**Dokumentácia k projektu**

**Dochádzkový systém s použitím ESP32, RFID čítačky a OLED displeja**

**Praktické aplikácie Internetu vecí 1**

**Vypracovala:**

Bc. Bernaťáková Janka

Bc. Brišová Markéta

**Úvod**

Tento projekt slúži na evidenciu dochádzky študentov pomocou RFID kariet, s napojením na webové rozhranie a notifikácie cez Telegram.  
Dáta sa ukladajú do databázy MySQL na serveri a sú zobrazované v modernom rozhraní.  
Systém umožňuje aj manuálne pridanie záznamov v prípade, že niekto zabudne kartu, export a download excel súboru, tlač dokumentu, vymazanie a úpravu jednotlivých položiek. Zároveň je tam pripojený aj aktívny buzzer, ktorý vydá zvuk pri každom priložení karty.

Obrázok, na ktorom je text, elektronika, elektronické zariadenie, snímka obrazovky

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

**Popis systému**

* ESP32 načíta RFID karty cez čítačku (MFRC522).
* Po priložení karty ESP32 pošle údaje cez HTTP POST na server (PHP script).
* Server údaje uloží do MySQL databázy (tabuľka RFID).
* Webové rozhranie (index.php) ich zobrazuje v tabuľke.
* Ak je karta priložená, zároveň sa odošle notifikácia na Telegram cez Telegram Bot API.

**Zloženie systému:**

**Obrázok, na ktorom je text, snímka obrazovky, dizajn

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.**

**PHP súbory:**

* index.php — hlavné rozhranie: tabuľka, export, tlač, filtrovanie, manuálne pridanie.
* insert\_manual.php — skript na pridanie manuálne zadaného záznamu.
* delete\_record.php — skript na vymazanie jedného záznamu.
* delete\_multiple.php — skript na hromadné vymazanie viacerých označených záznamov.
* get\_data.php — skript na získavanie dát z databázy.
* update\_record.php — skript na úpravu času prihlásenia.
* login.php, logout.php — jednoduchý prihlasovací systém.
* manifest.json
* service-worker.js

**CSS súbory:**

* styles.css — vlastné štýlovanie vzhľadu stránky (tabuľky, tlačidlá, pozadia, responsivita).

**Databáza:**

* Tabuľka RFID obsahuje:
  + id (primárny kľúč, AUTO\_INCREMENT)
  + meno (VARCHAR)
  + priezvisko (VARCHAR)
  + cardID (VARCHAR)
  + login (DATETIME)

**ESP32 program:**

* ESP32 načíta RFID kartu.
* Pošle údaje cez HTTP POST na PHP script insertRFID.php.

**Telegram Bot:**

* Vytvorený cez [BotFather](https://t.me/botfather) na Telegrame.
* PHP skript odošle správu pomocou https://api.telegram.org/bot<TOKEN>/sendMessage.
* Správa obsahuje: Meno, Priezvisko, ID karty a čas prihlásenia.

**Obal:**

* Obal zariadenia bol vytvorený v programe Fusion360 a následne vytlačený 3D tlačou.

Obrázok, na ktorom je elektronické zariadenie, smartfón, čierno-biela, elektronika

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.Obrázok, na ktorom je gadget, elektronické zariadenie, elektronika, smartfón

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

Postup vytvorenia systému je nasledovný. Najprv sa vytvorí databáza v MySQL a v nej sa vytvorí tabuľka s názvom RFID, ktorá bude obsahovať stĺpce id, meno, priezvisko, cardID a login. Následne sa ESP32 pripojí k Wi-Fi sieti cez router a po priložení RFID karty odošle údaje cez HTTP POST metódu na server. Webové rozhranie bude zabezpečené pomocou súboru index.php, ktorý bude načítavať údaje z databázy prostredníctvom skriptu get\_data.php. Pri vkladaní nového záznamu do databázy (napríklad cez skript insertRFID.php) sa zároveň zavolá API Telegramu, aby sa odoslali notifikácie o novom príchode.

**Kód na ESP32 (Arduino IDE)**

|  |
| --- |
| #include <SPI.h>  #include <MFRC522.h>  #include <LiquidCrystal\_I2C.h>  #include <WiFi.h>  #include <HTTPClient.h>  #include <NTPClient.h>  #include <WiFiUdp.h>  #include <time.h>  WiFiUDP ntpUDP;  NTPClient timeClient(ntpUDP, "pool.ntp.org", 3600, 60000); // 3600 = časový posun na Slovensku (UTC+1), 60000 = synchronizácia každú minútu  unsigned long lastDisplayTime = 0;  bool showingName = false;  #define SS\_PIN 5  #define RST\_PIN 22  #define BUZZER\_PIN 16  MFRC522 mfrc522(SS\_PIN, RST\_PIN);  LiquidCrystal\_I2C lcd(0x27, 16, 2);  // WiFi  const char\* ssid = "iPhone";  const char\* password = "123456789000";  // Telegram  String telegramToken = "7969960838:AAFCGwY-zIgRupyLKUA0GMcDmtewNKiCuTA";  String chatId = "5650127102";  #define MAX\_LOGS 10  String logs[MAX\_LOGS];  int logIndex = 0;  // Lokálny PHP server  const char\* serverUrl = "http://rabcan.home.sk/insertRFID.php";  // Lokálna "databáza" používateľov  struct User {    String cardID;    String meno;    String priezvisko;  };  User users[] = {    {"A3E0DF2C", "Bc.Marketa", "Brisova"},    {"3305FC2C", "Bc.Janka", "Bernatakova"},    {"D5C17130", "Liana", "Brisova"}  };  const int numUsers = sizeof(users) / sizeof(users[0]);  void setup() {    Serial.begin(115200);    SPI.begin();    mfrc522.PCD\_Init();    lcd.init();    lcd.backlight();    pinMode(BUZZER\_PIN, OUTPUT);    lcd.print("Skenujte kartu");    WiFi.begin(ssid, password);    while (WiFi.status() != WL\_CONNECTED) {      delay(1000);      Serial.println("Pripajam sa na WiFi...");    }    Serial.println("WiFi pripojene");    timeClient.begin();  }  void loop() {    timeClient.update();    if (!showingName) {      // Získaj aktuálny čas      time\_t rawTime = timeClient.getEpochTime();      struct tm \* timeInfo = localtime(&rawTime);      lcd.setCursor(0, 0);      lcd.print((timeInfo->tm\_mday < 10 ? "0" : "") + String(timeInfo->tm\_mday) + ".");      lcd.print((timeInfo->tm\_mon + 1 < 10 ? "0" : "") + String(timeInfo->tm\_mon + 1) + ".");      lcd.print(String(timeInfo->tm\_year + 1900));      lcd.setCursor(0, 1);      lcd.print(timeClient.getFormattedTime()); // zobrazí HH:MM:SS    } else {      if (millis() - lastDisplayTime > 5000) { // ak uplynulo 5 sekúnd        showingName = false;        lcd.clear();        // Hneď obnov dátum a čas        time\_t rawTime = timeClient.getEpochTime();        struct tm \* timeInfo = localtime(&rawTime);        lcd.setCursor(0, 0);        lcd.print((timeInfo->tm\_mday < 10 ? "0" : "") + String(timeInfo->tm\_mday) + ".");        lcd.print((timeInfo->tm\_mon + 1 < 10 ? "0" : "") + String(timeInfo->tm\_mon + 1) + ".");        lcd.print(String(timeInfo->tm\_year + 1900));        lcd.setCursor(0, 1);        lcd.print(timeClient.getFormattedTime());      }    }    if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial()) {      String cardID = "";      for (byte i = 0; i < mfrc522.uid.size; i++) {        cardID += String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");        cardID += String(mfrc522.uid.uidByte[i], HEX);      }      cardID.toUpperCase();      Serial.println("ID karty: " + cardID);      // Pípnutie buzzera      digitalWrite(BUZZER\_PIN, HIGH);      delay(100);      digitalWrite(BUZZER\_PIN, LOW);      String meno = "Nezname";      String priezvisko = "";      for (int i = 0; i < numUsers; i++) {        if (users[i].cardID.equalsIgnoreCase(cardID)) {          meno = users[i].meno;          priezvisko = users[i].priezvisko;          break;        }      }      lcd.clear();      lcd.setCursor(0, 0);      lcd.print(meno);      lcd.setCursor(0, 1);      lcd.print(priezvisko);      showingName = true;      lastDisplayTime = millis(); // zapamätaj si čas zobrazenia mena      Serial.println("Meno: " + meno);      Serial.println("Priezvisko: " + priezvisko);      sendDataToServer(cardID, meno, priezvisko);      // Zaznamenaj do zoznamu      String zaznam = meno + " " + priezvisko + " – " + cardID;      logs[logIndex] = zaznam;      logIndex = (logIndex + 1) % MAX\_LOGS;      // Odošli na Telegram      String sprava = "💳 " + meno + " " + priezvisko + " práve priložil/a kartu.";      sendTelegramMessage(sprava);      mfrc522.PICC\_HaltA();      mfrc522.PCD\_StopCrypto1();    }  }    void sendDataToServer(String cardID, String meno, String priezvisko) {    if (WiFi.status() == WL\_CONNECTED) {      HTTPClient http;      http.begin(serverUrl);      http.addHeader("Content-Type", "application/x-www-form-urlencoded");      String postData = "meno=" + meno + "&priezvisko=" + priezvisko + "&cardID=" + cardID;      Serial.println("POST data: " + postData);      int httpResponseCode = http.POST(postData);      if (httpResponseCode > 0) {        Serial.print("HTTP kód: ");        Serial.println(httpResponseCode);        String odpoved = http.getString();        Serial.println("Server odpovedal: " + odpoved);      } else {        Serial.print("HTTP chyba: ");        Serial.println(httpResponseCode);      }      http.end();    } else {      Serial.println("WiFi nie je pripojená");    }  }  void sendTelegramMessage(String message) {    if (WiFi.status() == WL\_CONNECTED) {      HTTPClient http;      String url = "https://api.telegram.org/bot" + telegramToken +                   "/sendMessage?chat\_id=" + chatId +                   "&text=" + urlencode(message);      http.begin(url);      int httpCode = http.GET();      if (httpCode > 0) {        Serial.println("✅ Telegram správa odoslaná");      } else {        Serial.println("❌ Chyba pri odosielaní na Telegram");      }      http.end();    }  }  void checkTelegramCommands() {    if (WiFi.status() == WL\_CONNECTED) {      HTTPClient http;      String url = "https://api.telegram.org/bot" + telegramToken + "/getUpdates";      http.begin(url);      int httpCode = http.GET();      if (httpCode == 200) {        String payload = http.getString();        if (payload.indexOf("/zoznam") >= 0) {          String zoznam = "📋 Posledné záznamy:\n";          for (int i = 0; i < MAX\_LOGS; i++) {            int idx = (logIndex + i) % MAX\_LOGS;            if (logs[idx] != "") {              zoznam += String(i + 1) + ". " + logs[idx] + "\n";            }          }          sendTelegramMessage(zoznam);        }      }      http.end();    }  }  // Pomocná funkcia na URL encoding (Telegram potrebuje)  String urlencode(String str) {    String encoded = "";    char c;    char code0, code1;    for (int i = 0; i < str.length(); i++) {      c = str.charAt(i);      if (isalnum(c)) {        encoded += c;      } else {        code1 = (c & 0xf) + '0';        if ((c & 0xf) > 9) code1 = (c & 0xf) - 10 + 'A';        code0 = ((c >> 4) & 0xf) + '0';        if (((c >> 4) & 0xf) > 9) code0 = ((c >> 4) & 0xf) - 10 + 'A';        encoded += '%';        encoded += code0;        encoded += code1;      }    }    return encoded;  } |

**Zdroje:**

<https://drive.google.com/drive/folders/1pVVohV-SlYDjMETumzXorOf9iBUJWQo9>

<https://randomnerdtutorials.com/esp32-mfrc522-rfid-reader-arduino/?utm_source>

<https://www.youtube.com/watch?app=desktop&v=tDQD2e2tLls&utm_source>

<https://www.youtube.com/watch?v=So83sH6-jwM>