Óbudai Egyetem



RFID óra időmérés Labview alkalmazássávaL

projektum  
Intelligens Rendszerek tárgyból

témavezető: Dr.Simon János hallgató: Kovács Árpád   
 főiskolai tanár Neptun kód: BPJZ56

Szabadka, 2020

Tartalom

[Bevezető 3](#_Toc57456653)

[1. Projektfeladat 3](#_Toc57456654)

[2. Elméleti alapok 3](#_Toc57456655)

[MySQL adatbázis:. 8](#_Toc57456656)

[3. A LabVIEW szoftver bemutatása 9](#_Toc57456657)

[A felhasznált rövidítések 16](#_Toc57456658)

[Irodalom 16](#_Toc57456659)

# Bevezető

A dokumentum az intelligens rendszerek tárgy dokumentálásának céljából jött létre.

# Projektfeladat

Az én projekt feladatom az volt, hogy egy adatbázis segítségével mérjem az időt az rfid kártyák lecsippantásával. Ehhez a feladathoz ESP32 mikrovezérlőt használtam fel.

# Elméleti alapok

Labview Program:

A labview az NI által fejlesztett ún. Virtuális műszerek létrehozására, és grafikus programozásra használt szoftver.

ESP32 egy 32 Mb os flash kapacitású Wifi-vel és Bluetooth-al rendelkező MCU.

Kisméretű modulok családja, amelyek ESP32 chipet és számos több kulcsfontosságú komponenst tartalmaznak, beleértve a kristály oszcillátort és MIFA antennát. Ez megkönnyíti az ESP32 alapú eszközök használatát, amely így készen áll a végtermékekbe való beszerelésre. Ezek a modulok fejlesztési platformként is használhatók, mivel számos egyéb összetevőt is tartalmaznak, például programozási felületet, a passzív alkatrészeket és tüskesort.



Figure 1 ESP32

RFID olvasó:

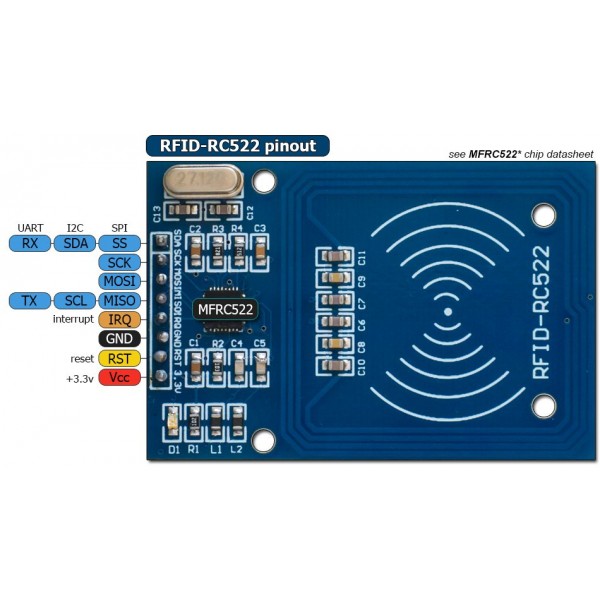
Az RFID (Radio Frequency IDentification) automatikus azonosításhoz és adatközléshez használt technológia, melynek lényege adatok tárolása és továbbítása RFID címkék és eszközök segítségével. Az RFID-címke egy apró tárgy, amely rögzíthető vagy beépíthető az azonosítani kívánt objektumba. Az objektum lehet tárgy, például egy árucikk, vagy alkatrész, illetve élőlény, így akár ember is. 

Figure 2 RFID olvasó

RFID rendszerek működése

1. A szenzor aktiválást indító eseményt érzékel

2. Az olvasó bekapcsol

3. Az olvasó leolvassa a bélyegben kódolt adatokat

4. Az olvasó továbbküldi az adatokat az adatfeldolgozó programot futtató számítógépnek (middleware)

5. Az adatfeldolgozó számítógép a kiegészített adatokat továbbküldi a vállalatirányítási rendszernek

Az általam felhasznált kártya olvasó is ezt az elvet követi.

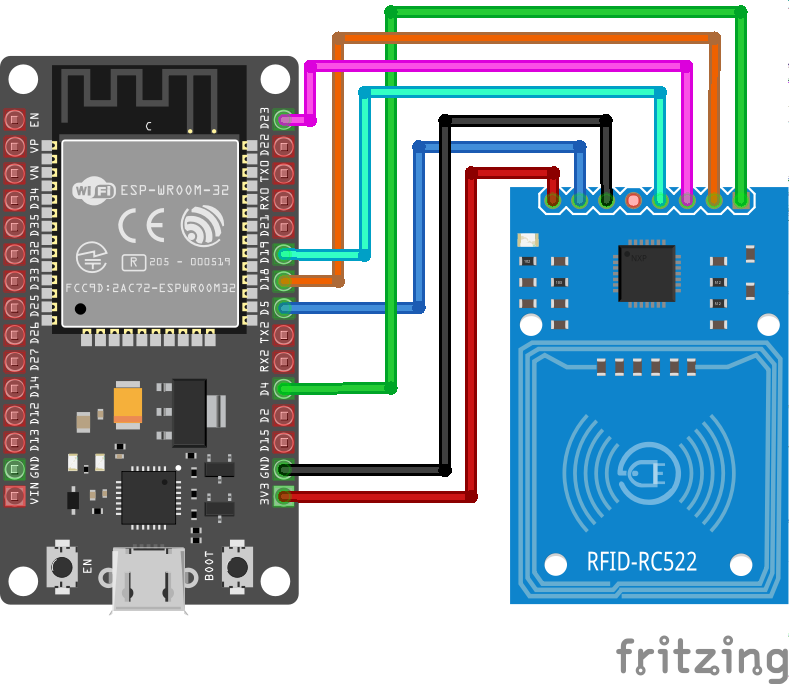


Figure 3 ESP32 és RFID olvasó

A felhasznált arduino program:.

|  |
| --- |
| #include <Arduino.h> #include <WiFi.h> #include <WiFiMulti.h> #include <MFRC522.h> #include <HTTPClient.h> #include <Arduino\_JSON.h>  #define USE\_SERIAL Serial #define SS\_PIN 21 #define RST\_PIN 22  WiFiMulti wifiMulti; JSONVar my\_elements = JSON.parse("{\"RFID-code\" : \"valami hexa\"}"); MFRC522 rfid(SS\_PIN, RST\_PIN); // Instance of the class MFRC522::MIFARE\_Key key; byte nuidPICC[4]; String s\_rfid="";   void setup() {  USE\_SERIAL.begin(9600);  USE\_SERIAL.println();  USE\_SERIAL.println();  USE\_SERIAL.println();  for(uint8\_t t = 4; t > 0; t--) {  USE\_SERIAL.printf("[SETUP] WAIT %d...\n", t);  USE\_SERIAL.flush();  delay(1000);  }  wifiMulti.addAP("TP-Link", "asdfghjkl123#");  SPI.begin(); // Init SPI   rfid.PCD\_Init(); // Init MFRC522  for (byte i = 0; i < 6; i++) {  key.keyByte[i] = 0xFF;  } } void loop() {  if ( ! rfid.PICC\_IsNewCardPresent()) return;  if ( ! rfid.PICC\_ReadCardSerial()) return;   Serial.print(F("PICC type: "));  MFRC522::PICC\_Type piccType = rfid.PICC\_GetType(rfid.uid.sak);  Serial.println(rfid.PICC\_GetTypeName(piccType));  if (piccType != MFRC522::PICC\_TYPE\_MIFARE\_MINI &&  piccType != MFRC522::PICC\_TYPE\_MIFARE\_1K &&  piccType != MFRC522::PICC\_TYPE\_MIFARE\_4K) {  Serial.println(F("Your tag is not of type MIFARE Classic."));  return;  }  for (byte i = 0; i < 4; i++) nuidPICC[i] = rfid.uid.uidByte[i];    Serial.println(F("The NUID tag is:"));  Serial.print(F("In hex: "));  s\_rfid="";  for(byte i=0;i<rfid.uid.size;i++)  {  s\_rfid+=String(rfid.uid.uidByte[i],HEX);  }  Serial.println(s\_rfid);  //printHex(rfid.uid.uidByte, rfid.uid.size);  Serial.println();  //setClock();  // wait for WiFi connection  if((wifiMulti.run() == WL\_CONNECTED)) {   HTTPClient http;   USE\_SERIAL.print("[HTTP] begin...\n");  // configure traged server and url  //http.begin("https://www.howsmyssl.com/a/check", ca); //HTTPS  http.begin("http://46.40.46.94:81/timers.php?json="+s\_rfid); //HTTP   USE\_SERIAL.print("[HTTP] GET...\n");  // start connection and send HTTP header  int httpCode = http.GET();   // httpCode will be negative on error  if(httpCode > 0) {  // HTTP header has been send and Server response header has been handled  USE\_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);   // file found at server  if(httpCode == HTTP\_CODE\_OK) {  String payload = http.getString();  USE\_SERIAL.println(payload);  }  } else {  USE\_SERIAL.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c\_str());  }   http.end();  }   // Halt PICC  rfid.PICC\_HaltA();   // Stop encryption on PCD  rfid.PCD\_StopCrypto1(); } |

### MySQL adatbázis:.

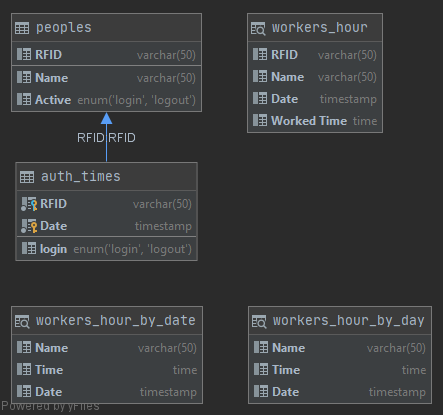


Figure 4 Táblák

A szoftverhoz felhasznált adatbázis kezelő rész.

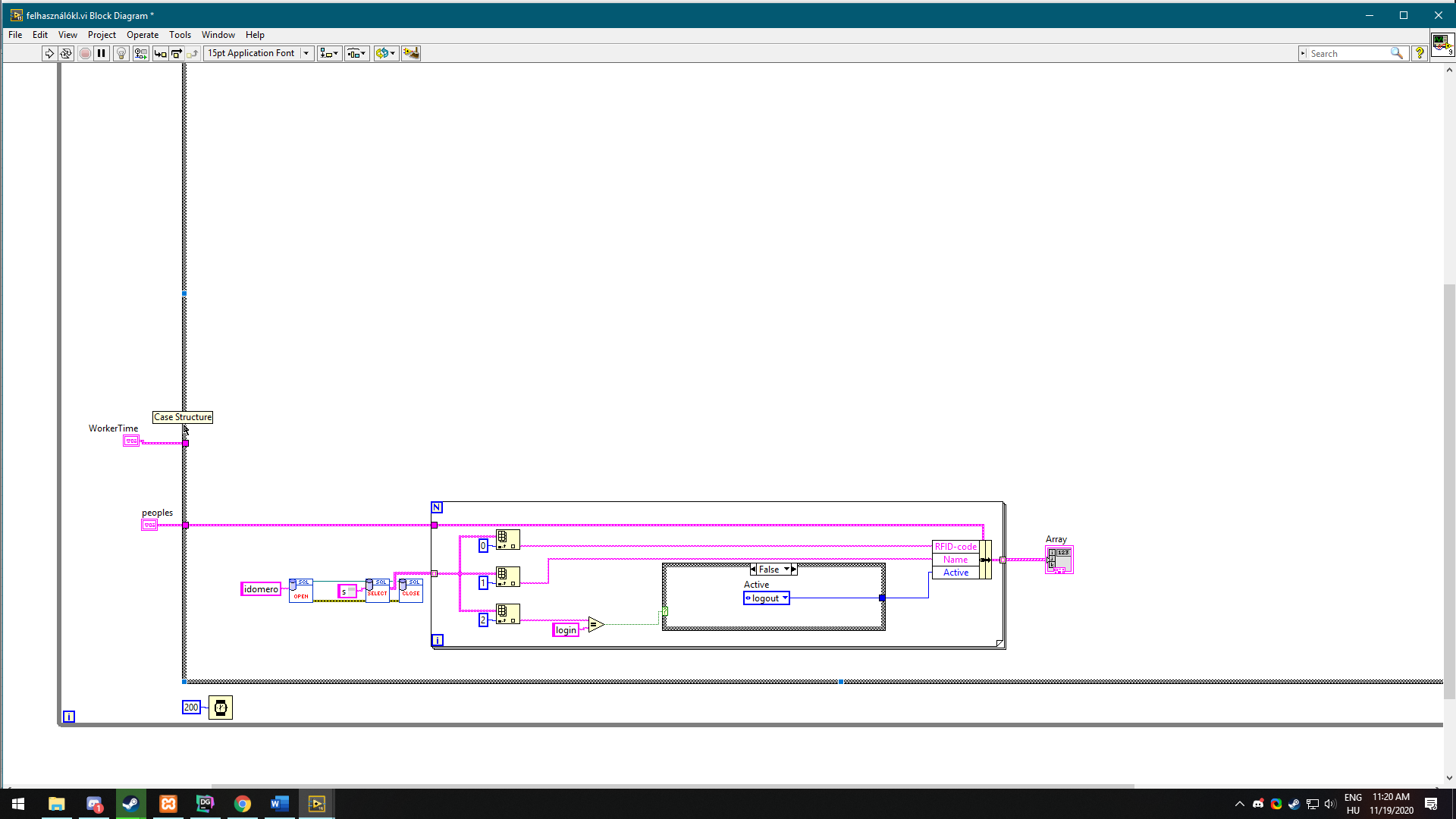
|  |
| --- |
| drop procedure if exists *auth\_times\_pre*; delimiter $$ create procedure *auth\_times\_pre*(in RFID varchar(50))  begin  declare c enum('login','logout');  select Active into c from peoples where peoples.RFID=RFID;  -- select c;  if isnull(c) then  insert into peoples values (RFID,'Temporary Joe',1);  end if;   if c='login' then  -- select 'login';  insert into auth\_times values (RFID,NOW(),'logout');  update peoples set Active='logout' where peoples.RFID=RFID;  else  -- select 'logout';  insert into auth\_times values (RFID,NOW(),'login');  update peoples set Active='login' where peoples.RFID=RFID;  end if;   end $$  DELIMITER ; |

Adatbázisba való adatbevitel.

|  |
| --- |
| <?php $servername = "localhost"; $username = "root"; $password = ""; $conn = new mysqli($servername, $username, $password,"idomero");  //var\_dump($\_GET); if(isset($\_GET["json"])) {  $json=$\_GET["json"];  $res=mysqli\_query($conn,"call auth\_times\_pre('{$json}')");  //var\_dump($res); } |

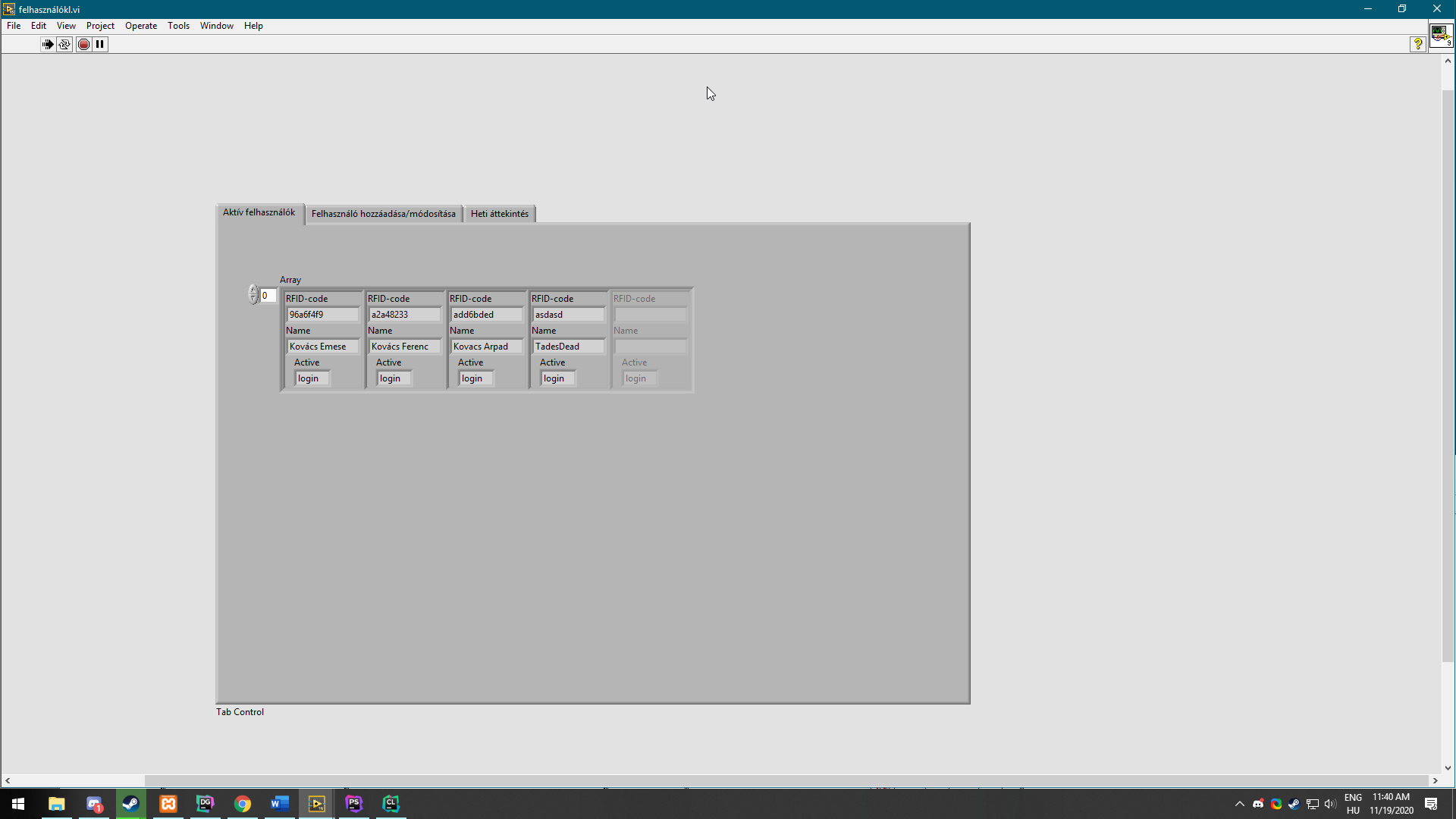
# 3. A LabVIEW szoftver bemutatása

Tartalmazza a szimulációs séma részeit.

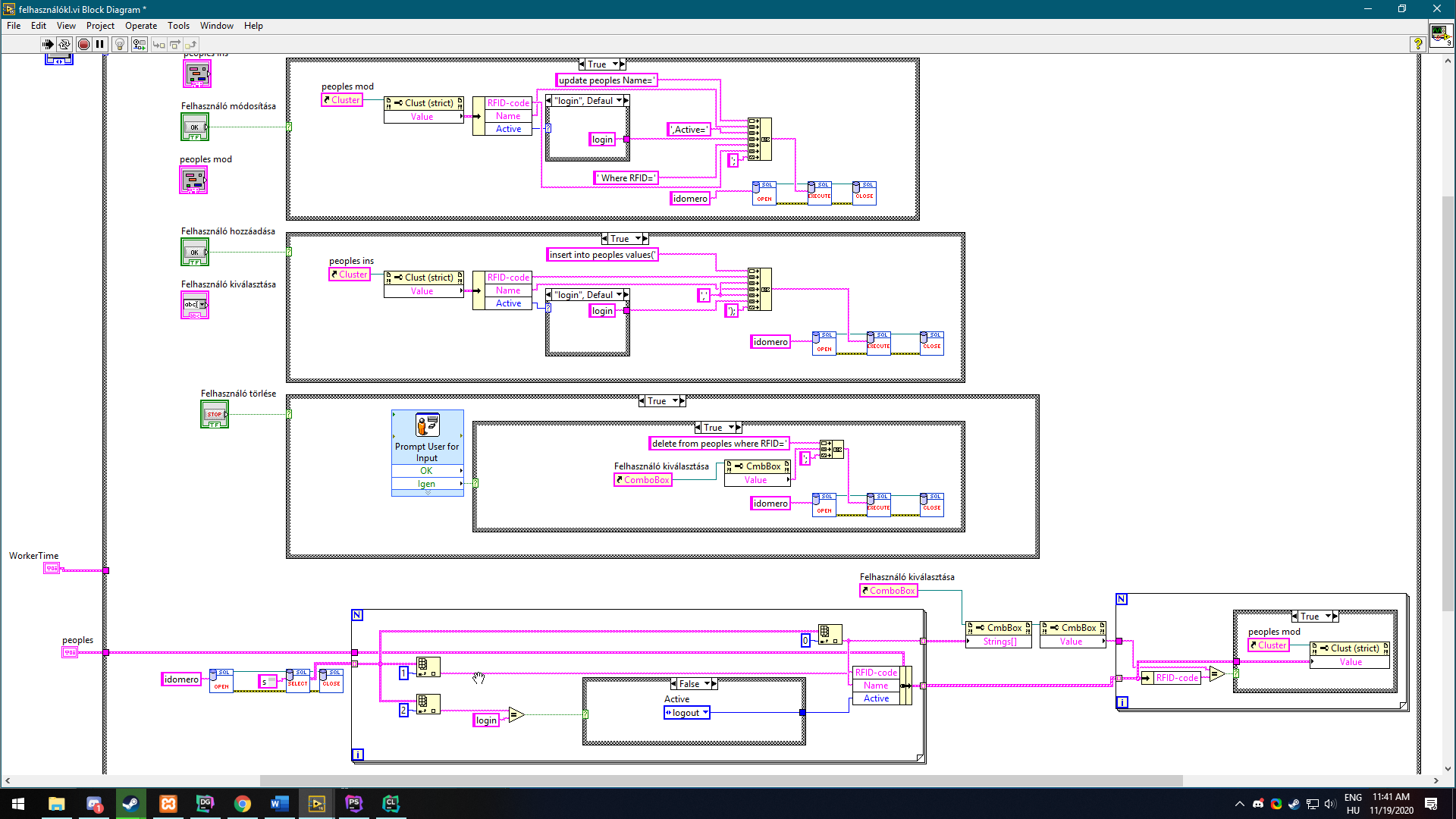


ábra felhasználók áttekintése

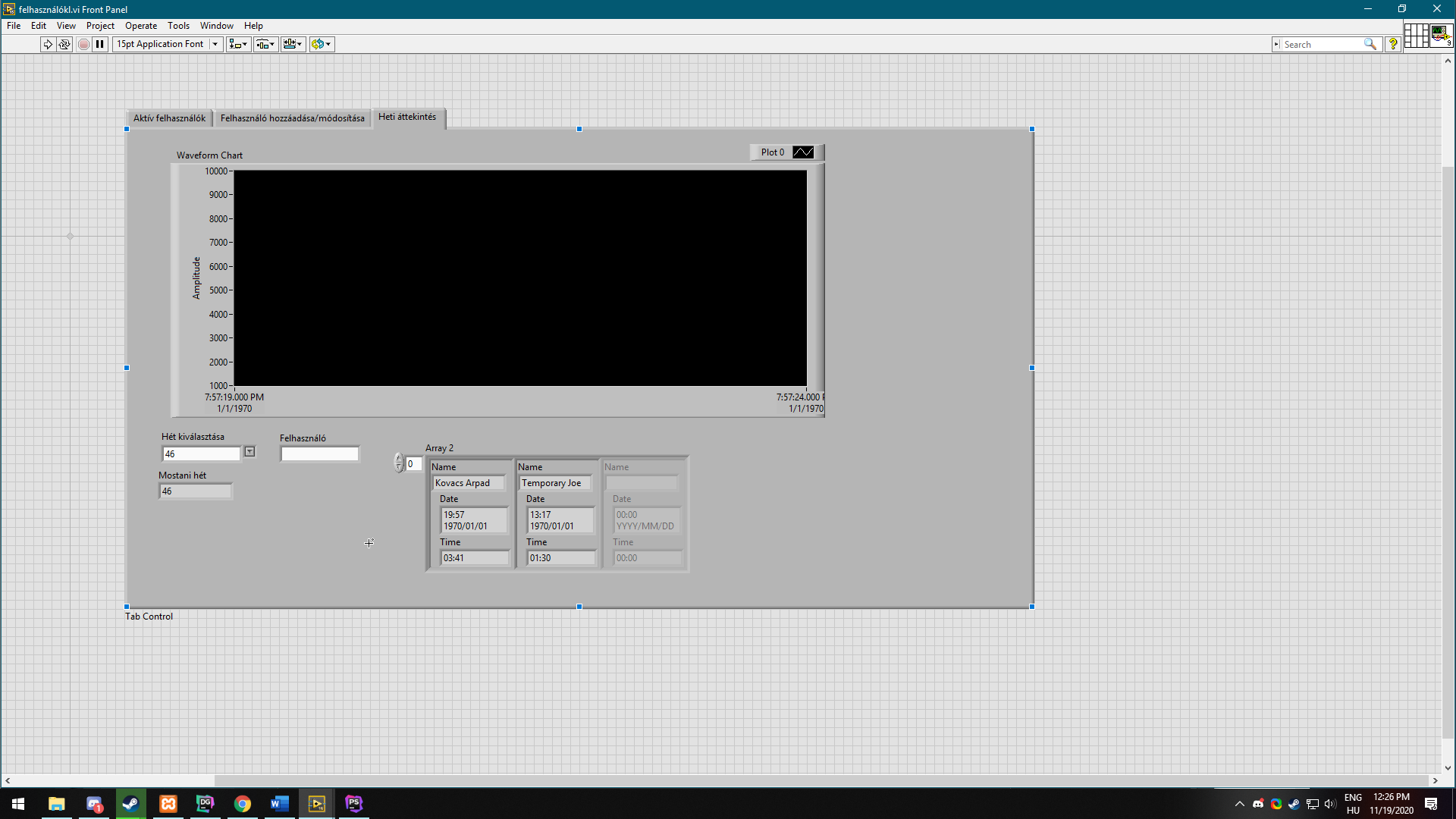
A labview front panelje:.

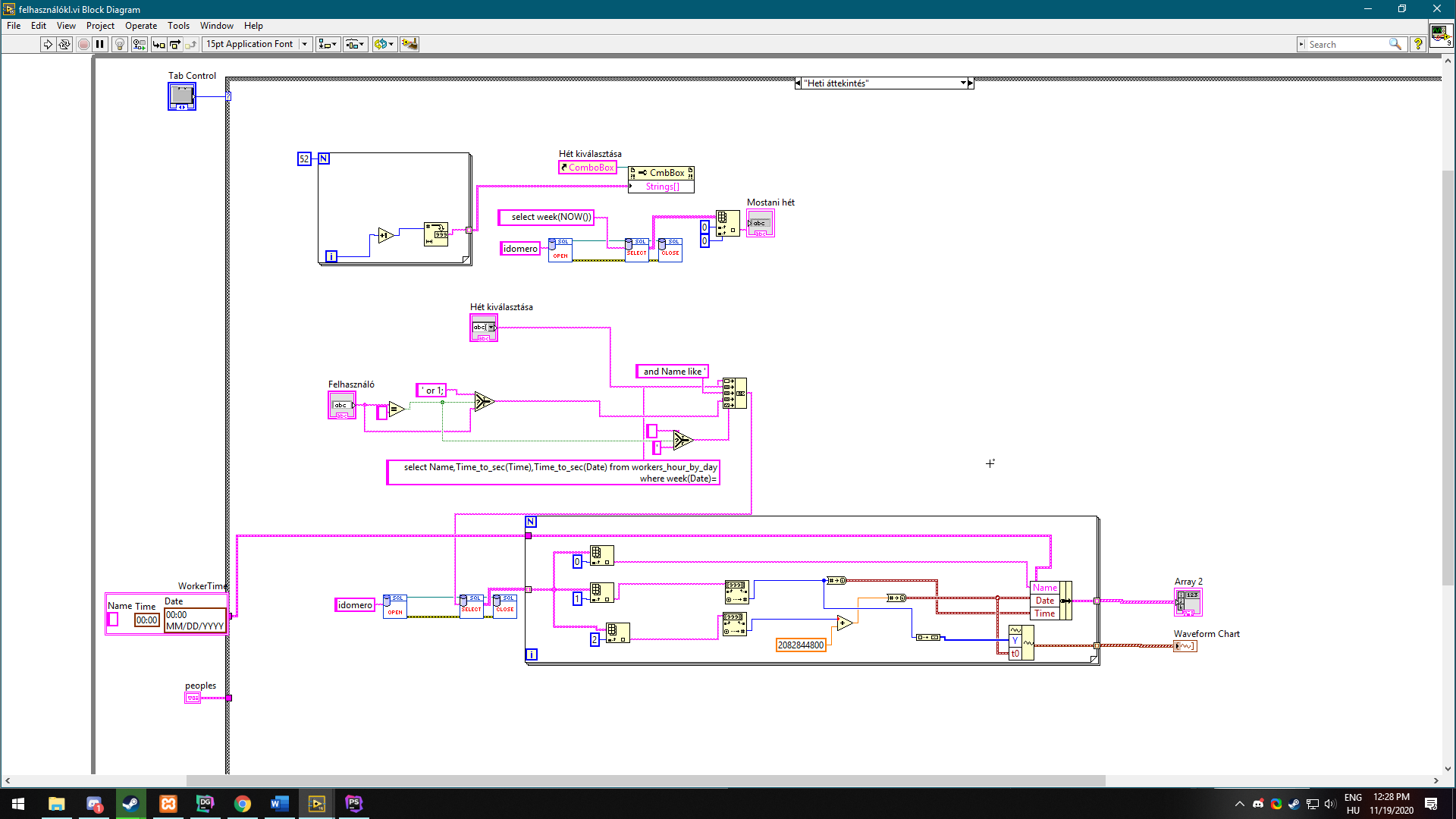


A felhasználók bevitele/módosítása:.



A felhasználók heti áttekintése



A felhasználók heti áttekintése sémája:. 

Kódok:

|  |
| --- |
| drop view if exists workers\_hour; create view workers\_hour as  select a.RFID,p.Name,a.Date,timediff(b.Date,a.Date) as 'Worked Time' from auth\_times a inner join peoples p on a.RFID = p.RFID  right join (select RFID,Date,login from auth\_times aub order by Date) as b on a.RFID=b.RFID where a.Date<b.Date and day(b.Date)=day(a.Date) and a.login='login';  Kijelentkezés és bejelentkezés közötti idő mérés:.  drop view if exists workers\_hour\_by\_day; create view workers\_hour\_by\_day as  select Name, SEC\_TO\_TIME(SUM(TIME\_TO\_SEC(`Worked Time`)))as Time, Date from workers\_hour group by Name;  MySQL Dump  SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO"; START TRANSACTION; SET time\_zone = "+00:00";   /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/; /\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/; /\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/; /\*!40101 SET NAMES utf8mb4 \*/;  -- -- Database: `idomero` --  DELIMITER $$ -- -- Procedures -- CREATE DEFINER=`root`@`localhost` PROCEDURE *`auth\_times\_pre`* (IN `RFID` VARCHAR(50)) begin  declare c enum('login','logout');  select Active into c from peoples where peoples.RFID=RFID;  -- select c;  if isnull(c) then  insert into peoples values (RFID,'Temporary Joe',1);  end if;   if c='login' then  -- select 'login';  insert into auth\_times values (RFID,NOW(),'logout');  update peoples set Active='logout' where peoples.RFID=RFID;  else  -- select 'logout';  insert into auth\_times values (RFID,NOW(),'login');  update peoples set Active='login' where peoples.RFID=RFID;  end if;   end$$  DELIMITER ;  -- --------------------------------------------------------  -- -- Table structure for table `auth\_times` --  CREATE TABLE `auth\_times` (  `RFID` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,  `Date` timestamp NOT NULL DEFAULT current\_timestamp(),  `login` enum('login','logout') COLLATE utf8mb4\_unicode\_ci DEFAULT 'login' ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;  -- -- Dumping data for table `auth\_times` --  INSERT INTO `auth\_times` (`RFID`, `Date`, `login`) VALUES ('8327bf1a', '2020-11-19 11:17:09', 'login'), ('8327bf1a', '2020-11-19 11:17:26', 'logout'), ('8327bf1a', '2020-11-19 11:18:18', 'login'), ('8327bf1a', '2020-11-19 11:18:20', 'logout'), ('8327bf1a', '2020-11-19 11:24:35', 'login'), ('8327bf1a', '2020-11-19 11:24:39', 'logout'), ('8327bf1a', '2020-11-25 13:25:03', 'login'), ('96a6f4f9', '2020-11-19 11:17:01', 'logout'), ('a2a48233', '2020-11-17 19:25:33', 'login'), ('add6bded', '2020-11-17 17:57:24', 'login'), ('add6bded', '2020-11-17 17:58:48', 'logout'), ('add6bded', '2020-11-17 19:25:06', 'login'), ('add6bded', '2020-11-19 11:16:20', 'logout'), ('add6bded', '2020-11-19 11:16:30', 'login'), ('add6bded', '2020-11-19 11:16:46', 'logout'), ('add6bded', '2020-11-19 11:16:48', 'login'), ('add6bded', '2020-11-19 11:16:54', 'logout'), ('add6bded', '2020-11-19 11:17:56', 'login'), ('add6bded', '2020-11-19 11:17:59', 'logout'), ('add6bded', '2020-11-19 11:18:04', 'login'), ('add6bded', '2020-11-19 11:18:06', 'logout'), ('add6bded', '2020-11-19 11:24:45', 'login'), ('add6bded', '2020-11-19 11:24:55', 'logout');  -- --------------------------------------------------------  -- -- Table structure for table `peoples` --  CREATE TABLE `peoples` (  `RFID` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,  `Name` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,  `Active` enum('login','logout') COLLATE utf8mb4\_unicode\_ci DEFAULT 'login' ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;  -- -- Dumping data for table `peoples` --  INSERT INTO `peoples` (`RFID`, `Name`, `Active`) VALUES ('8327bf1a', 'Kovács Péter', 'logout'), ('96a6f4f9', 'Kovács Emese', 'logout'), ('a2a48233', 'Kovács Ferenc', 'login'), ('add6bded', 'Kovacs Arpad', 'logout'), ('asdasd', 'TadesDead', 'login');  -- --------------------------------------------------------  -- -- Stand-in structure for view `workers\_hour` -- (See below for the actual view) -- CREATE TABLE `workers\_hour` ( `RFID` varchar(50) ,`Name` varchar(50) ,`Date` timestamp ,`Worked Time` time );  -- --------------------------------------------------------  -- -- Stand-in structure for view `workers\_hour\_by\_date` -- (See below for the actual view) -- CREATE TABLE `workers\_hour\_by\_date` ( `Name` varchar(50) ,`Time` time ,`Date` timestamp );  -- --------------------------------------------------------  -- -- Stand-in structure for view `workers\_hour\_by\_day` -- (See below for the actual view) -- CREATE TABLE `workers\_hour\_by\_day` ( `Name` varchar(50) ,`Time` time ,`Date` timestamp );  -- --------------------------------------------------------  -- -- Structure for view `workers\_hour` -- DROP TABLE IF EXISTS `workers\_hour`;  CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `workers\_hour` AS SELECT `a`.`RFID` AS `RFID`, `p`.`Name` AS `Name`, `a`.`Date` AS `Date`, timediff(`b`.`Date`,`a`.`Date`) AS `Worked Time` FROM ((select `aub`.`RFID` AS `RFID`,`aub`.`Date` AS `Date`,`aub`.`login` AS `login` from `auth\_times` `aub` order by `aub`.`Date`) `b` left join (`auth\_times` `a` join `peoples` `p` on(`a`.`RFID` = `p`.`RFID`)) on(`a`.`RFID` = `b`.`RFID`)) WHERE `a`.`Date` < `b`.`Date` AND dayofmonth(`b`.`Date`) = dayofmonth(`a`.`Date`) AND `a`.`login` = 'login' ;  -- --------------------------------------------------------  -- -- Structure for view `workers\_hour\_by\_date` -- DROP TABLE IF EXISTS `workers\_hour\_by\_date`;  CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `workers\_hour\_by\_date` AS SELECT `tmp`.`Name` AS `Name`, `tmp`.`Time` AS `Time`, `tmp`.`Date` AS `Date` FROM (select `workers\_hour`.`Name` AS `Name`,sec\_to\_time(sum(time\_to\_sec(`workers\_hour`.`Worked Time`))) AS `Time`,`workers\_hour`.`Date` AS `Date` from `workers\_hour` group by dayofmonth(`workers\_hour`.`Date`)) AS `tmp` GROUP BY `tmp`.`Name` ;  -- --------------------------------------------------------  -- -- Structure for view `workers\_hour\_by\_day` -- DROP TABLE IF EXISTS `workers\_hour\_by\_day`;  CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `workers\_hour\_by\_day` AS SELECT `workers\_hour`.`Name` AS `Name`, sec\_to\_time(sum(time\_to\_sec(`workers\_hour`.`Worked Time`))) AS `Time`, `workers\_hour`.`Date` AS `Date` FROM `workers\_hour` GROUP BY `workers\_hour`.`Name` ;  -- -- Indexes for dumped tables --  -- -- Indexes for table `auth\_times` -- ALTER TABLE `auth\_times`  ADD PRIMARY KEY (`RFID`,`Date`);  -- -- Indexes for table `peoples` -- ALTER TABLE `peoples`  ADD PRIMARY KEY (`RFID`);  -- -- Constraints for dumped tables --  -- -- Constraints for table `auth\_times` -- ALTER TABLE `auth\_times`  ADD CONSTRAINT `auth\_times\_ibfk\_1` FOREIGN KEY (`RFID`) REFERENCES `peoples` (`RFID`) ON DELETE CASCADE ON UPDATE CASCADE; COMMIT; |

# A felhasznált rövidítések

A használt rövidítések jegyzéke és azok jelentése.

RFID-**R**adio **F**requency **ID**entification

MCU-mikrokontroller

# Irodalom

<https://www.w3schools.com/php/php_mysql_intro.asp>

<https://people.vts.su.ac.rs/~simon/bp2/Baze_Podataka2_Prirucnik_SR_2015.pdf>

<https://rf-id.lap.hu/>