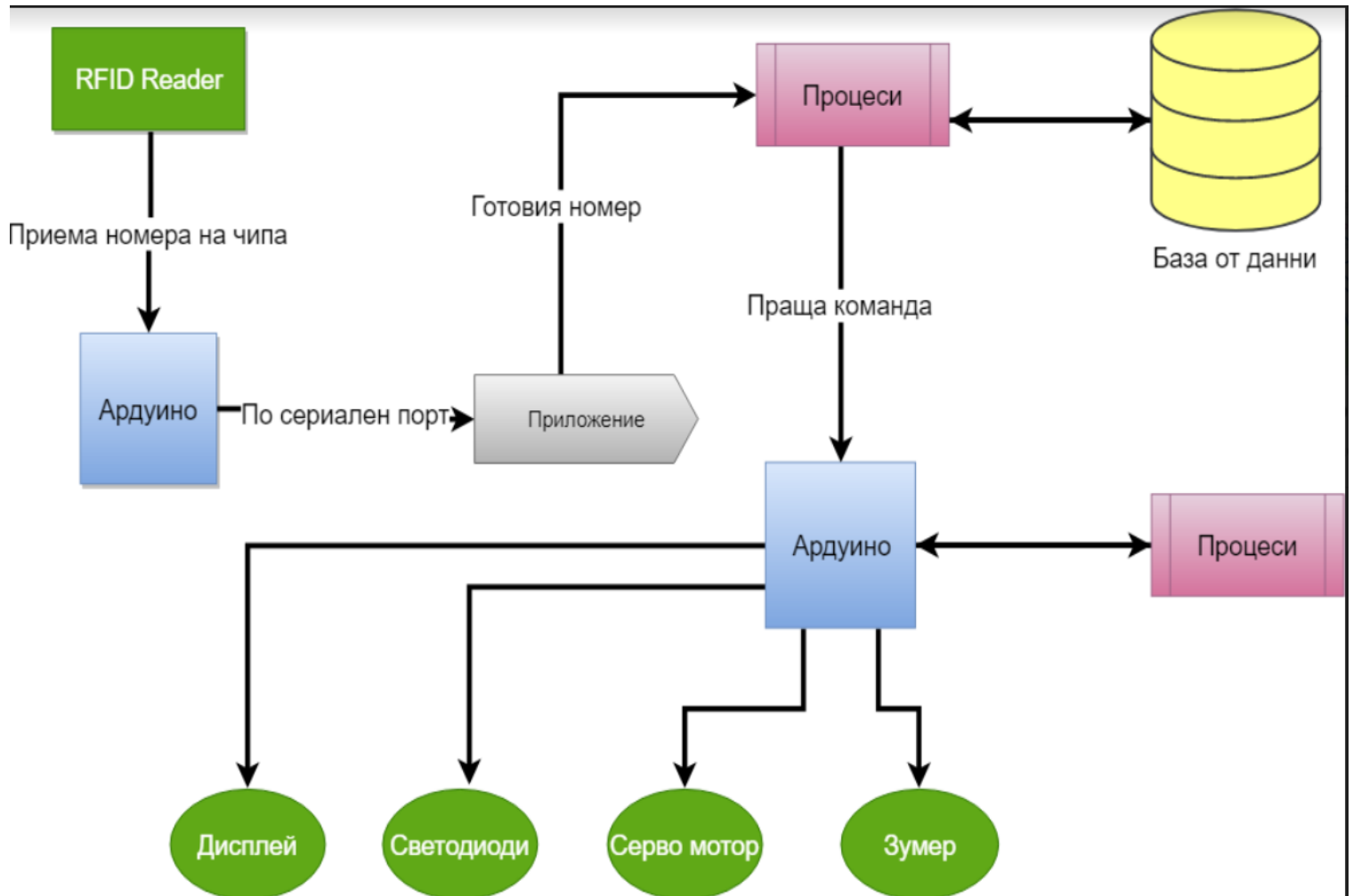
Електрическа схема



Блокова схема на програмното осигуряване за хардуера



Блокова схема на Софтуер и Хардуер



Employee-Management

1. Тема:

Employee-Management – това е проект, който представлява софтуер управляващ хардуер. Темата на този проект е RFID система за контрол на достъпа. Целта на проекта е да се ограничи достъпа на външни лица до дадена сграда и да се пази в база от данни кой кога е влизал и колко часа е работил месечно.

2. Автор:

- **Йордан Живков Йорданов**, ЕГН: 0143285804, телефон: 088 980 3172, имейл: dani_yordani@abv.bg, училище: ППМГ „Добри Чинтулов“, клас: 11^А
- **Ръководител:** доц. д-р **Димитър Минчев**, тел: +359 899 148 872 , имейл: dimitar.minchev@gmail.com, преподавател към Център по информатика и технически науки при Бургаски свободен университет (БСУ).

3. Резюме:

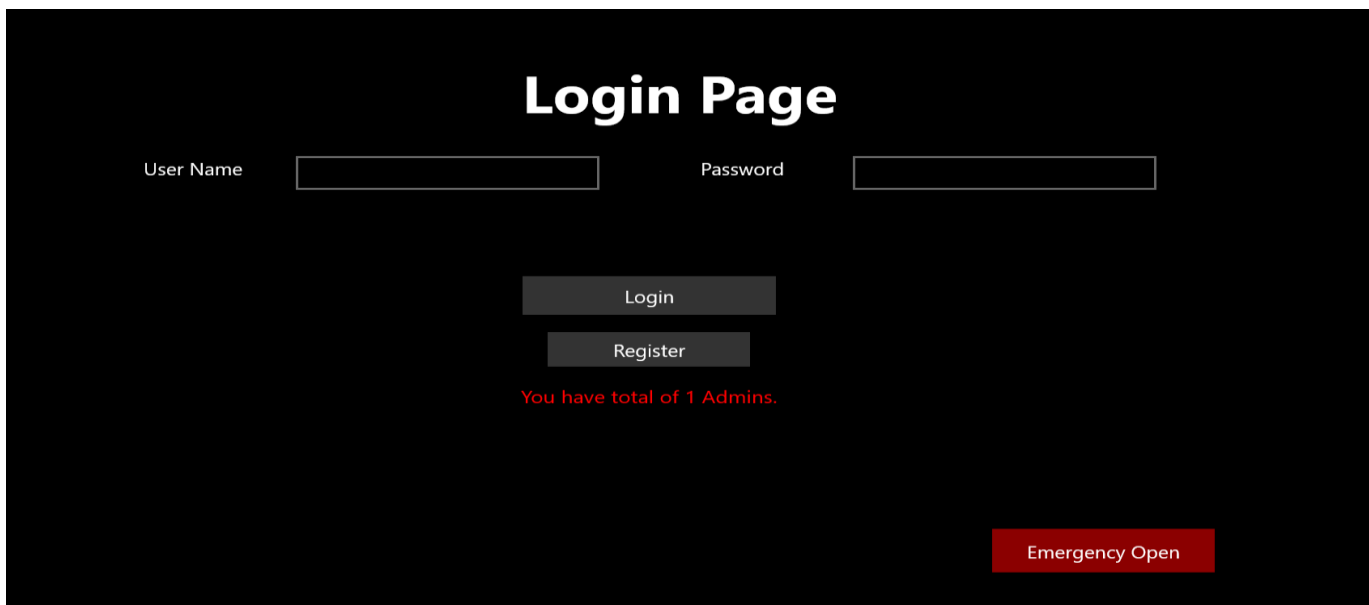
3.1.Пректът Employee-Management се състои от софтуер и хардуер. Софтуерът е разработен на Visual Studio и на платформата UWP (Универсалната Windows платформа) създадена от Microsoft използвайки: C ++ , C # , VB.NET и XAML. Софтуерът има за цел да складира в база от данни всички служители (с тяхната информация) от една компания. Предоставя информация за всеки служител кога е влязъл и напуснал сградата и колко часа е работил. Също пази информация колко часа е работил служител за целия месец (което улеснява калкулирането му на месечна заплата).

3.2. Хардуерът представлява RFID четец който прочита индефициращият код на една карта или чип и го праща по сериален порт на Софтуерът. Има и бутон който служи за отключване на ключалката ръчно без да се нуждае Хардуерът от връзка с база от данни (Софтуерът). За улесняването на потребителя съм предоставил светодиоди които предоставят информация (кога има комуникация между Хардуер и Софтуер, кога може да влезе и кога не може) и също така и Дисплей който предоставя името и фамилията на човека сканирал чипа (Дисплеят показва чрез текст кога ключалката е отворена поради извънредна ситуация или ръчно някой я е задействал).

4. Основните етапи в реализирането на проекта са разделени в няколко групи:

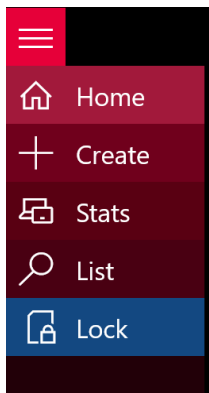
- оформяне на основната идея на проекта;
- схема на Хардуерната част
- Реализиране на основната част от Хардуерът
- Създаване на база от данни
- Създаване на страници
- Валидация на страници
- Създаване на логиката по MVC модел.
- Оправяне на конвенциите за писане на код
- Завършване на Хардуерът

5. Описание на Софтуерът



The screenshot shows a dark-themed login page. At the top center, the title "Login Page" is displayed in a large, white, sans-serif font. Below the title, there are two input fields: "User Name" on the left and "Password" on the right, both with white text labels and empty text boxes. Underneath these fields, there are two buttons: a "Login" button and a "Register" button, both with white text on a dark background. Below the "Register" button, there is a red text message: "You have total of 1 Admins.". In the bottom right corner, there is a red button with white text that says "Emergency Open".

При стартирането на проекта сме посрещнати от Логин страницата която изисква от нас да въведем потребителско име и парола и ни дава възможност да се отправим към Регистър страницата с бутона „Register“ . Текстовото поле под този бутон ни предоставя колко активни потребителя имаме в текущо време. „Emergency Open“ бутона ни дава опцията да отключим всички ключалки при извънредна ситуация.



Навигационият панел ни предоставя 5 опции за пренасочване към отделните страници.

- Първата ни препраща към главната страница на приложението.
- Втората ни препраща към страница за създаване на потребители
- Третата ни препраща към страницата която показва кой кога е влизал и излязъл
- Четвъртата ни препраща към списък от всички служители
- Петата заключава приложението и го препраща към>Login страницата

Create Page

First Name	<input type="text"/>	Second Name	<input type="text"/>
Last Name	<input type="text"/>	EGN	<input type="text"/>
Town	<input type="text"/>	Duty	<input type="text"/>
Phone Number	<input type="text"/>	Card Number	<input type="text"/>

Scan

Get Code

Create

Това е Create страницата която има за предназначение да създава нови служители. Не може да се създаде нов служител дори и ако едно от полетата е непълнено . Полето „Card Number“ се попълва по следния начин: 1- натиска се бутон „Scan“ след това се сканира чипа който ще пренадлежи на дадения служител и чак след това се натиска бутона „Get Code“ който поставя идентификационния номер на сканираната карта в текстовото поле.

Това е страницата която ни предоставя информация за влизането и излизането на служителите.

Имаме падащ списък който ни предоставя две опции да видим кой и кога е влязъл също колко време е работил (ако е излязъл се разбира колко часа е работил) за днеска. А другата опция ни предоставя същите данни като предишната но само че за периода от един месец.

Бутона „Refresh“ ни обновява списака и го прави актуален.

Employee Info Page							
FirstName	SecondName	LastName	EGN	Duty	Town	TelephoneNumber	ScannerCardNumber
Jordan	Jivkov	Jordanov	0143285804	CEO	Sliven	0889994541	21023802!
Plamen2	Vasilev2	Vasilev	028786278175	Biolog	Varna	0878752578	21021041!
Test	Test	Test	Test	Test	Test	Test	21023802!
CreateEmployee							

Тази страница ни предоставя списък с всички регистрирани служители с тяхна най важна информация при натискане на даден служител от списъка страницата ни препраща в „Edit“ страницата.

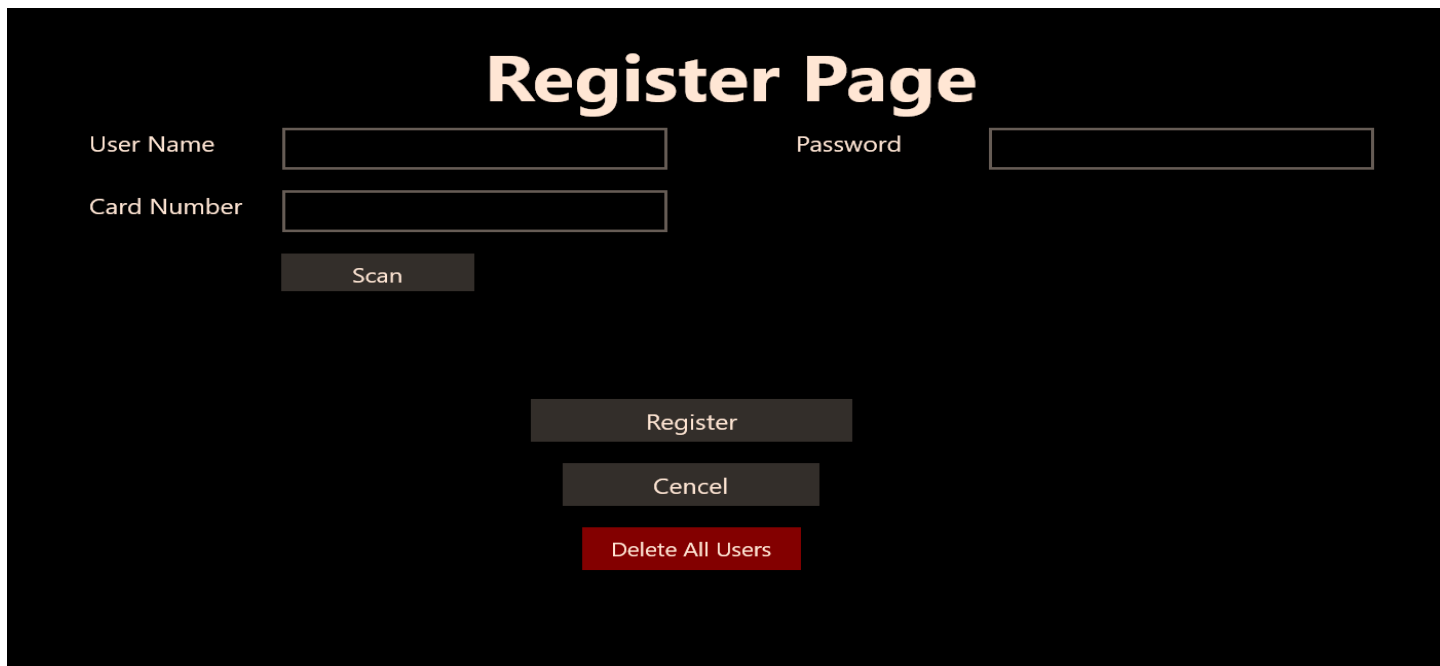
Бутона „Create Employee“ ни препраща към „Create“ страницата.

Тази страница ни предоставя възможността да редактираме всяко от полетата на даден служител.

Бутон „Edit“ ни потвърждава и запазва всички промени по даден служител

Бутон „Cencel“ ни отказва от всички въведени промени

Бутон „Delete“ изтрива служителя от всички таблици но преди да се извърши тази операция ни излиза потвърждаващо поле.

The image shows a web form titled "Register Page" in a large, bold, light blue font. Below the title, there are four input fields: "User Name" and "Password" on the top row, and "Card Number" and a "Scan" button on the bottom row. The "Scan" button is a small, dark grey rectangle with the word "Scan" in white. Below the input fields, there are three buttons stacked vertically: "Register" (dark grey), "Cencel" (dark grey), and "Delete All Users" (red with white text). The entire form is set against a solid black background.

Register Page

User Name

Password

Card Number

Scan

Register

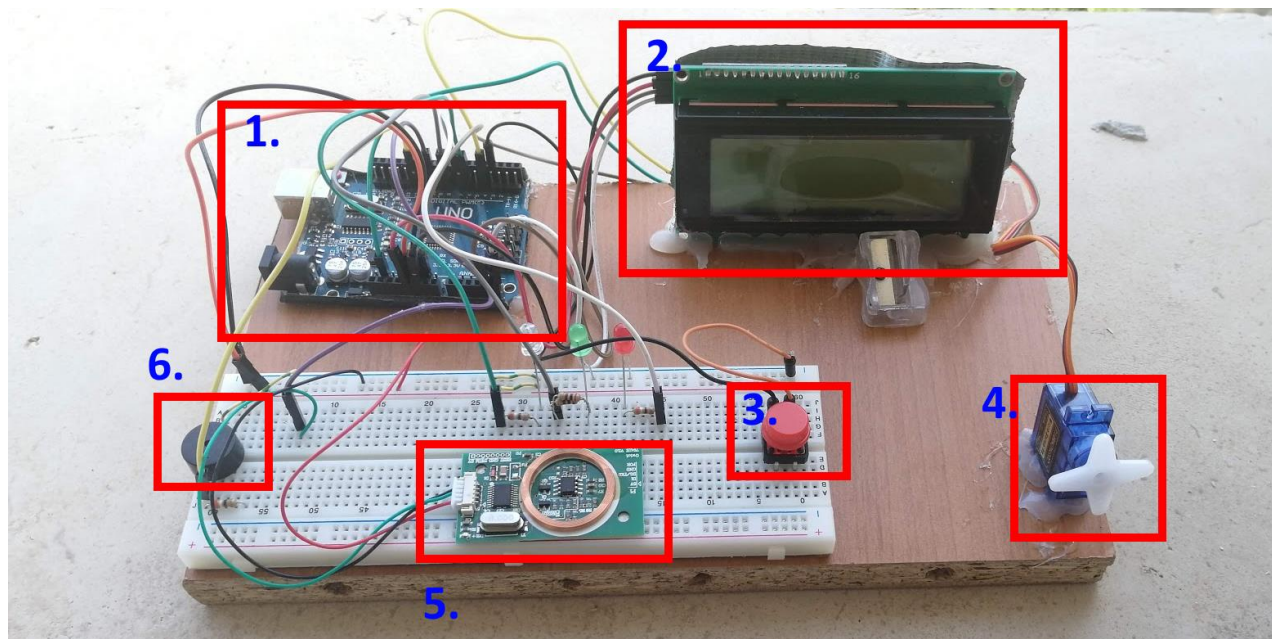
Cencel

Delete All Users

Тази страница ни предоставя възможността да създаваме админи които да имат достъп до данните на служителите.

За да създадем админ ние трябва да предоставим в полето „Card Number“ индефикационият номер на „Mastar Card“ която я предоставяме ние.

Имаме опцията да изтрием всички админи регистрирани до момента но преди това ще ни излезе потвърждаващо поле.



Съставни части:

1. ARDUINO UNO R3 (CH340G)

- Осъществява комуникацията между софтуер и хардуер.

2. Character Display LCD Modules 20x4

- Насочва потребителя
- Иписва името на потребителя който е сканирал



3. Бутон

- Ръчно отключва и заключва „ключалката“
- Без да има връзка към програмата

4. Micro continuous Servo TS90A

- Играе ролята на ключалка

5. RFID Reader

- Прочита номера на картата
- Праща идентификационния номер на програмата

6. Зумер

- Задейства се при отваряне и затваряне на ключалката



7. Ляв светодиоди

- Мига (в синя светлина)
- Индикира за осъществена комуникация между хардуер и софтуер

8. Среден светодиоди

- Светва (в зелена светлина) когато кода на сканираната карта е базата

9. Десен светодиоди


- Светва (в червена светлина) когато кода на сканираната карта не е в базата

Описание на функционалността

Отключване и заключване на ключалка

```
private void BackgroundWorker_Do_Work(object sender, DoWorkEventArgs e)
{
    //Connects to ARDUINO
    BackgroundWorker worker = sender as BackgroundWorker;
    mySerialPort = new SerialPort("COM4");
    mySerialPort.BaudRate = 9600;
    mySerialPort.Open();

    //Used to Power Yellow LED ON
    mySerialPort.Write("CommunicationON");
    while (true)
    {
        string cmd = string.Empty;
        //Used to Read all 10 Serial Sends Strings from READER
        for (int i = 1; i <= 10; i++)
        {
            cmd = cmd + $"{mySerialPort.ReadLine()}";
            cmd = cmd.Substring(0, cmd.Length - 1);
        }
    }
}
```



При стартирането на програмата се активира „BackgroundWorker“

1. Това е метод на „BackgroundWorker“ който се изпълнява през цялото време когато програмата е пусната.
2. Осъществява се връзката по сериален порт между програмата и Ардуиното.
3. Праща се команда по сериален порт на Ардуиното която влючва Светодиод който индикира комуникацията между Софтуер и Хардуер.
4. Цикълът чака идентификационния номер на сканираната карта от Четецът и го събита в една променлива.

```
while(RF.available()>0) {
    code =RF.read();
    //Sends the readed code from the card to Serial Port
    Serial.println(code);
    vleze = true;
}
```

В ардуиното когато някой доближи карта до четеца се задейства while цикълът и се прочита номерът на картата (разделен на десет части) и се праща на Serial port.

```
else if (cmd.Length > 3 && cmd.Substring(cmd.Length - 2) != "83")
{
    //Gets the employee with the SAME Code
    var CurEmp = this.EmpContext.Employees.FirstOrDefault(x => x.ScannerCardNumber == cmd);
    if (CurEmp == null)
    {
        // 0 - Sent to ARDUINO that there is no Employee with this Code
        mySerialPort.Write("0");
    }
    else
    {
        //Sends to ARDUINO the FirstName and LastName of the currently scanned Employee
        mySerialPort.Write($"{CurEmp.FirstName} {CurEmp.LastName}");
    }
}
```

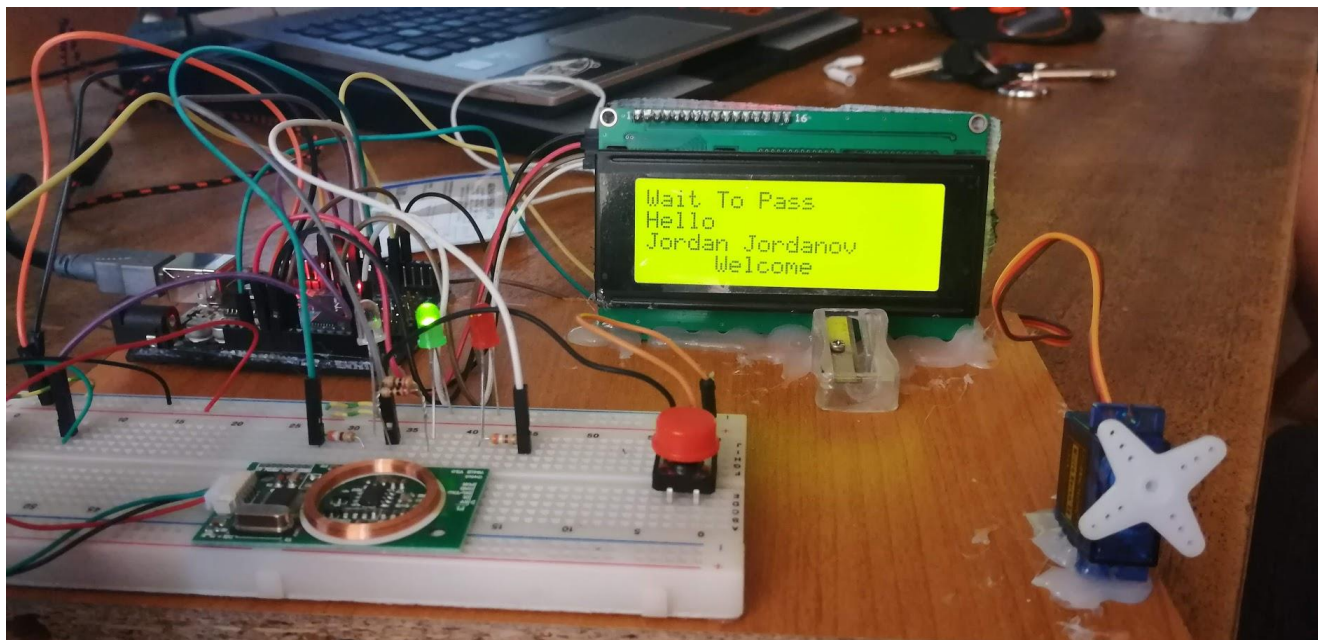
Взема се готовата променлива и се проверява в базата от данни дали има служител с такъв номер. Ако нама такъв ще върне „0“, а ако има ще му върне двете имена по сериален порт на Ардуиното.


```
brak = Serial.readString();
```

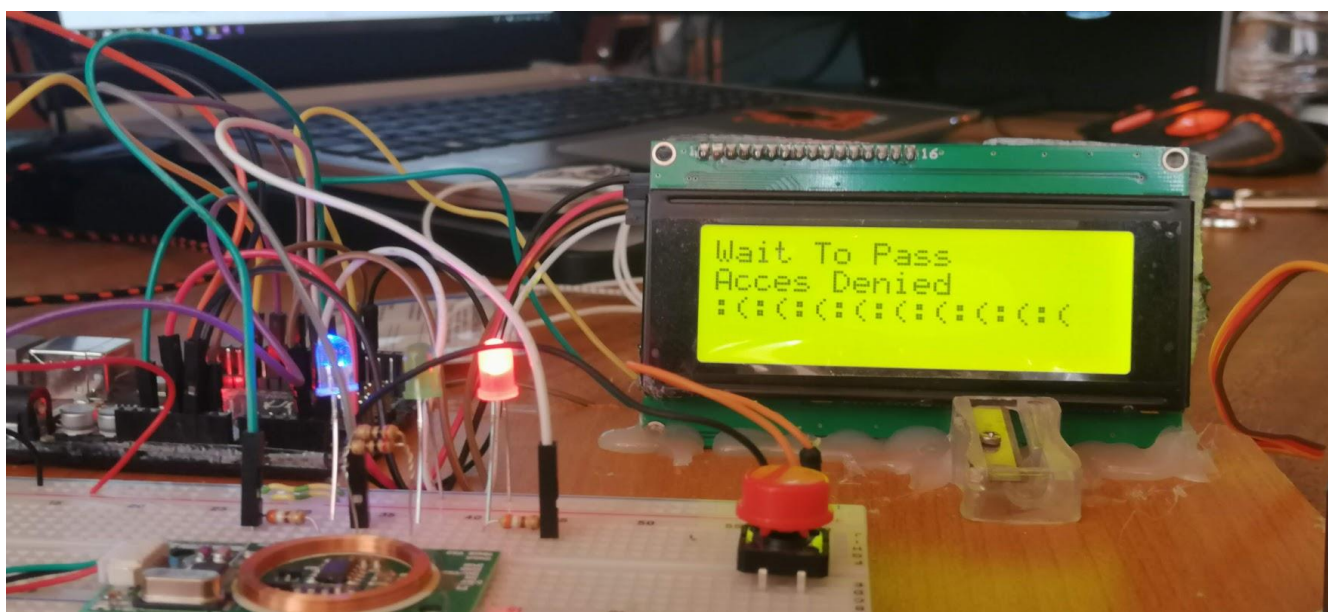
```
if (brak!="0"&&brak!="ScanedCardForRegistration")  
  LEDPassLCD (brak);  
  Pass();  
}  
if (brak=="0") {  
  NotPass();  
}
```

Първо Ардуиното прочита командата пратена по сериален порт от "Visual Studio" и прави нужните проверки

Достъпът приет



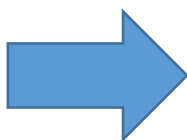
Достъпа отказан



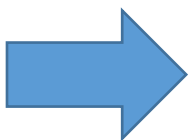
Номер на картата на служител

Card Number





```
private void BtnScan_Click(object sender, RoutedEventArgs e)
{
    App.StopWorker();
    this.BtnGet.Visibility = Visibility.Visible;
}
```



```
public static void StopWorker()
{
    //Say that sended Arduino CODE is Used for Registration
    PressedScanButton = false;
}
```

```
if (PressedScanButton == false)
{
    Lastscanned CardNumber = new Lastscanned()
    {
        ScannerCardNumber = $"{cmd}"
    };
    EmpContext.Lastscanned.Add(CardNumber);
    EmpContext.SaveChanges();
    //Add to Database The Scanned code
    //To be used when pressed "GET BTN"

    PressedScanButton = true;
    //Send to ARDUINO not to wait for Person
    mySerialPort.Write("ScannedCardForRegistration");
}
```

1

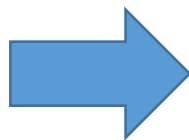
2

1. Праца в таблицата от базата „Last Scaned“ току що сканирания индефикацияния код на сканираната карта

2. Праца команда на Ардуиното (да бездейства 😊)

Card Number





```
private void BtnGet_Click(object sender, RoutedEventArgs e)
{
    this.CardNUmberTxtBox.Text = $"{GraphBLL.GetLast()}";
    this.BtnGet.Visibility = Visibility.Collapsed;
    App.StartWorker();
}
```

```
public string GetLast()
{
    var LastScannedEmployee = this.Database.Lastscanned.LastOrDefault();
    if (LastScannedEmployee == null)
    {
        return "";
    }
    this.Database.Lastscanned.Remove(LastScannedEmployee);
    this.Database.SaveChanges();
    return LastScannedEmployee.ScannerCardNumber;
}
```

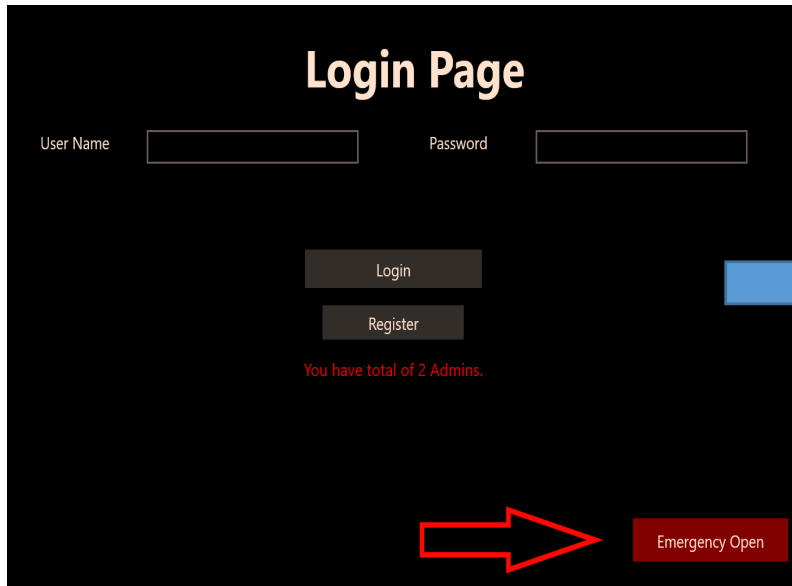
Взема от таблицата (ако потребителя е сканирал нещо) последния запис и го постявя в текстовот поле след това изтрива записа от таблицата.



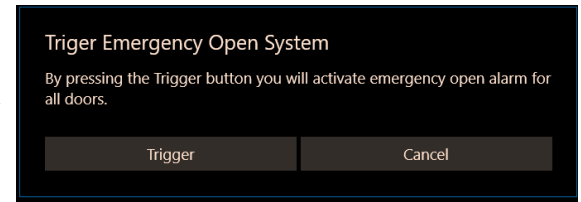
Card Number



Бутон за инзвънредни ситуации



Този бутон има задача да отключи ключалките при инзвънредни ситуации



```
// Otherwise, do nothing.
if (result == ContentDialogResult.Primary)
{
    this.EmergencyOpenTxtBox.Visibility = Visibility.Visible;
    App.TrigerEmergencyAlarm();
    this.TrigeredOnce = true;
    this.EmergencyOpenBut.Content = "Emergency Close";
}
else
{
    // The user clicked the CloseButton, pressed ESC, Gamepad B,
    // Do nothing.
}
```

При потвърждаване натискането на бутона „Emergancy Open“ ако го натискме за първи път се появява прозореца за потвърждаване ако не викаме метода „TrigerEmergencyAlarm“ на статичния клас „App“.

```
public static void TrigerEmergencyAlarm()
{
    mySerialPort.Write("EmergencyAlarmOn");
}
```

Този статичен метод праща по сериален порт на ардуино следната команда

```
if(Serial.available()){
    //Reads the input from the program
    SerialCommand = Serial.readString();
    //if programs is now started
    if(SerialCommand=="EmergencyAlarmOn" {
        LCDEmergencyOpen();
        ServoOpen();
        EmergencyLEDOn();
    }
}
```



Сорс код – Описание на функционалността

```
#include <SoftwareSerial.h>

#include <LiquidCrystal_PCF8574.h>

#include <Wire.h>

#include <Servo.h>

#include <Servo.h>

Servo ServoLock;

const int LedGreen = 8;

const int LedRed = 7;

const int LedYellow = 6;

const int Zoomer = 12;

const int buttonPin = 3;

bool buttonState ;

SoftwareSerial RF(10, 11); // RX, TX

int cardNumber;

bool wasInWhile = false;

String brak;

String SerialCommand;

LiquidCrystal_PCF8574 lcd(0x27); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup() {

  Serial.begin(9600);

  RF.begin(9600);

  ServoLock.attach(4);

  digitalWrite(Zoomer,LOW);

  pinMode(buttonPin, INPUT_PULLUP);

  InitializeLEDs();

  InitializeLCD();

}
```



```

void loop() {

    //Waits for input from the program
    if(Serial.available()){
        //Reads the input from the program
        SerialCommand = Serial.readString();
        //if prograServoLock is now started
        if(SerialCommand=="EmergencyAlarmOn"){
            //Displays on LCD "Emergency Open"
            LCDEmergencyOpen();
            ServoOpen();
            //Power onn both LED
            EmergencyLEDOOn();
            //When you cencel the Emergency
            if(SerialCommand=="EmergencyAlarmOff"){
                ServoClose();
                LcdClearWithoutDelay();
                EmergencyLEDOff();
            }

            if(SerialCommand=="CommunicationON"){
                //Turns on Communication LED
                digitalWrite(LedYellow,HIGH);
            }
            //if program is closing
            if(SerialCommand=="CommunicationOFF"){
                //Turns off Communication LED
                digitalWrite(LedYellow,LOW);
            }
        }
    }
}

```

```
//Reads the cardNumber from the RF Reader
while(RF.available(>0)){
    cardNumber =RF.read();
    //Sends the readed cardNumber from the card to Serial Port
    Serial.println(cardNumber);
    wasInWhile = true;
}

if(wasInWhile){
    wasInWhile = false;
    //Reads the input from the program
    brak = Serial.readString();
    //Checks for false Scanned text
    if(brak!="0"&&brak!="ScanedCardForRegistration"&&brak
!="CommunicationON"&&brak!="CommunicationOFF"&&brak!=""){
        LEDPassLCD(brak);
        Pass();
    }
    if(brak=="0"){
        NotPass();
    }
}

//Checks for pressed "Manual Opne" button
CheckBUT();
}
```

```

void Pass(){
    digitalWrite(Zoomer,HIGH);

    ServoOpen();

    LEDPass();

    digitalWrite(Zoomer,LOW);

    LcdClear();

    ServoClose();
}

void NotPass(){
    LCDNotPass();

    LEDNotPass();

    LcdClear();
}

void InitializeLCD(){
    Wire.begin();

    Wire.beginTransmission(0x27);

    lcd.begin(20, 4); // initialize the lcd

    lcd.setBacklight(255);

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Scanning...");
}

void InitializeLEDs(){
    pinMode(LedGreen,OUTPUT);

    pinMode(LedYellow,OUTPUT);

    pinMode(LedRed,OUTPUT);

    digitalWrite(LedYellow,LOW);
}

```

```
void LEDPass(){  
    digitalWrite(LedGreen,HIGH);  
    delay(2200);  
    digitalWrite(LedGreen,LOW);  
  
}  
  
void EmergencyLEDOn(){  
    digitalWrite(LedGreen,HIGH);  
    digitalWrite(LedRed,HIGH);  
  
}  
  
void EmergencyLEDOff(){  
    digitalWrite(LedGreen,LOW);  
    digitalWrite(LedRed,LOW);  
  
}  
  
void ServoOpen(){  
    ServoLock.write(50);  
    delay(653);  
    ServoLock.write(90);  
  
}  
  
void ServoClose(){  
    ServoLock.write(150);  
    delay(653);  
    ServoLock.write(90);  
  
}
```

```
void CheckBUT(){  
    //If button is pressed  
    if(digitalRead(buttonPin)==LOW){  
        //If button was pressed before  
        if (buttonState == LOW) {  
            digitalWrite(LedGreen,HIGH);  
            // digitalWrite(Zoomer,HIGH);  
            LCDOpenedByManual();  
            ServoOpen();  
  
        }  
        else {  
            // turn LED off:  
            digitalWrite(LedGreen,LOW);  
            // digitalWrite(Zoomer,LOW);  
            LcdClearWithoutDelay();  
            ServoClose();  
  
        }  
        //Change the button state to opasite  
        buttonState=!buttonState;  
        delay(200);  
    }  
}
```

```
void LEDNotPass(){  
  for(int a=0;a<2;a++)  
  {  
    digitalWrite(LedRed,HIGH);  
  
    digitalWrite(Zoomer,HIGH);  
    delay(1000);  
    digitalWrite(Zoomer,LOW);  
  
    digitalWrite(LedRed,LOW);  
  
    delay(1000);  
    digitalWrite(Zoomer,LOW);  
  }  
}
```

```
void LCDOpenedByManual(){  
  lcd.clear();  
  lcd.setCursor(0, 0);  
  lcd.print("Wait To Pass");  
  lcd.setCursor(0, 1);  
  lcd.print("Opened By Manual");  
}
```

```
void LCDEmergencyOpen(){  
  lcd.clear();  
  lcd.setCursor(0, 0);  
  lcd.print("Wait To Pass");
```

```
void LEDPassLCD(String EmpName){  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Wait To Pass");  
    lcd.setCursor(0, 1);  
    lcd.print("Hello");  
    lcd.setCursor(0, 2);  
    //Prints the name of scanned Employee  
    lcd.print(EmpName);  
    lcd.setCursor(5, 3);  
    lcd.print("Welcome");  
  
}  
  
void LCDNotPass(){  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Wait To Pass");  
    lcd.setCursor(0, 1);  
    lcd.print("Acces Denied");  
    lcd.setCursor(0, 2);  
    lcd.print(":(:(:(:(:(:(:(:(:('");  
  
}  
  
void LcdClearWithoutDelay(){  
    lcd.clear();  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Scanning...");  
  
}
```

```
void LcdClear(){  
    delay(3000);  
    lcd.clear();  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Scanning...");  
}
```


Заклучение

Положителни страни

Проектът позволява да се следи колко часа работи на ден един служител

Проектът позволява да се следи колко часа е работил за един месец

Проектът позволява лесно да се редактират данните на служителите

Улеснен и опростен дизайн на приложението

Проектът предоставя опция за извънредно отваряне на ключалките

Проектът предоставя опция за ръчно отваряне на ключалките без да изисква връзка със софтуер

Проектът ограничава достъпа на външни лица в дадената сграда

Приложението към проекта не може да се достъпи от всеки

Отрицателни страни

Липса на Bluetooth модул

Изискването на постоянна връзка Приложение и Хардуер

Липса на Уеб приложение

Contents

RFID – радиочестотна идентификация	1
Какво е RFID?	1
Анализи и прогнози	1
Бъдещи иновации за RFID	1
Електрическа схема.....	2
Блокова схема на програмното осигуряване за хардуера	3
Employee-Management	5
1. Тема	5
2. Автор:	5
3. Резюме:	5
4. Основните етапи в реализирането на проекта са разделени в няколко групи	6
5. Описание на Софтуерът	6
Описание на функционалността.....	11
Номер на картата на служител.....	13
Заключение.....	15
Положителни страни.....	24
Проектът позволява да се следи колко часа работи на ден един служител	24
Проектът позволява да се следи колко часа е работил за един месец.....	24
Проектът позволява лесно да се редактират данните на служителите	24
Улеснен и опростен дизайн на приложението.....	24
Проектът предоставя опция за инзвънредно отваряне на ключалките.....	24
Проектът предоставя опция за ръчно отваряне на ключалките без да изисква връзка със софтуер.....	24
Проектът ограничава достъпа на външни лица в дадената сграда	24
Приложението към проекта не може да се достъпи от всеки	24
Отрицателни страни.....	24
Липса на Bluetooth модул.....	24
Изискването на постоянна връзка Приложение и Хардуер	24
Липса на Уеб приложение	24
Бутон за инзвънредни ситуации	Error! Bookmark not defined.