Dokumentácia k „Digitálny Vrátnik“

Cieľ projektu:

- Cieľom tohto projektu je vytvoriť IoT riešenie, ktoré prepojí prístupové terminály založené na mikrokontroléri ESP32 s čítačkou RC522-RFID a zvukovým buzzerom, ktoré prostredníctvom bezdrôtovej siete (WiFi) pošlú údaje na vzdialený PHP server postavený na Raspberry Pi (OrangePi). Server bude zaznamenávať príchody a odchody zamestnancov, cez ktorý terminál sa prihlásili a celé to zobrazí na webovom rozhraní.

Súčiastky:

- OrangePi Zero LTS (Dual-Core, 512MB RAM)
- ESP32 LilyGO TTGO T-Display 1,14“
- RC522-RFID
- RFID čipové karty
- Buzzer
- F-F a M-F GPIO káble
- Breadboard
- Krabička pre terminál

OrangePi Zero LTS:

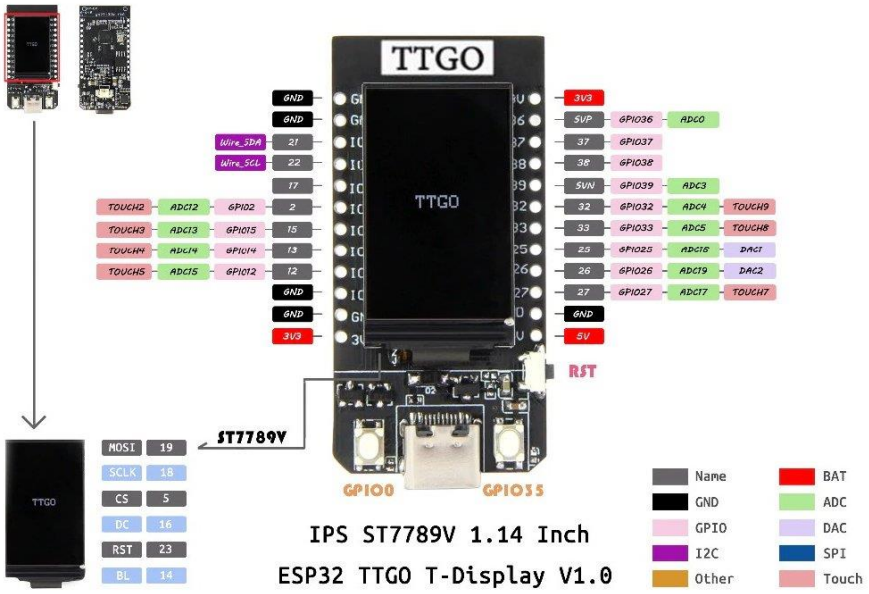
- Procesor: H3 Quad-core Cortex-A7
- Pamäť: 512MB DDR3 SDRAM
- FLASH: SD karta 16GB
- Pripojenie: 100Mbit Ethernet, WiFi IEEE 802.11 b/g/n

Obr. 1 – OrangePi Zero LTS

LilyGO TTGO T-Display 1,14“ ESP32:

- Typ: ESPRESSIF-ESP32 240MHz Xtensa®
- Pracovné napätie: 2.7V – 4.2V
- Procesor: Dual Core Tensilica LX6 240 MHz s výkonom 600 DMIPS

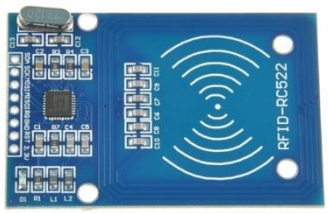
- SRAM: 520 kb
- FLASH: QSPI 4 MB
- WiFi: 802.11 B/G/N/E/I
- Bluetooth: 4.2 EBR s Legacy módom



Obr. 2 – PINOUT LilyGo TTGO ESP32

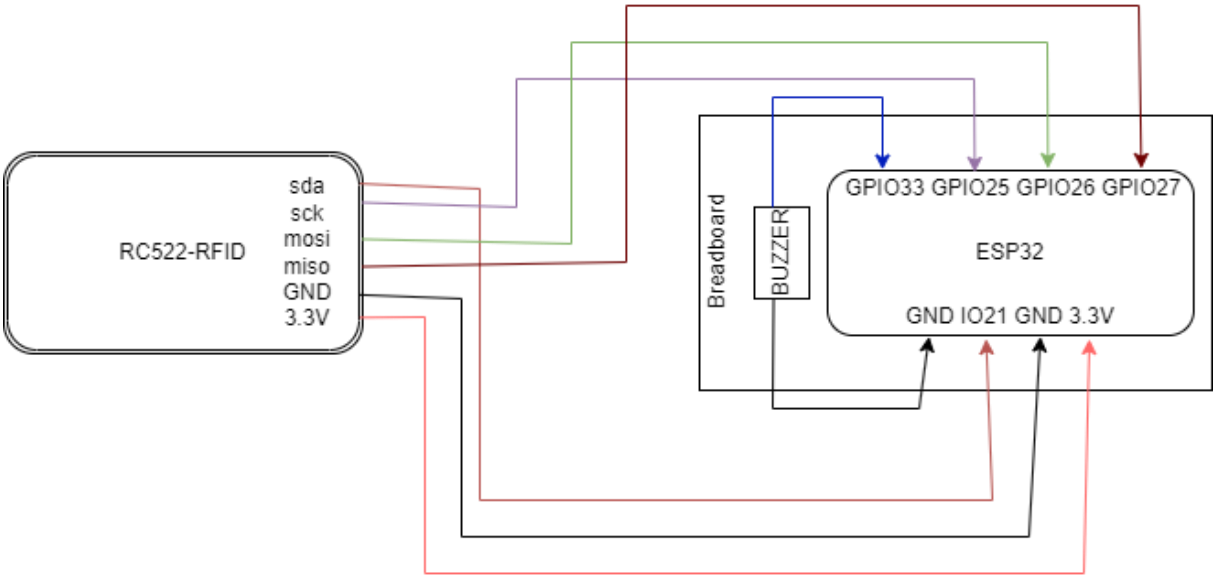
RC522-RFID :

- Operačný prúd :13-26mA/DC 3.3V
- Prúd počas nečinnosti :10-13mA/DC 3.3V
- Prúd pri spaní: <80uA
- Maximálny prúd: <30mA
- Rýchlosť prenosu: maximum 10Mbit/s
- Operačná frekvencia: 13.56MHz
- Podporované typy kariet: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire



Obr. 3 – RC522-RFID

Schéma zapojenia terminálu:



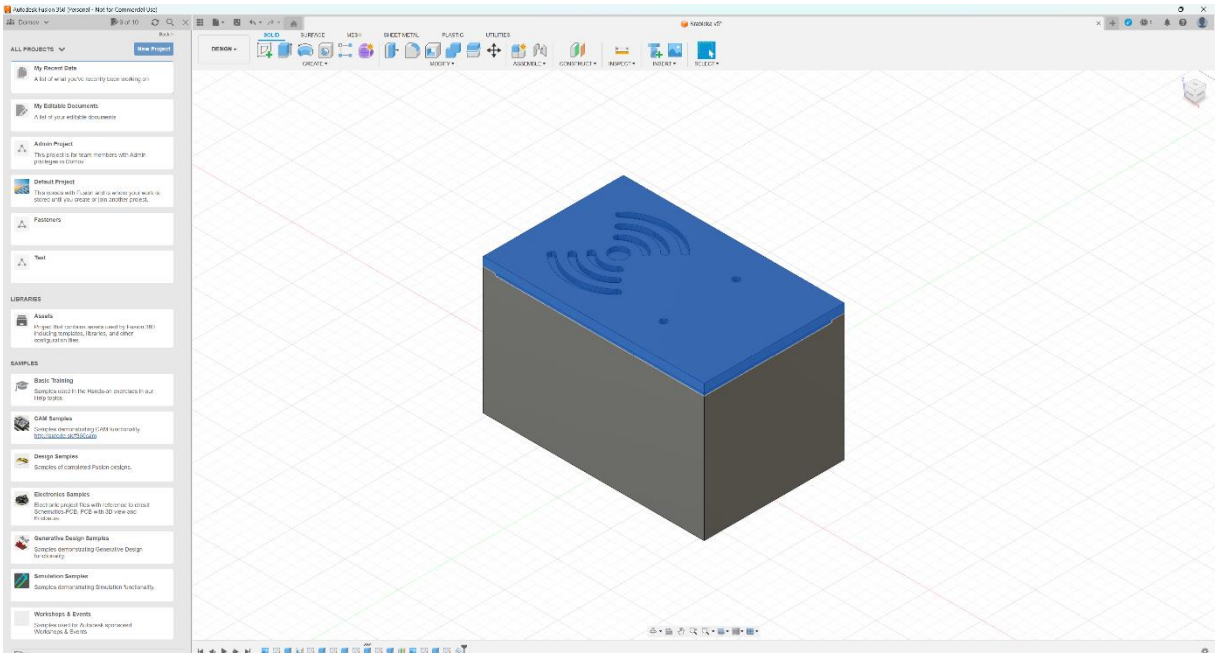
Obr. 4 – schéma zapojenia

Terminál:

Obr. 5 – fotka výsledného terminálu

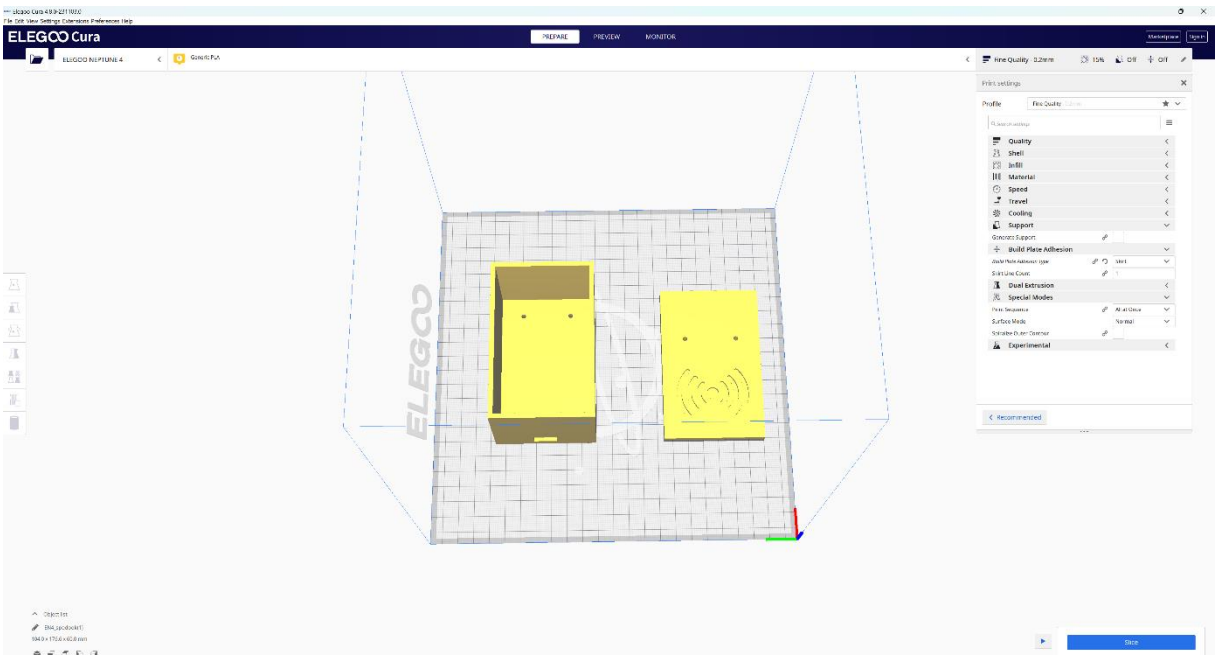
Krabička pre terminál:

- Namodelovaná v aplikácií Fusion 360



Obr. 6 – Krabička v aplikácií Fusion 360

- Vytlačená na 3D tlačiarni Elegoo Neptune 4



Obr. 7 – Pripravený 3D projekt pre tlač

Kódy:

- 1. MicroPython kód pre terminál (ESP32 + RC522-RFID)
- 2. PHP kód a webové rozhranie pre server (OrangePi Zero LTS)

1. MicroPython kód pre terminál (ESP32 + RC522-RFID)

1.1 „mfrc522.py“

```
from machine import Pin, SPI
from os import uname
class MFRC522:

    OK = 0
    NOTAGERR = 1
    ERR = 2

    REQIDL = 0x26
    REQALL = 0x52
    AUTHENT1A = 0x60
    AUTHENT1B = 0x61

    def __init__(self, spi, cs):

        self.spi = spi
        self.cs = cs
        self.cs.value(1)
        self.spi.init()
        self.init()

    def _wreg(self, reg, val):

        self.cs.value(0)
        self.spi.write(b'%c' % int(0xff & ((reg << 1) & 0x7e)))
        self.spi.write(b'%c' % int(0xff & val))
        self.cs.value(1)

    def _rreg(self, reg):

        self.cs.value(0)
        self.spi.write(b'%c' % int(0xff & (((reg << 1) & 0x7e) | 0x80)))
        val = self.spi.read(1)
        self.cs.value(1)

        return val[0]

    def _sflags(self, reg, mask):
        self._wreg(reg, self._rreg(reg) | mask)

    def _cflags(self, reg, mask):
        self._wreg(reg, self._rreg(reg) & (~mask))

    def _tocard(self, cmd, send):

        recv = []
```

```
        bits = irq_en = wait_irq = n = 0
        stat = self.ERR

        if cmd == 0x0E:
                irq_en = 0x12
                wait_irq = 0x10
        elif cmd == 0x0C:
                irq_en = 0x77
                wait_irq = 0x30

        self._wreg(0x02, irq_en | 0x80)
        self._cflags(0x04, 0x80)
        self._sflags(0x0A, 0x80)
        self._wreg(0x01, 0x00)

        for c in send:
                self._wreg(0x09, c)
        self._wreg(0x01, cmd)

        if cmd == 0x0C:
                self._sflags(0x0D, 0x80)

        i = 2000
        while True:
                n = self._rreg(0x04)
                i -= 1
                if ~((i != 0) and ~(n & 0x01) and ~(n & wait_irq)):
                        break

        self._cflags(0x0D, 0x80)

        if i:
                if (self._rreg(0x06) & 0x1B) == 0x00:
                        stat = self.OK

                        if n & irq_en & 0x01:
                                stat = self.NOTAGERR
                        elif cmd == 0x0C:
                                n = self._rreg(0x0A)
                                lbits = self._rreg(0x0C) & 0x07
                                if lbits != 0:
                                        bits = (n - 1) * 8 + lbits
                                else:
                                        bits = n * 8

                                if n == 0:
                                        n = 1
                                elif n > 16:
                                        n = 16

                                for _ in range(n):
                                        recv.append(self._rreg(0x09))
                else:
                        stat = self.ERR

        return stat, recv, bits

def _crc(self, data):
```

```
        self._cflags(0x05, 0x04)
        self._sflags(0x0A, 0x80)

        for c in data:
                self._wreg(0x09, c)

        self._wreg(0x01, 0x03)

        i = 0xFF
        while True:
                n = self._rreg(0x05)
                i -= 1
                if not ((i != 0) and not (n & 0x04)):
                        break

        return [self._rreg(0x22), self._rreg(0x21)]

def init(self):

        self.reset()
        self._wreg(0x2A, 0x8D)
        self._wreg(0x2B, 0x3E)
        self._wreg(0x2D, 30)
        self._wreg(0x2C, 0)
        self._wreg(0x15, 0x40)
        self._wreg(0x11, 0x3D)
        self.antenna_on()
def reset(self):
        self._wreg(0x01, 0x0F)

def antenna_on(self, on=True):

        if on and ~(self._rreg(0x14) & 0x03):
                self._sflags(0x14, 0x03)
        else:
                self._cflags(0x14, 0x03)

def request(self, mode):

        self._wreg(0x0D, 0x07)
        (stat, recv, bits) = self._tocard(0x0C, [mode])

        if (stat != self.OK) | (bits != 0x10):
                stat = self.ERR

        return stat, bits

def anticoll(self):

        ser_chk = 0
        ser = [0x93, 0x20]

        self._wreg(0x0D, 0x00)
        (stat, recv, bits) = self._tocard(0x0C, ser)

        if stat == self.OK:
                if len(recv) == 5:
                        for i in range(4):
                                ser_chk = ser_chk ^ recv[i]
```

```
                        if ser_chk != recv[4]:
                                stat = self.ERR
                else:
                        stat = self.ERR

        return stat, recv

    def select_tag(self, ser):

        buf = [0x93, 0x70] + ser[:5]
        buf += self._crc(buf)
        (stat, recv, bits) = self._tocard(0x0C, buf)
        return self.OK if (stat == self.OK) and (bits == 0x18) else self.ERR

    def auth(self, mode, addr, sect, ser):
        return self._tocard(0x0E, [mode, addr] + sect + ser[:4])[0]

    def stop_crypto1(self):
        self._cflags(0x08, 0x08)

    def read(self, addr):

        data = [0x30, addr]
        data += self._crc(data)
        (stat, recv, _) = self._tocard(0x0C, data)
        return recv if stat == self.OK else None

    def write(self, addr, data):

        buf = [0xA0, addr]
        buf += self._crc(buf)
        (stat, recv, bits) = self._tocard(0x0C, buf)

        if not (stat == self.OK) or not (bits == 4) or not ((recv[0] & 0x0F) == 0x0A):
            stat = self.ERR
        else:
            buf = []
            for i in range(16):
                buf.append(data[i])
            buf += self._crc(buf)
            (stat, recv, bits) = self._tocard(0x0C, buf)
            if not (stat == self.OK) or not (bits == 4) or not ((recv[0] & 0x0F) == 0x0A):
                stat = self.ERR

        return stat
```

1.2 „main.py“

```
from time import sleep_ms
from machine import Pin, SPI, PWM
from mfrc522 import MFRC522
import time
import urequests

sck = Pin(25, Pin.OUT)
mosi = Pin(26, Pin.OUT)
miso = Pin(27, Pin.OUT)
spi = SPI(baudrate=100000, polarity=0, phase=0, sck=sck, mosi=mosi, miso=miso)
```

```
sda = Pin(2, Pin.OUT)

adresa = "http://192.168.1.158:8000"

def do_connect():
    import network
    sta_if = network.WLAN(network.STA_IF)
    if not sta_if.isconnected():
        print('connecting to network...')
        sta_if.active(True)
        sta_if.connect('ASUS_CC', '*********') # Nakonfigurovane pripojenie na pripraveny Router
        while not sta_if.isconnected():
            pass
    print('network config:', sta_if.ifconfig())
    response = urequests.get(adresa)
    print(type(response))

def do_read():
    try:
        while True:
            rdr = MFRC522(spi, sda)
            uid = ""
            (stat, tag_type) = rdr.request(rdr.REQIDL)
            if stat == rdr.OK:
                (stat, raw_uid) = rdr.anticoll()
                if stat == rdr.OK:
                    uid = ("0x%02x%02x%02x%02x" % (raw_uid[0], raw_uid[1], raw_uid[2], raw_uid[3]))
                    print(uid)
                    response = urequests.get(adresa+"/"+uid)
                    print(type(response))
                    beeper = PWM(Pin(33, Pin.OUT), freq=400, duty=512)
                    sleep_ms(1000)
                    beeper.deinit()
    except KeyboardInterrupt:
        print("Bye")

do_connect()
do_read()
```

1.3 „read.py“

```
sck = Pin(25, Pin.OUT)
mosi = Pin(26, Pin.OUT)
miso = Pin(27, Pin.OUT)
spi = SPI(baudrate=100000, polarity=0, phase=0, sck=sck, mosi=mosi, miso=miso)

sda = Pin(2, Pin.OUT)

def do_read():
    try:
        while True:
            rdr = MFRC522(spi, sda)
            uid = ""
            (stat, tag_type) = rdr.request(rdr.REQIDL)
            if stat == rdr.OK:
```

```
            (stat, raw_uid) = rdr.anticoll()
            if stat == rdr.OK:
               uid = ("0x%02x%02x%02x%02x" % (raw_uid[0], raw_uid[1], raw_uid[2], raw_uid[3]))
               print(uid)
               sleep_ms(100)
   except KeyboardInterrupt:
      print("Bye")
```

2. Kód pre PHP server a webové rozhranie (OrangePi Zero LTS)

2.0.1 „boot.py“

```
# This file is executed on every boot (including wake-boot from deepsleep)
#import esp
#esp.osdebug(None)
#import webrepl
#webrepl.start()
```

2.1 „proxy.py“ - Python Server Example

```
from http.server import BaseHTTPRequestHandler, HTTPServer
import time
from datetime import datetime
import mysql.connector

hostName = "0.0.0.0"
serverPort = 8000
sql = "INSERT INTO `prichody` (`id`, `zamestnanec_id`, `terminal_id`, `cas`) VALUES (NULL, %, %, NOW);"
val = ("John", "Highway 21")
class MyServer(BaseHTTPRequestHandler):
   def do_GET(self):
      self.send_response(200)
      self.send_header("Content-type", "text/html")
      self.end_headers()
      mydb = mysql.connector.connect(
         host="localhost",
         user="Dochadzka",
         password="**********“,
         database=“dochadzka“
      )
      mycursor = mydb.cursor()
      terminalId = 1
      mycursor.execute(„SELECT id FROM `zamestnanci` WHERE `karta_id` LIKE ‚“+self.path[1:]+“‘ „)
      result = mycursor.fetchone()
      now = datetime.now()
      current_time = now.strftime(„%H:%M:%S“)
      if(result != None):
         userId = result[0]
         mycursor.execute(„SELECT * FROM `prichody` WHERE `zamestnanec_id` = „ + str(userId) + „ AND
DATE(cas) = CURDATE()“)
         result = mycursor.fetchone()
         if(result == None):
            print(userId)
            print(terminalId)
```

```
            result = mycursor.execute(„INSERT INTO `prichody` (`id`, `zamestnanec_id`, `terminal_id`, `cas`)
VALUES (NULL, %s, %s, NOW());“, (userId, terminalId))
            print(„Prichod zamestnanca s ID „ + str(userId) + „ cez terminal s id „ + str(terminalId) + „ o case „ +
current_time)
            mydb.commit()
        else:
            mycursor.execute(
                „SELECT * FROM `odchody` WHERE `zamestnanec_id` = „ + str(userId) + „ AND DATE(cas) =
CURDATE()“)
            result = mycursor.fetchone()
            if(result == None):
                result = mycursor.execute(„INSERT INTO `odchody` (`id`, `zamestnanec_id`, `terminal_id`, `cas`)
VALUES (NULL, %s, %s, NOW());“, (userId, terminalId))
                print(
                    „Odchod zamestnanca s ID „ + str(userId) + „ cez terminal s id „ + str(terminalId) + „ o case „ +
current_time)
                mydb.commit()
    mydb.disconnect()

if __name__ == „__main__“:
    webServer = HTTPServer((hostName, serverPort), MyServer)
    print(„Server started http://%s:%s“ % (hostName, serverPort))


    try:
        webServer.serve_forever()
    except KeyboardInterrupt:
        pass

    webServer.server_close()
    print(„Server stopped.“)
```

2.2 Webové rozhranie (HTML, PHP)

2.2.1 „index.php“

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <script src="js/bootstrap.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Zoznam zamestnancov</title>
</head>
<body>
<div class="container">
    <div class="row">
        <h1 class="text-center mt-2 mb-5">
            Zoznam zamestnancov
        </h1>
    </div>
    <div class="row">
        <div class="col">
            <table class="table">
            <thead>
                <tr>
                <th scope="col">Meno</th>
                <th scope="col">Pracovna doba</th>
```

```
            <th scope="col">Id Karty</th>
            <th scope="col"></th>
            </tr>
          </thead>
<?php
$servername = "localhost";
$username = "Dochadzka";
$password = "***********";
$dbname = "dochadzka";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

// Name query
$sql = "SELECT * FROM `zamestnanci`" ;

$result = $conn->query($sql);

if ($result->num_rows > 0) {
   // Output data of each row
   while($row = $result->fetch_assoc()) {
      echo "<tbody>";
      echo "<tr>";
      echo "<td>".$row["meno"]."</td>";
      echo "<td>".$row["doba"]."</td>";
      echo "<td>".$row["karta_id"]."</td>";
      echo "<td><a href=\"userattendance.php?id=".$row["id"]."\" class=\"text-center\"><button type=\"button\"
class=\"btn btn-primary\">Dochadzka</button></a></td>";
      echo "</tr>";
      echo "</tbody>";
   }
} else {
   echo "0 results";
}

// Close connection
$conn->close();
?>
         </table>
      </div>
   </div>
</div>
</body>
</html>
```

2.2.2 „userattendance.php“

```
<?php
   $id = $_GET["id"];
   $name = "";
   $servername = "localhost";
   $username = "Dochadzka";
   $password = "***********";
   $dbname = "dochadzka";

   // Create connection
   $conn = new mysqli($servername, $username, $password, $dbname);
```

```
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // Name query
    $sql1 = "SELECT * FROM `prichody` WHERE zamestnanec_id=" . $id;
    $result1 = $conn->query($sql1);

    // Name query
    $sql2 = "SELECT * FROM `odchody` WHERE zamestnanec_id=" . $id;
    $result2 = $conn->query($sql2);

    $rows1 = 0;
    $rows2 = 0;

    if ($result1->num_rows > 0) {
        // Fetch results into an array
        $rows1 = $result1->fetch_all(MYSQLI_ASSOC);

    }

    if ($result2->num_rows > 0) {
        // Fetch results into an array
        $rows2 = $result2->fetch_all(MYSQLI_ASSOC);
    }

    $sql = "SELECT * FROM `zamestnanci` WHERE id=".id ;

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // Output data of each row
        while($row = $result->fetch_assoc()) {
            $name = $row["meno"];
        }
    }
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <script src="js/bootstrap.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dochadzka zamestnanca</title>
</head>
<body>
<div class="container">
    <div class="row">
        <h1 class="text-center mt-2 mb-5">
            Dochadzka zamestnanca <?php echo $name;?>
        </h1>
    </div>
    <div class="row">
        <div class="col">
            <table class="table">
            <thead>
                <tr>
                <th scope="col">Den</th>
                <th scope="col">Prichod</th>
                <th scope="col">Odchod</th>
```

```
            </tr>
          </thead>
<?php

            if ($result1->num_rows > 0) {
               $a = 1;
               foreach($rows1 as $rowin) {
                  $arrival = new DateTime($rowin["cas"]);
                  if ($result2->num_rows > 0) {
                     foreach($rows2 as $rowout) {
                        $departure = new DateTime($rowout["cas"]);

                        if($arrival->format('Y-m-d') === $departure->format('Y-m-d'))
                        {
                           echo "<tbody>";
                           echo "<tr>";
                           echo "<td>".$arrival->format('Y-m-d')."</td>";
                           echo "<td>".$arrival->format('H:i')."</td>";
                           echo "<td>".$departure->format('H:i')."</td>";
                           echo "</tr>";
                           echo "</tbody>";
                           continue 2;
                        }
                     }
                     echo "<tbody>";
                     echo "<tr>";
                     echo "<td>".$arrival->format('Y-m-d')."</td>";
                     echo "<td>".$arrival->format('H:i')."</td>";
                     echo "<td>Zamestnanec sa neodhlasil</td>";
                     echo "</tr>";
                     echo "</tbody>";
                  }
               }
            }else {
               echo "No data";
            }

            // Close connection
            $conn->close();
         ?>
         </table>
      </div>
   </div>
   <div class="row">
   <a href="index.php" class="text-center"><button type="button" class="btn btn-primary">Zoznam</button></a>
   </div>
</div>
</body>
</html>
```
