Bachelor's thesis

Information and Communications Technology

2020

Ikhlas Jenfi

ATTENDANCE SYSTEM USING RFID-MFRC522

TURKU AMK

TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor's degree in Information and Communications Technology

2020 | 31

Ikhlas Jenfi

ATTENDANCE SYSTEM USING RFID-MFRC522

Nowadays, automatized attendance systems play a huge role in our daily life when it comes to tracking employees. Companies, schools, and public services are some prime examples that take advantage of this technology. Generally, tracking systems are more reliable, accurate, and efficient when it comes to tracking working hours of employees.

The main objective of this thesis was to build a prototype of an attendance system for interns and students in the learning environment of Turku University of Applied Science, theFIRMA. The development took place into four phases. Firstly, the planning and research phase, which includes planning and researching the hardware and software that would be used, such as Raspberry Pi 4 Model microcontroller, RFID-MFRC522, I2C LCD and three different colors LEDs. Secondly, the system requirement and design phase where the system specifications were discussed and planned. Thirdly, the implementation and development phase, where the hardware is put all together and the software is running; and lastly the testing phase, in which the whole system would be tested to be verified that there is no error.

The result of this thesis was a wireless system for hours tracking where a new user can be added, and the user's hours will be counted and saved in the database. Thus, the developed attendance system resulted in time saved for both the employee and their supervisors.

KEYWORDS:

RFID MFRC522, Raspberry Pi 4, LCD I2C model 16x2, Raspbian, Linux.

CONTENTS

FIGURES

PICTURES

LIST OF ABBREVIATIONS

DB	Database
ER	Entity-relationship
FR	Functional requirements
GPIO	General Purpose Input/output
I2C	Multi-Master Bus
LCD	Liquid-Crystal Display
LED	Light-Emitting Diode
Micro SD card	Micro Secure Digital card
NFR	Non-functional requirements
OS	Operating system
QA	Quality Assurance
R&D	Researches and Development
RFID	Radio-frequency identification
SPI	Serial-Peripheral Interface
SQL	Structured Query Language
SyRS	System Requirements Specification

1 INTRODUCTION

This thesis was commissioned by theFIRMA, a public project office located in Turku University of Applied Sciences, Turku, Finland.

theFIRMA is a study and learning environment created by Turku University of Applied Sciences (theFIRMA, n.d). It helps students to learn and improve their skills in real world projects, as it opens job opportunities for some of them.

theFIRMA operates by working on R&D and customers projects, they offer services such as (theFIRMA, n.d):

- Web development: from a simple appearance update to full packages on websites.
- Software development: designing and developing different small software, applications and games.
- Testing: accomplishing and planning for debugging and testing for websites, mobile applications, games and other programs.
- Graphical design / Marketing materials: creating logos, brochures, advertising posters and other marketing material.
- Quality specification: charting software and compiling documents of functional and qualitive requirements.

All of the company employees, students and interns are required to track their hours done daily in an Excel sheet. The Excel sheet attendance is inconvenient, time consuming and non-reliable. Here comes the thesis purpose, making an attendance system that will be more efficient, accurate and reliable. The advantage of this attendance system is that it is of low cost, easy to use, and fast.

The system will consist an RFID reader and writer of MFRC522 module connected with a Raspberry Pi 4 and an I2C LCD and three different colors of LEDs.

The RFID reader and writer will be the main device of this system. Through it, each user will be able to tag his or her RFID fob or card to sign in and sign out.

The I2C LCD will help the user to understand and keep track of his status.

The database will then collect and store each user's data, what time the user signed in or out plus their personal information while simultaneously creating the user. As the database will be able to provide the total time per day that user spent in the office as it could also provide the total time of all user attendance.

The red, yellow, and green LEDs will be used for indicating the status of the attendance system.

2 BACKGROUND

2.1 RFID communication overview

On January 23, 1973, Mario Cardullo invented the first true working ancestor of modern read and write radio frequency with a passive radio transponder memory. The initial device was powered by signal interrogation (Global Venture, n.d).

Nowadays, RFID is a technology used for tracking, identification and detection. The technology is based on data storage in the RFID card or fob, and thanks to the radio waves that make the wireless reading of RFID reader possible (Phillips, 2017), as shown in Figure 1.

The main idea of the RFID is that its fob or card is attached to the designated object. The data is then read and written from the fob or card using the RFID reader, and finally data is saved in the database (Last Minute Engineers, n.d).

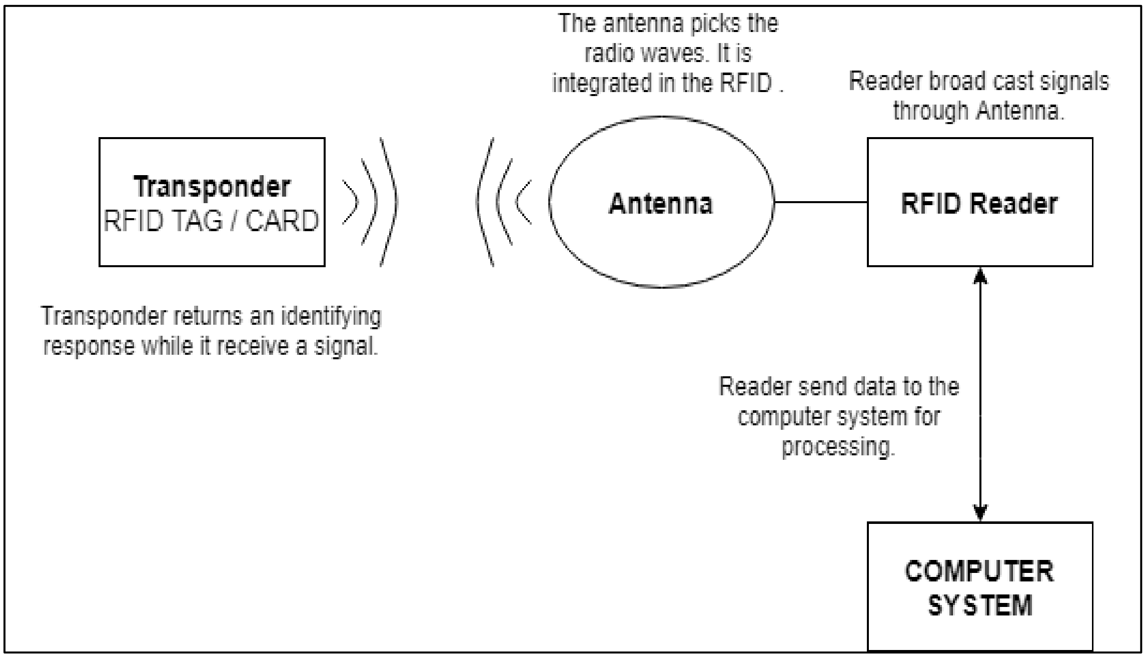

The antenna picks the radio waves. It is integrated in the RFID .
Reader broad cast signals through Antenna.
Transponder
RFID TAG / CARD
Antenna
RFID Reader
Transponder returns an identifying response while it receive a signal.
Reader send data to the computer system for processing.
COMPUTER SYSTEM

Figure 1. Diagram showing how RFID works.

2.2 RFID usage

RFID has three primary frequency ranges used for its transmissions (AtlasRFIDstore, n.d):

- Low frequency: it has a primary frequency range of 125 kHz to 134 kHz, which makes its read range from a 0 to 10 centimeters far away from the RFID reader. It is usually used for access control, human or animal tracking, car key-fob and other more similar fields.
- High frequency: with a primary frequency range of 13.56 MHz, which allows its read range from 0 to 30 centimeters. This frequency is mostly used in DVD kiosks, personal ID cards, NFC applications, library books and poker or gaming chips.
- Ultra-High frequency: has a primary frequency range of 433 MHz to 960 MHz. The ultra-high frequency has two types of RFID, active and passive. The active RFID has a read range of 30 to +100 meters. It is usually used in auto manufacturing, asset or vehicle tracking, mining and other similar areas.
  The passive RFID has a read range of 0 to 25 meters, and it is mostly used in electronic tolling, inventory tracking, pharmaceuticals and also manufacturing.

For the purposes of this project thesis, a low frequency RFID has been used, RFID MFRC522 with its fob/card. The fob/card contains a small radio transponder, a radio receiver and transmitter, as shown in Figure 1. The other advantage of this system is that its RFID fobs or cards are very cheap, simple and fast to use. While on the other hand, other systems as alternatives can be more expensive and less efficient. This attendance system will address each fob or card to one user, this user will have the ability to sign in and out, the role of the system is to save the data of the user in the database. When the user signs in and out with his/her total hours will be calculated.

3 SOFTWARE DEVELOPMENT

Developing a software might be challenging, as it has many different processes for it to be able to succeed. Starting with the planning phase, its purpose is to find the main problem and determine ideas and solutions that will need to be solved. Followed by functional requirements and specifications phase, where system analysis and design takes place and that is by discussing and creating how the system shall be working and what the customer is expecting, with the combination of this last two, the system will meet the requirements of its customer and satisfy both the user and the client. After those phases, the next procedure is the software or system development. This is the most important phase of the whole process because without it there would be nothing but just empty words on a piece of paper. The development phase is the critical step of the process. The real work begins because it marks the end of the initial section. Once the third phase is completed, the integration and the testing phases will start. It is involving the system integration and testing which usually are carried out with QA, and it is where the QA professionals ascertain that the design and the whole system meets the initial set of the system goals and client's needs (Innotiative Architects, n.d).

All those phases are mandatory to have in a system, so that it will be working perfectly and meets all the desired criteria.

3.1 Planning phase

Planning the system might be one of the most challenging phases, it is the first phase in the development process. The company's expectation for this project was to have a simple working attendance system that would count the total hours of its workers with saving their time and date of signing in and out into a database. While preparing the planning stage, an opening report for the project has been written where all steps have been discussed. The phase includes naturally the main idea of this thesis which was to develop an attendance system using an RFID MFRC522 model, a Raspberry Pi 4 model and an I2C LCD display.

The general idea of the system was that the user will sign in or out using their RFID fob or card, the I2C LCD will display the action of the user, and in case the user does not exist the LCD will display a message warning that the RFID fob or card is not saved in a user’s name in the database. As for the LEDs, the green LED will turn on if the user signs in successfully, red LED when the user signs out and yellow LED when the RFID fob or card is not properly saved under a user’s name.

3.2 Functional requirements and specifications phase

This phase is the most important and demanding of all of the phases, it is an important part of the software design and development. A well-designed ground work for the functional requirements and specifications are the key elements for of the whole process of the design and development (Altexsoft, 2018). The more time spent on this phase, the happier customers are with the final product and the easier it is for the developer to develop the system. It is important to note that the system analysis takes place so that the new system can match the needs of the end users and meet their requirements. In addition, this program has to be precise and easy to use.

3.2.1 System requirements

3.2.1.1 User story: user of the system

The user story is a part of the software development. It is informative as it describes one or more features of a user using the system. The user story is typically written from the system user’s or end user’s perspective (T2informatik, n.d).

Here are the user stories for this thesis project (Cohn, 2008):

- ➢ As a system administrator, I can add/overwrite a new student or intern to the system so that the attendance system will recognize the user and the RFID fob or card destinated to this person.
- ➢ As a system administrator, I should always be able to help the users of the attendance system in case they faced difficulties or forgot to sign in or out from the system.
- ➢ As an intern or a student, I need to always carry my RFID fob or card to be able to mark my hours in the attendance system.

3.2.1.1.1 Use case example

In system requirements, a use case is derived from user stories. Its diagram is the main form for a new, but yet undeveloped system. It shows and clarifies the outcome, and in particular, what the system will do as it specifies how the system will work. In order to achieve the goal and the outcome, it is important to have the use case and show the actual actions and steps that the user has to follow (Alexander, 2003), as shown in Figure 2 and Figure 3.

In contrast, a misuse case is a negative form of the use case. In simpler terms it is a form of the use case but in the user point of view. It does describe a user performance in case of a malicious act targeted against the system. Misuse case has many possible applications although it occurs in a highly specific situation, it is continuously threatening the system (Adams, 2011).

The use and misuse cases are developed recursively, going from the level of system to the subsystem level.

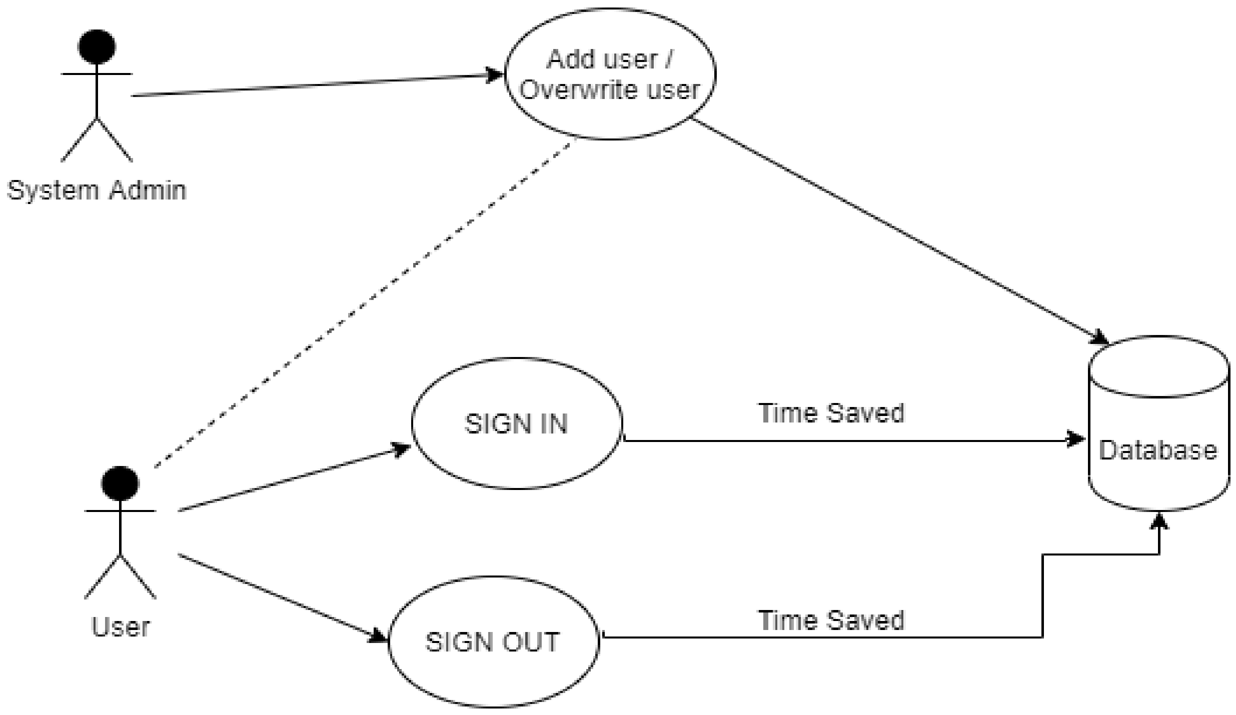


Figure 2. Use cases diagram.

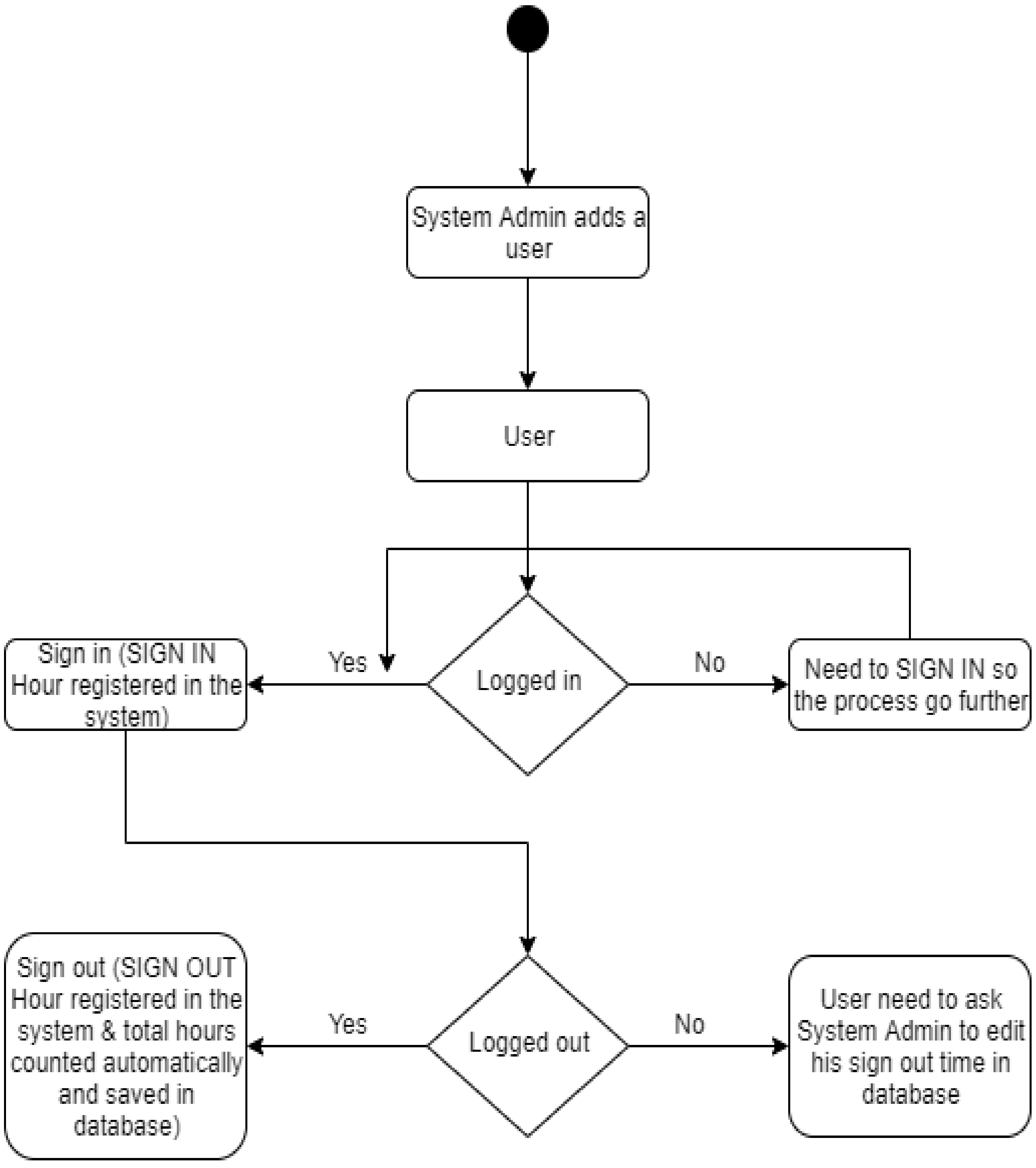

System Admin adds a user
User
Logged in
Yes
No
Sign in (SIGN IN Hour registered in the system)
Need to SIGN IN so the process go further
Logged out
Yes
No
Sign out (SIGN OUT Hour registered in the system & total hours counted automatically and saved in database)
User need to ask System Admin to edit his sign out time in database

Figure 3. Use cases flowchart diagram.

3.2.1.2 System requirements specification (SyRS)

3.2.1.2.1 Functional requirements (FR)

- The system shall record log-in and log-out times of all users into the database.
- The system needs to identify each user with his/her own RFID fob or card.
- The system shall provide the data for calculation of total logged hours by the user.
- The system shall turn on the green LED for 3 seconds when the user signs in with the RFID fob or card.
- The system shall turn on the red LED for 3seconds when the user signs out with the RFID fob or card.
- The system shall turn on the yellow LED for 3 seconds when the RFID fob or card is not recognized by the system.
- The I2C LCD display needs to show the current status of the RFID fob or card when user signs in or out from the system.
- The system needs throw an error message on the I2C LCD display when the RFID fob or card is not recognized.
- The system administrator should be able to sign in or out the user manually if the user forgot to use the system.
- The system administrator should be able to correct the time of signing in or out manually for the user.
- The system administrator shall have the ability to edit the RFID fob/card ID's in the database.

3.2.1.2.2 Non-functional requirements (NFR)

- The microcontroller along with the database should always be on so that the system could work fully.
- The system needs to be running on a Raspberry Pi model 4, using an RFID MFRC522 model, an I2C LCD display and a green, yellow and red LEDs.
- Only system administrator knows all passwords access to the system.
- The administrator has the reponsability to set up fob/card ID's in the database so that the system can operate as intended.

3.2.1.2.3 State Diagram

State diagram describes how the system behaves. It requires the description and the analysis by a series of events which can occur in one or multiple states in the system, shown in Figure 4.

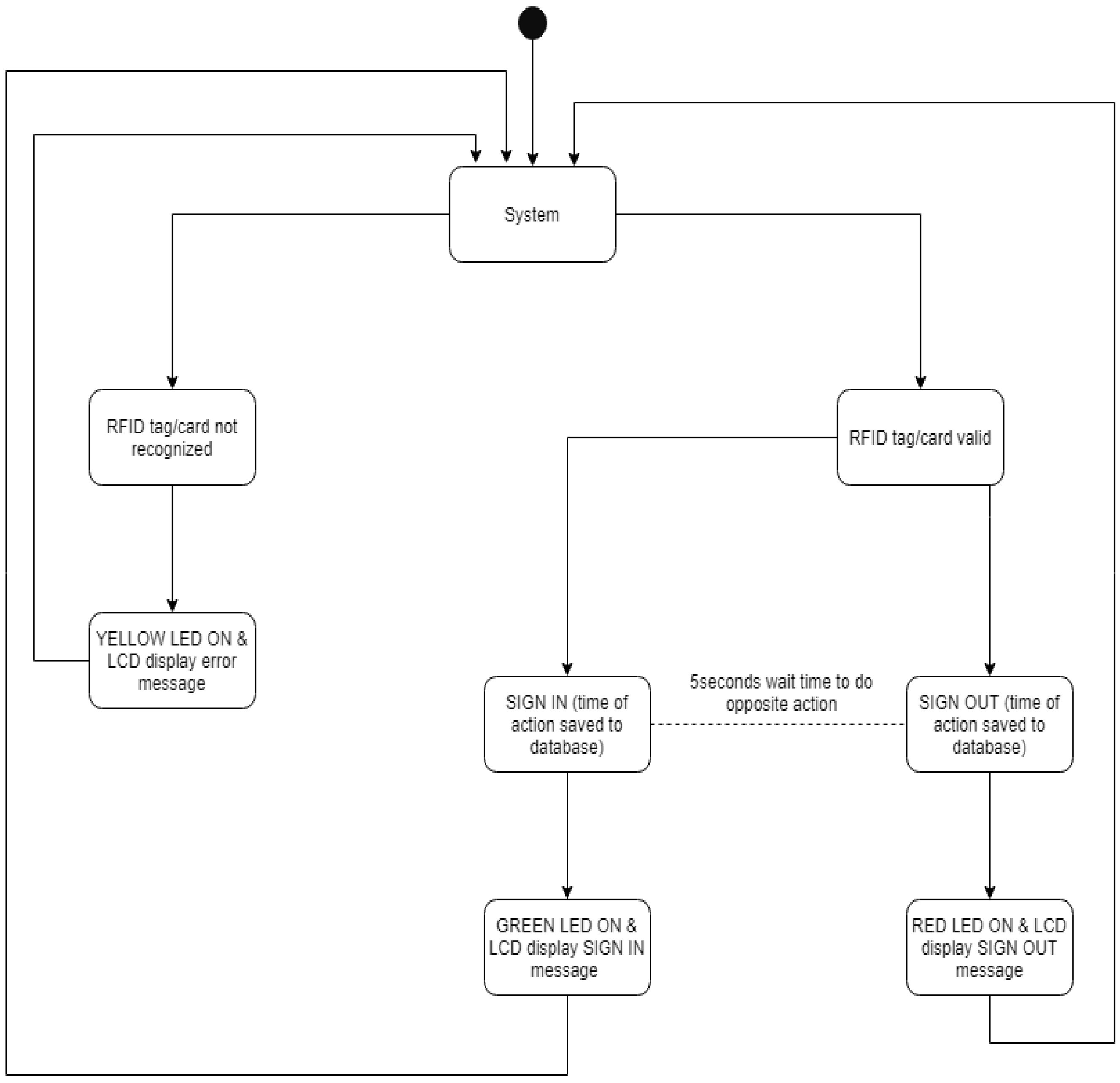

System
RFID tag/card not recognized
RFID tag/card valid
YELLOW LED ON & LCD display error message
SIGN IN (time of action saved to database)
5seconds wait time to do opposite action
SIGN OUT (time of action saved to database)
GREEN LED ON & LCD display SIGN IN message
RED LED ON & LCD display SIGN OUT message

Figure 4. System functioning state diagram.

3.3 Development phase

Development stage is considered the most important phase of the system development cycle. Although the planning, the system requirements, and specifications phases are the base of all the next steps to succeed, the system development phase remains the principal and major. In view of the fact, it marks the end of the initial section of the entire process.

In this thesis project, based on the company's needs, it was discussed by the commissioner of this thesis that the design and the planning of the software was free to be chosen the best way possible for it to be ideal for the development and for the users.

The first step of the development phase was to have a micro SD card that will hold the Raspbian OS (Raspberry Pi Org, n.d 1). Once the correct OS is downloaded for the Raspberry Pi 4 model, the SD card will be plugged in the microcontroller and set into the convenient setting that the user would like and that will suit the user's work. Before booting for the first time the Raspberry Pi, a keyboard, mouse, monitor and HDMI cable are needed for the microcontroller to work as a full computer. Once it turns on, the basic settings will need to be done. The next step was to go to the terminal and update and upgrade the OS.

Here the real work begins. The first step was to search and retrieve the datasheet of the components used to be able to utilize them with the microcontroller the right way.

3.3.1 I2C LCD set-up

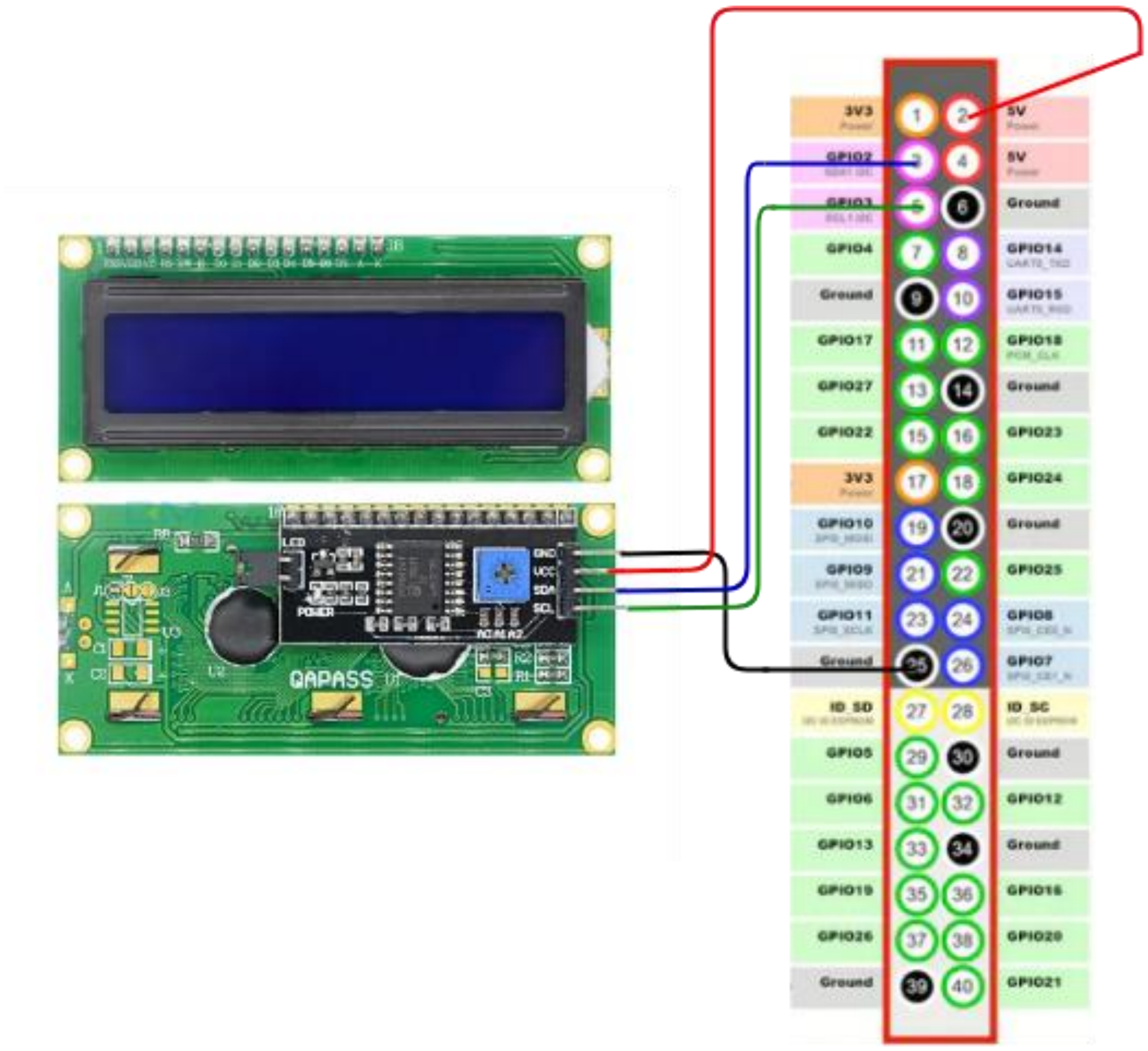
As for the I2C LCD display, it is important to enable the I2C on the Raspberry Pi. Following that, some installments within the terminal are needed for the LCD to work perfectly. The I2C tool is required because it tells the address of the I2C LCD:

```
sudo apt-get install i2c-tools
```

The SMBUS gives a python library that uses access to the I2C bus on the Raspberry Pi:

```
sudo apt-get install python-smbus
```

Once finishing installing the I2C tools and the SMBUS, the Raspberry Pi need to be rebooted. And I2C LCD need to be set up with the microcontroller as shown in Picture 1.



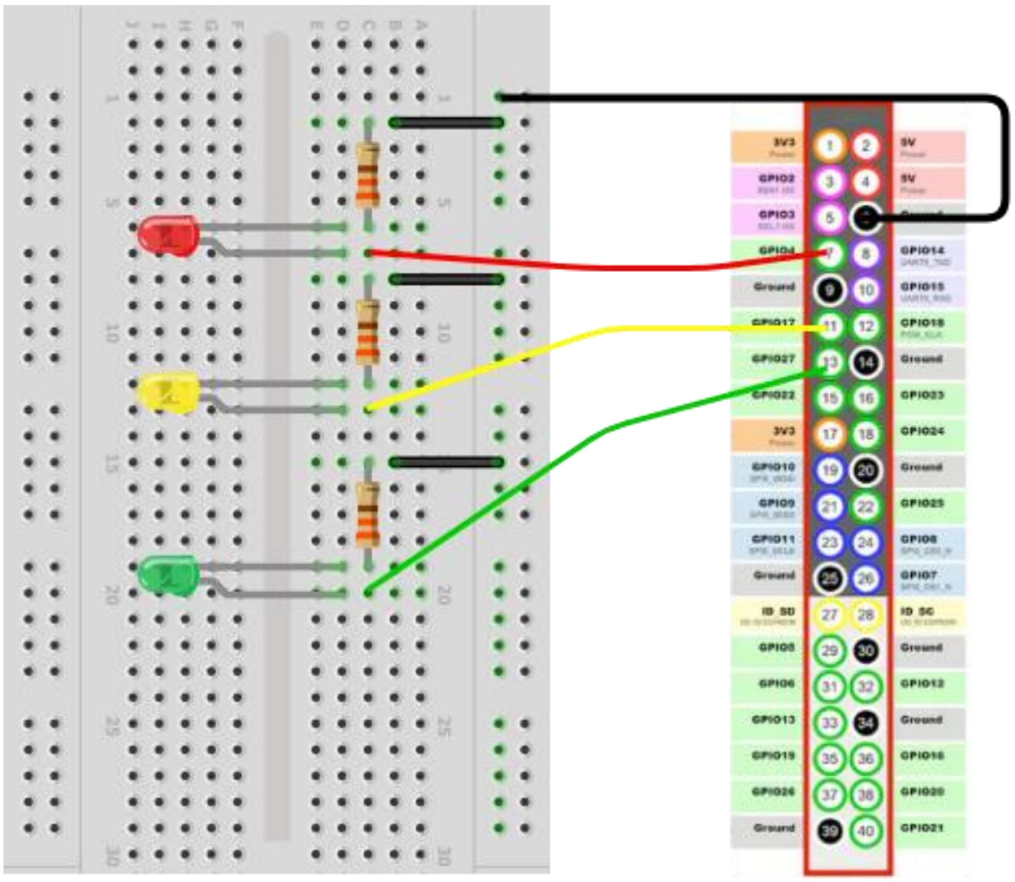
Picture 1. The I2C LCD set-up with the Raspberry Pi 4.

- 4 female to female wires are needed for this set-up.
- GND will be connected to a ground GPIO in the Raspberry Pi (Physical Pin 6).
- VCC will be connected to 3.3V (Physical Pin 1).
- SDA will be connected to GPIO 2 (Physical Pin 3).
- SCL will be connected to GPIO 3 (Physical Pin 5).

The next step is to download a python I2C library that has a good set of functions and works well with no bugs. In the I2C library file in python, it might be demanded to change the port number of the I2C bus and that is depending on the model of the microcontroller used. 0 is used for older models while 1 is used for newer ones. As it might be also needed to change the number of the I2C address and that is depending on the address of the I2C we have. Once all is set-up, testing the display by writing something on it is going to be the next step to verify that the LCD is working perfectly (MATT, 2015).

3.3.2 LEDs set-up

To set-up 3 LEDs: red, green and yellow. Three male to male and 4 male to female jumper wires, 3 of 330Ohm resistors and a breadboard will be needed. The set-up needs to be as shown in Picture 2.

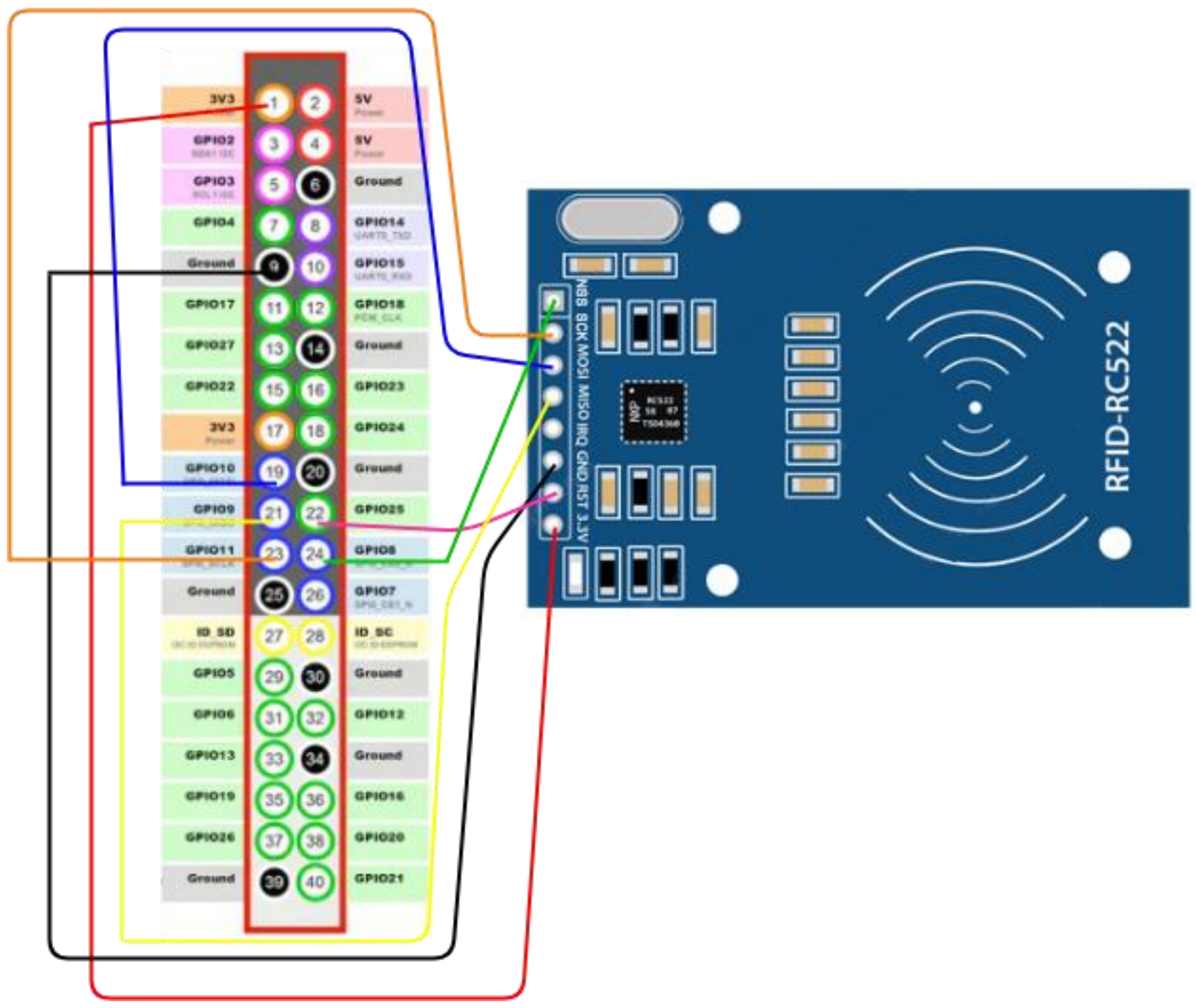


Picture 2. LCDs set-up with the microcontroller.

- Red LED in GPIO4 (Physical Pin 7).
- Yellow LED in GPIO17 (Physical Pin 11).
- Green LED in GPIO27 (Physical Pin 13).

Once the LCDs are set-up, we run our code to test them and verify to approve that there is no error, and all is working as expected (Thepihut, 2015).

3.3.3 RFID MFRC522 set-up



Picture 3. The RFID-MFRC522 set-up with Raspberry Pi 4.

For the RFID-RC522 reader/writer set-up, shown in Picture 3:

- 7 female to female wires will be needed.
- GND will be connected to a ground GPIO in the Raspberry Pi (Physical Pin 9).
- VCC will be connected to 3.3V (Physical Pin 17).
- RST will be connected to GPIO25 (Physical Pin 22).
- MISO will be connected to GPIO9 (Physical Pin 21).
- MOSI will be connected to GPIO10 (Physical Pin 19).
- SCK will be connected to GPIO11 (Physical Pin 23).
- NSS will be connected to GPIO8 (Physical Pin 24).

- IRQ will not be connected to any GPIO in the raspberry pi.

Following the previous action, installation of packages in the OS terminal, such as build-essential, git, python3-dev, python3-pip, and python3-smbus are required for this project to work:

```
sudo apt-get install build-essential git python3-dev python3-pip python3-smbus
```

For the RFID reader and writer, it is important to enable the SPI interface in the Raspberry Pi for its communication with the RFID-MFRC522 module, and then followed by rebooting the microcontroller. When the computer is on, it is recommended to check the status of the SPI in the kernel module by using this command in the terminal:

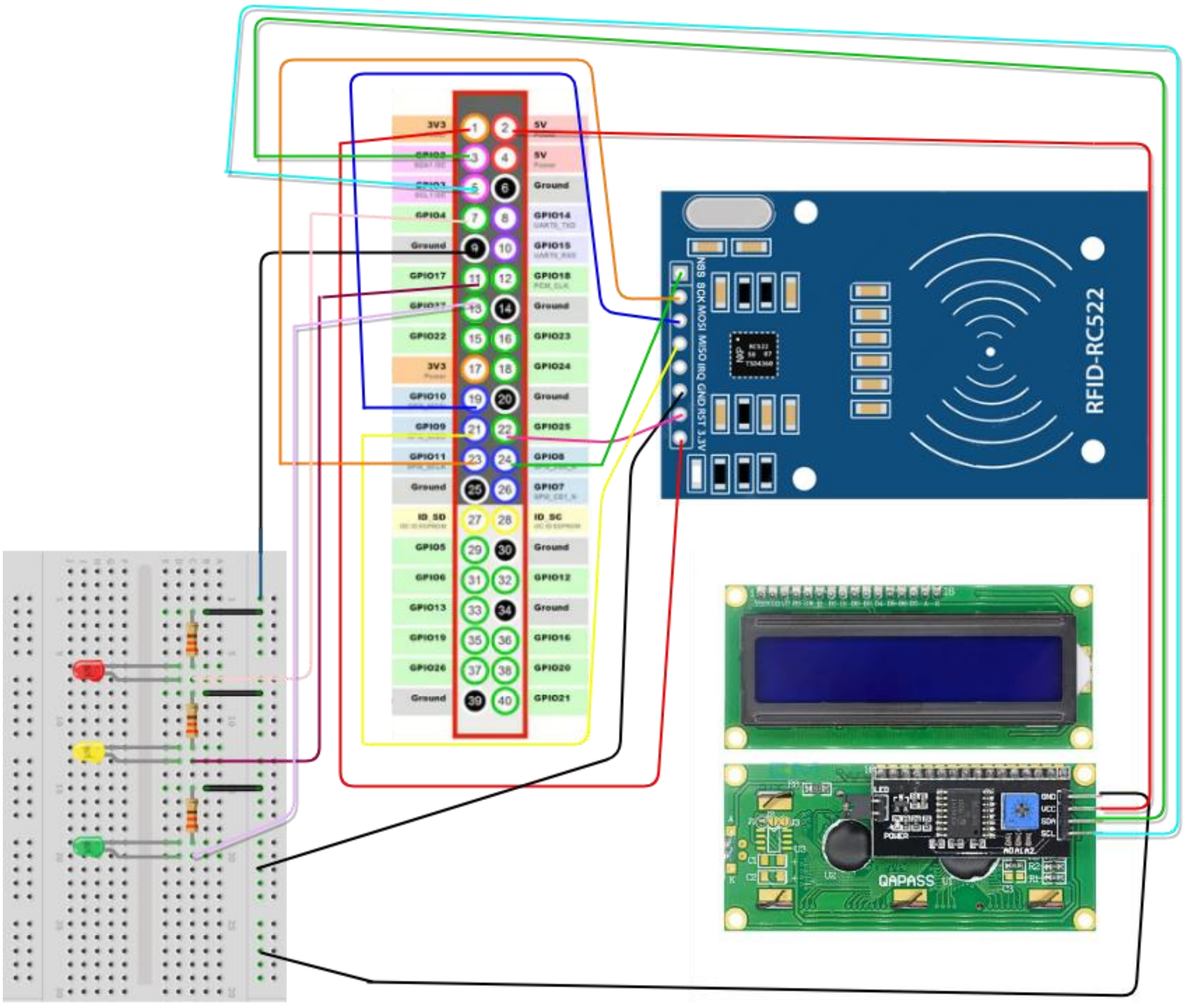
```
lsmod | grep spi
```

If the ”spi_bcm2835” appears, next step will be to install libraries for the RFID reader and writer such as ’spidev’ which interacts with the RFID reader to interface and ’mfrc522’ that will handle the attendance system’s grunt work (Emmet, 2019).

Finally, testing the RFID-MFRC522 reader and writer are done at this step to check that the RFID is working.

After testing the RFID, the next step will be to install a database which through it, data of the RFID fob or card and its user will be saved. The latest version of MariaDB will be installed, following the creation of a database and its tables. In this case, a main table which has all users will have the ID of each user, user’s name, email, and RFID ID. In the other hand, the other table will contain the user ID, ID for each row created, and the time of sign in and out and the total hours attended daily.

3.3.4 The attendance system set-up



Picture 4. The attendance system diagram.

At this step, the attendance system needed to be exactly as described in the previous steps and the final diagram in Picture 4. Since we previously tested each component alone to prove that all are working. We will now need to combine and develop our code of what is needed and connect the attendance system to the database. This system will run in a Linux Raspbian environment.

Below in Figure 5 is an ER diagram of the database tables:

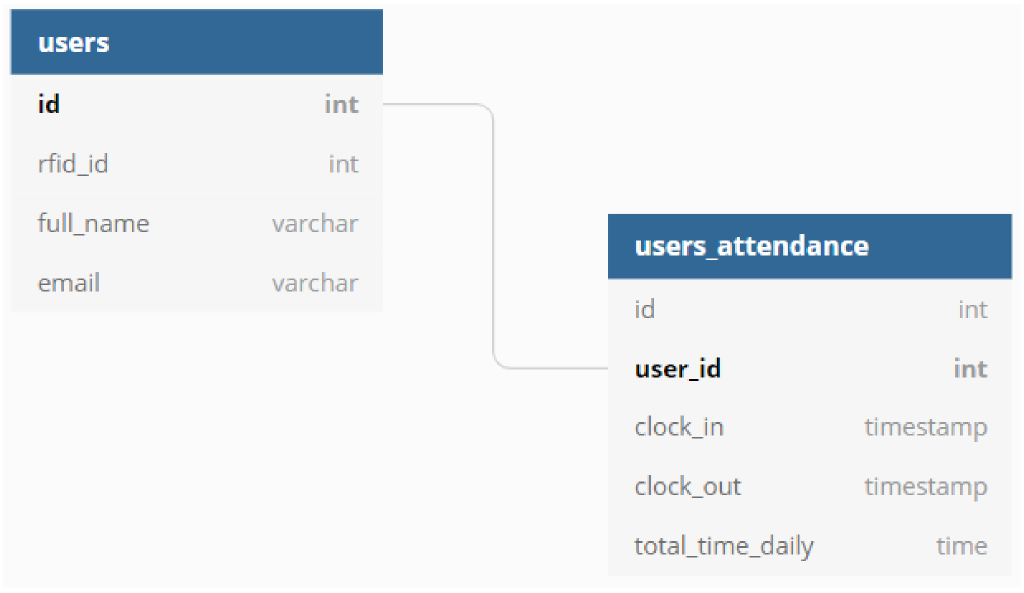


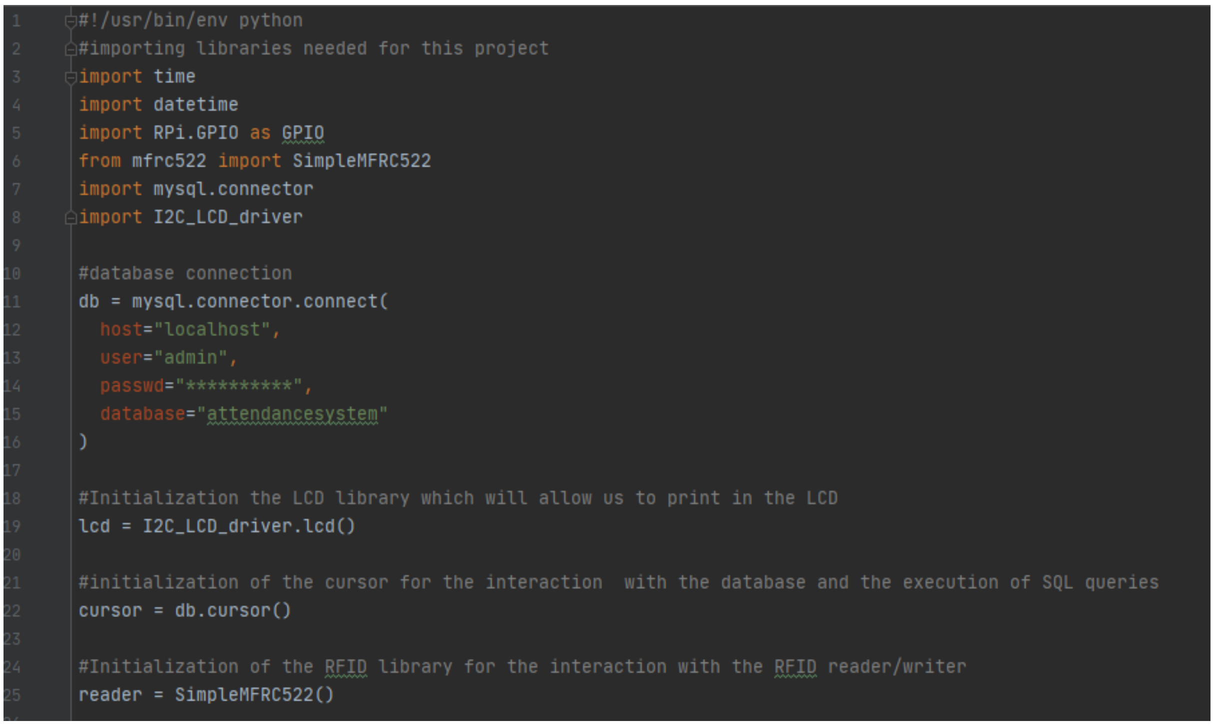
Figure 5. The database diagram.

The attendance system will consist of two files to run. The first file under the name of "save_user.py" which is in charge of saving RFID fob or card ID to a new user or overwrite the existing user in case of a spelling mistake or change of the possession of the RFID fob or card of the user. This file needs to be run in the terminal by the system administrator who is going to be in charge of the system. Through this file the system administrator will be able to add or overwrite user and save it to the database. The database table of this file consists of an 'id' that is going to be auto-incremented and also the private key of this table, it will be the user's ID which will be addressed unique to each user. A 'rfid_id' that will hold the ID of the RFID fob/card, a 'full_name' column that will consists of a full name of the user and an 'email' of each of the user which will make the difference in case there are similar names.

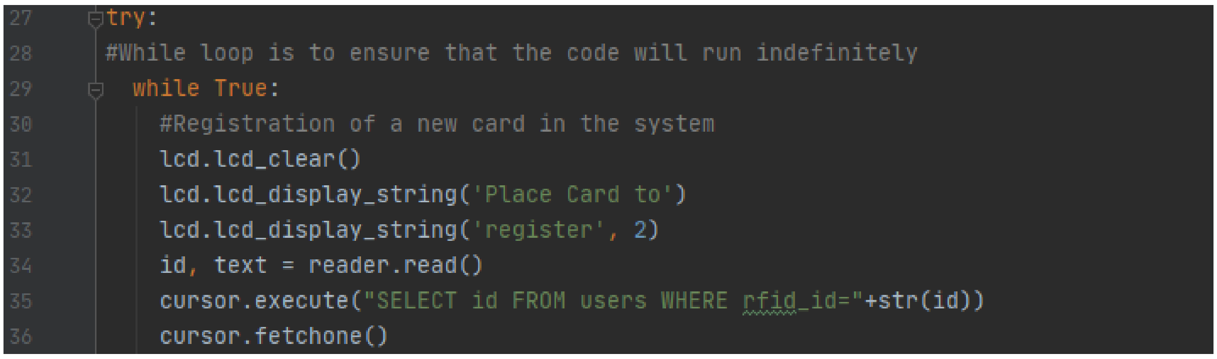
Principally, when the code will be executed it will either allow you to add a new user or give you the choice to overwrite and delete previous user in case you hold an already existing RFID fob/card that was previously saved in the system.

To run the code of the file "save_user.py", the system administrator needs to open the Linux Raspbian terminal and execute:

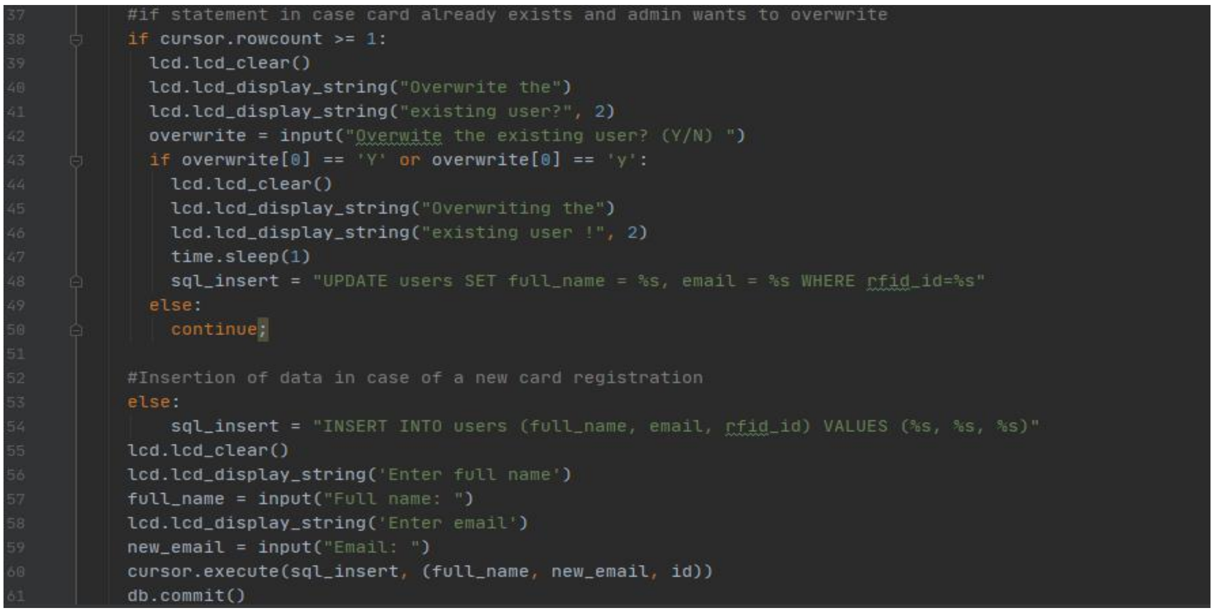
```
sudo python3 ~/attendancesystem/save_user.py
```



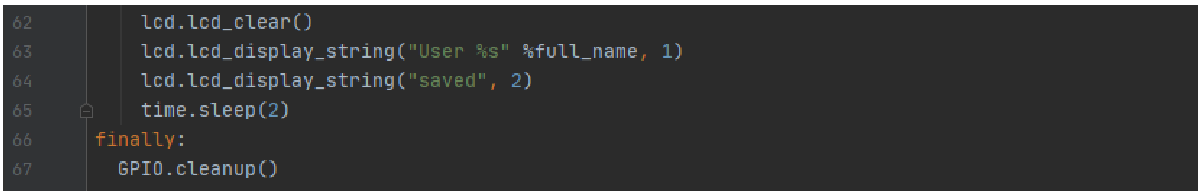
```python
#!/usr/bin/env python
#importing libraries needed for this project
import time
import datetime
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
import mysql.connector
import I2C_LCD_driver

#database connection
db = mysql.connector.connect(
  host="localhost",
  user="admin",
  passwd="**********",
  database="attendancesystem"
)

#Initialization the LCD library which will allow us to print in the LCD
lcd = I2C_LCD_driver.lcd()

#initialization of the cursor for the interaction  with the database and the execution of SQL queries
cursor = db.cursor()

#Initialization of the RFID library for the interaction with the RFID reader/writer
reader = SimpleMFRC522()
```



```python
try:
#While loop is to ensure that the code will run indefinitely
 while True:
  #Registration of a new card in the system
  lcd.lcd_clear()
  lcd.lcd_display_string('Place Card to')
  lcd.lcd_display_string('register', 2)
  id, text = reader.read()
  cursor.execute("SELECT id FROM users WHERE rfid_id="+str(id))
  cursor.fetchone()
```



```python
  #if statement in case card already exists and admin wants to overwrite
  if cursor.rowcount >= 1:
   lcd.lcd_clear()
   lcd.lcd_display_string("Overwrite the")
   lcd.lcd_display_string("existing user?", 2)
   overwrite = input("Overwite the existing user? (Y/N) ")
   if overwrite[0] == 'Y' or overwrite[0] == 'y':
    lcd.lcd_clear()
    lcd.lcd_display_string("Overwriting the")
    lcd.lcd_display_string("existing user !", 2)
    time.sleep(1)
    sql_insert = "UPDATE users SET full_name = %s, email = %s WHERE rfid_id=%s"
   else:
    continue;

  #Insertion of data in case of a new card registration
  else:
    sql_insert = "INSERT INTO users (full_name, email, rfid_id) VALUES (%s, %s, %s)"
  lcd.lcd_clear()
  lcd.lcd_display_string('Enter full name')
  full_name = input("Full name: ")
  lcd.lcd_display_string('Enter email')
  new_email = input("Email: ")
  cursor.execute(sql_insert, (full_name, new_email, id))
  db.commit()
```



```
  lcd.lcd_clear()
  lcd.lcd_display_string("User %s" %full_name, 1)
  lcd.lcd_display_string("saved", 2)
  time.sleep(2)
finally:
 GPIO.cleanup()
```

The result of this file running in the database will look like:

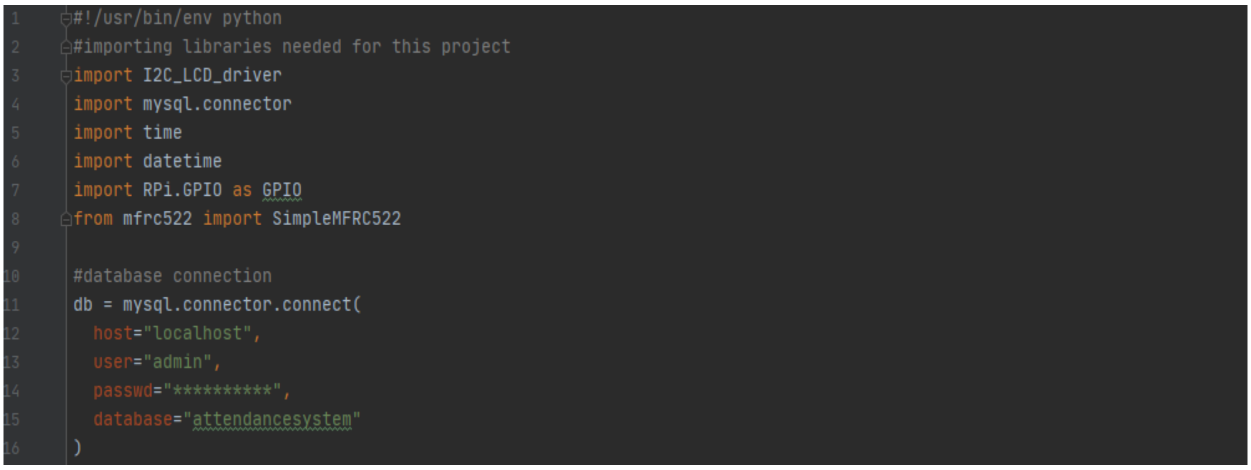
```
SELECT * FROM users;
```

```
+----+----------------------+----------------+---------------------------------------+
| id | rfid_id              | full_name      | email                                 |
+----+----------------------+----------------+---------------------------------------+
|  1 | 97754106881          | Ikhlas Jenfi   | ikhlas.jenfi@edu.turkuamk.fi |
+----+----------------------+----------------+---------------------------------------+
```

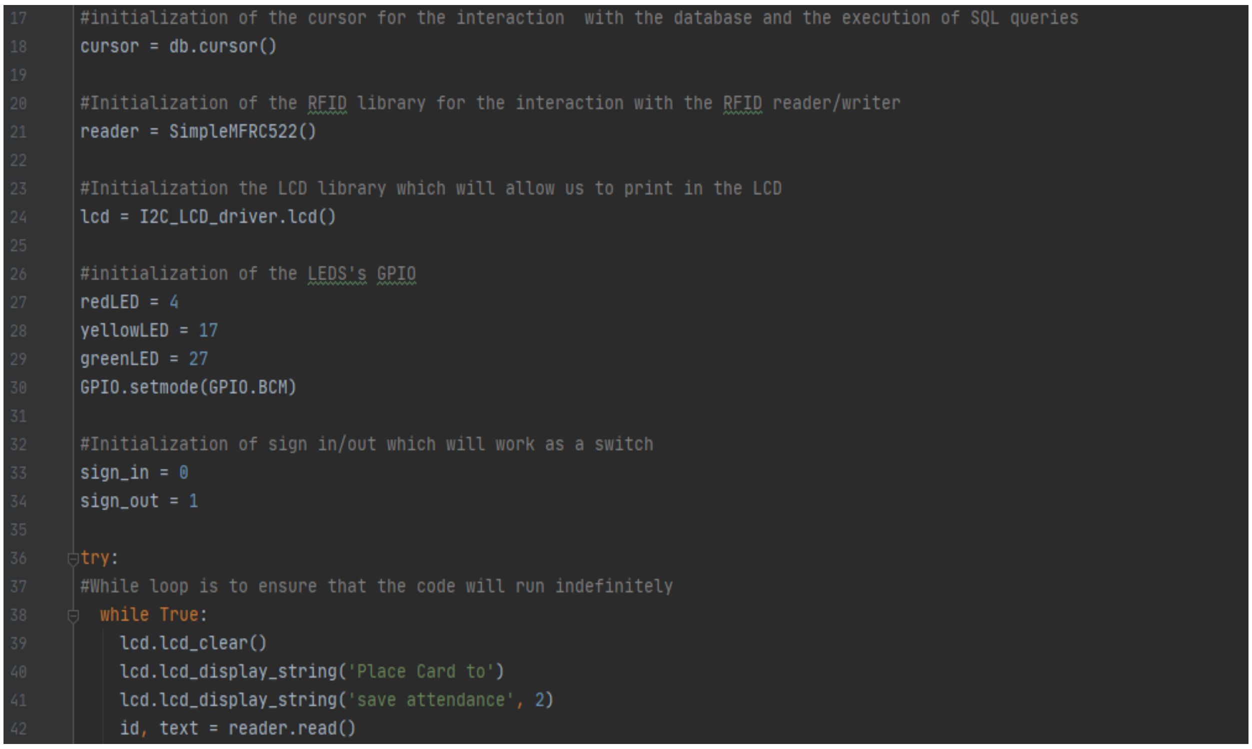
The second file will be the major file of this attendance system, through ”save_attendance.py” file the system will be able to save the current time of when the user signed in or out. The ”users_attendance” table in the database will be then responsible of saving the data captured from the reader, as saving the current time when user signed in the ’clock_in’ column and the time when user sign out in the ‘clock_out’. While the ”total_time_daily” counts the time difference daily between the sign in and sign out time.

To start the system this file needs to be run in the terminal by:

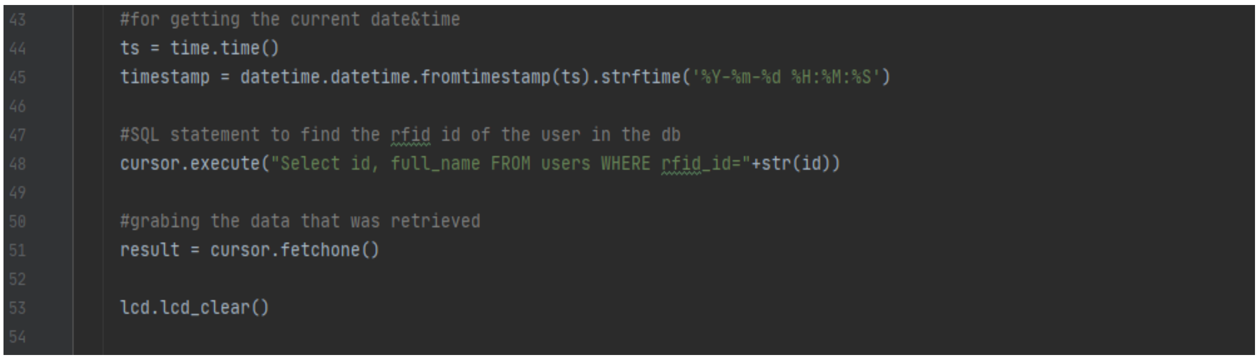
```
sudo python3 ~/attendancesystem/save_attendance.py
```



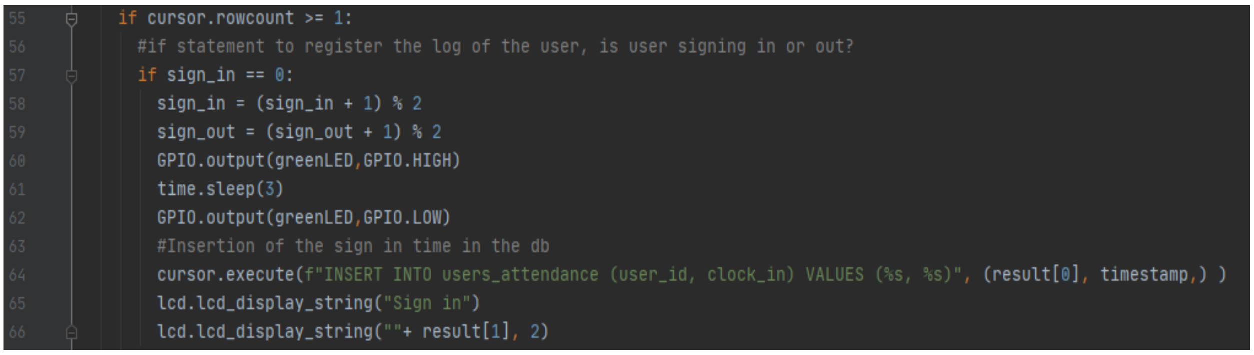
```
#!/usr/bin/env python
#importing libraries needed for this project
import I2C_LCD_driver
import mysql.connector
import time
import datetime
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

#database connection
db = mysql.connector.connect(
  host="localhost",
  user="admin",
  passwd="**********",
  database="attendancesystem"
)
```



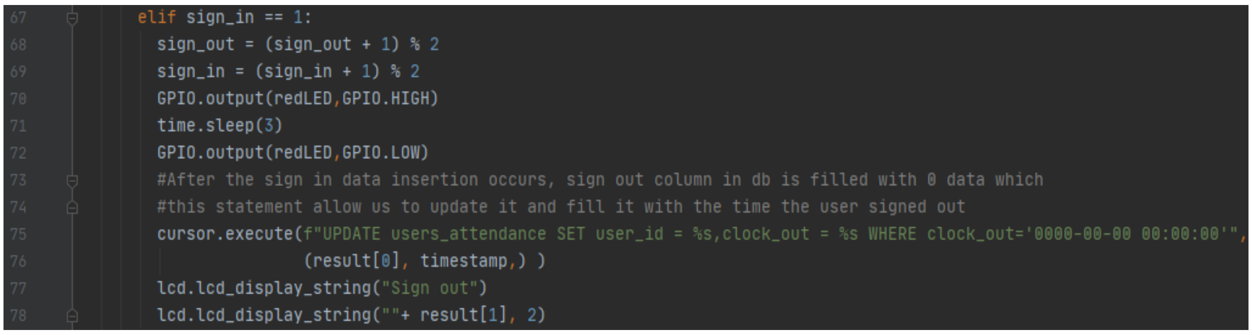
```
#initialization of the cursor for the interaction  with the database and the execution of SQL queries
cursor = db.cursor()

#Initialization of the RFID library for the interaction with the RFID reader/writer
reader = SimpleMFRC522()

#Initialization the LCD library which will allow us to print in the LCD
lcd = I2C_LCD_driver.lcd()

#initialization of the LEDS's GPIO
redLED = 4
yellowLED = 17
greenLED = 27
GPIO.setmode(GPIO.BCM)

#Initialization of sign in/out which will work as a switch
sign_in = 0
sign_out = 1

try:
#While loop is to ensure that the code will run indefinitely
  while True:
    lcd.lcd_clear()
    lcd.lcd_display_string('Place Card to')
    lcd.lcd_display_string('save attendance', 2)
    id, text = reader.read()
```



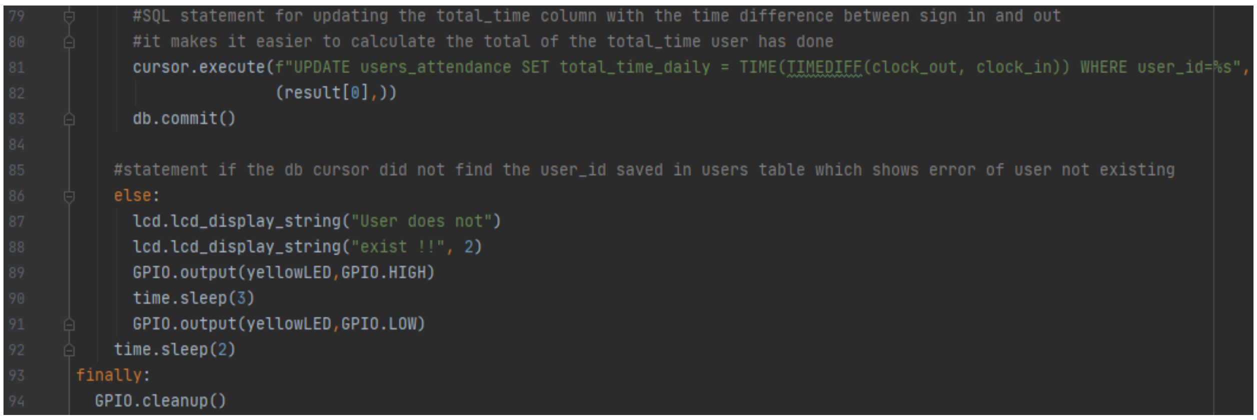
```
    #for getting the current date&time
    ts = time.time()
    timestamp = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')

    #SQL statement to find the rfid id of the user in the db
    cursor.execute("Select id, full_name FROM users WHERE rfid_id="+str(id))

    #grabing the data that was retrieved
    result = cursor.fetchone()

    lcd.lcd_clear()

```



```
    if cursor.rowcount >= 1:
      #if statement to register the log of the user, is user signing in or out?
      if sign_in == 0:
        sign_in = (sign_in + 1) % 2
        sign_out = (sign_out + 1) % 2
        GPIO.output(greenLED,GPIO.HIGH)
        time.sleep(3)
        GPIO.output(greenLED,GPIO.LOW)
        #Insertion of the sign in time in the db
        cursor.execute(f"INSERT INTO users_attendance (user_id, clock_in) VALUES (%s, %s)", (result[0], timestamp,) )
        lcd.lcd_display_string("Sign in")
        lcd.lcd_display_string(""+ result[1], 2)
```



```
    elif sign_in == 1:
      sign_out = (sign_out + 1) % 2
      sign_in = (sign_in + 1) % 2
      GPIO.output(redLED,GPIO.HIGH)
      time.sleep(3)
      GPIO.output(redLED,GPIO.LOW)
      #After the sign in data insertion occurs, sign out column in db is filled with 0 data which
      #this statement allow us to update it and fill it with the time the user signed out
      cursor.execute(f"UPDATE users_attendance SET user_id = %s,clock_out = %s WHERE clock_out='0000-00-00 00:00:00'",
                     (result[0], timestamp,) )
      lcd.lcd_display_string("Sign out")
      lcd.lcd_display_string(""+ result[1], 2)
```



```
      #SQL statement for updating the total_time column with the time difference between sign in and out
      #it makes it easier to calculate the total of the total_time user has done
      cursor.execute(f"UPDATE users_attendance SET total_time_daily = TIME(TIMEDIFF(clock_out, clock_in)) WHERE user_id=%s",
                     (result[0],))
      db.commit()

    #statement if the db cursor did not find the user_id saved in users table which shows error of user not existing
    else:
      lcd.lcd_display_string("User does not")
      lcd.lcd_display_string("exist !!", 2)
      GPIO.output(yellowLED,GPIO.HIGH)
      time.sleep(3)
      GPIO.output(yellowLED,GPIO.LOW)
    time.sleep(2)
finally:
  GPIO.cleanup()
```

Results of this code running in the database will look like:

```
SELECT * FROM users_attendance;
```

```
+----+----------+---------------------+---------------------+------------------+
| id | user_id | clock_in             | clock_out           | total_time_daily |
+----+----------+---------------------+---------------------+------------------+
|  1 |      1   | 2020-04-20 09:00:20 | 2020-04-20 13:15:23 |         04:15:03 |
|  2 |      1   | 2020-04-23 11:23:54 | 2020-04-23 16:50:33 |         05:26:39 |
|  3 |      1   | 2020-04-25 10:12:41 | 2020-04-25 20:10:03 |         09:57:22 |
|  4 |      1   | 2020-04-22 08:12:50 | 2020-04-22 13:40:55 |         05:28:05 |
|  5 |      1   | 2020-04-25 08:00:50 | 2020-04-25 10:00:51 |         02:00:01 |
|  6 |      1   | 2020-04-28 09:47:35 | 2020-04-28 14:55:51 |         05:08:16 |
+----+----------+---------------------+---------------------+------------------+
```

In case system administrator wants to know the total hours that a user performed, the administrator needs to run this SQL statement in the database terminal:

```
SELECT SEC_TO_TIME(SUM(TIME_TO_SEC(clock_out) - TIME_TO_SEC(clock_in))) AS total_hours
FROM users_attendance WHERE user_id = 1;
```

Which will show the sum of the ”total_time_daily” like:

```
+----------------+
| total_hours |
+----------------+
|    32:15:26   |
+--------------- +
```

In case the user forgets to sign in or out from the system, it is the system administrator’s duty to update the columns in the database and that would be by running an SQL statement and saving the changes made into the file.

In case user forgot to sign out from the system:

```
UPDATE users_attendance SET clock_out = ‘????-??-?? ??:??:??’ WHERE user_id = ? ;
```

In case user forgot to sign into the system:

```
INSERT INTO users_attendance (user_id, clock_in) VALUES (‘?’, ‘????-??-?? ??:??:??’) ;
```

When the user forgets to sign in then the system administrator inserts his sign in time to DB. When the user would like to sign out, user needs to put the RFID fob/card twice until the system in the I2C LCD shows sign out message.

Alternatively, the user sign in and out by giving the real times to the system administrator. This last approach will update both columns in the database with the real values.

4 TESTING

Once the development phase is finished, the testing phase will follow. This phase is very important in any system development because through testing we can assure the quality of the final product. Is the final product working perfectly with no bugs and errors? Is the final product as discussed with the customer? Does the final product meet the criteria of the principal product? Is the customer happy with the final product? These are just some questions that must be answered.

It is hard to achieve all those needs in only one version of the system software development. Although, it may seem possible to achieve all what was discussed in the first meetings of the planning phase, once it comes to the development phase, some developers just lack the information of the software should be developed or can it be done at all. Thus, changing the outcome or procedure of the project might need to be considered.

When developing and testing this type of attendance system prototype; everything is working perfectly with no errors or bugs for one user, but once you add another user to the system, it somehow confuses the ”users_attendance” table in the database by filling data of user 2 into data of user 1. If user 1 sign in with his card, the user 2 first action is not sign in but sign out from the system, which declares that the system and database cannot make the difference between the two users. For this, future development is needed to this system so that multi-users will be able to save their attendance with correctly. The best suggestion would be to create a new table for saving the check in and out time for each user created separately, and the table name in the database can have the name of the user.

Also, it might be helpful in the future to have a user interface for this system for the easy readability for any person with no technical background.

5 CONCLUSION

The goal of this thesis was to develop a prototype of an attendance system using Raspberry Pi microcontroller, RFID MFRC522, I2C LED and green, yellow, and red LEDs. Those components would make for the thesis commissioner a full attendance system where the user signs in and out with saving that data read from the RFID reader to the database. In doing so, this has been successfully created to perform with one to multiple users.

Having in mind the initial idea that the rising demand for solutions in today's world regarding employee tracking, using RFID readers have become more consistent. It is important that companies monitor their workers by using the clock in and out method as presented here, instead of employees writing their own hours. Therefore, the tracking of the employees allows the company not only to maintain stability, but also consistency in their work place practices.

On a general note, the stages and planning for the project was not what was anticipated. By starting with the initial idea of creating a user interface for adding users would have meant that the project had started from the wrong step of the development. As a result, the procedures were changed through the course of the project by overcoming such obstacles.

In conclusion, the project fulfilled the needs of a basic attendance system, as this will go further by it being developed further by the commissioner. Through proper code adjustments, revisions, and tests, the product can be developed to its full capabilities. In doing so, this product will eventually be added to the premises of theFIRMA along with future possibilities of using it also at other schools and locations.

As mentioned in the testing phase, the next step of this project would be to add a user interface for the database. Firstly, the current layout is quite difficult for the end user to use, especially for those with a non-technical background, so in the future creating a simpler user interface (UI) is needed. This will include a webpage containing the information regarding users from the database which allows the easy accessibility and reading of each individual user's data separately. Secondly, the system is only a prototype and future development is needed to streamline the performance of multiple users.

REFERENCES

Adams, C. (2011). What is a Mis-Use Case? Retrieved April 7, 2020, at https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/2107/What-is-a-Mis-Use-Case.aspx

Alexander, I. (2003). Use Cases with Hostile Intent. Retrieved April 11, 2020, at https://www.scenarioplus.org.uk/papers/misuse_cases_hostile_intent/misuse_cases_hostile_intent.htm

Altexsoft. (2018). Functional and Nonfunctional Requirements: Specification and Types. Retrieved April 5, 2020, at https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/

AtlasRFIDstore. What is RFID? | The beginner's guide to RFID systems. Retrieved April 6, 2020, at https://www.atlasrfidstore.com/rfid-beginners-guide/

Cohn, M. (2008). Advantages of the "As a user, I want" user story template. Retrieved April 8, 2020, at http://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template

Deligence Technologies. (2018). Attendance System Using MYSQL With Raspberry Pi and RFID-RC522. Retrieved April 20, 2020, at https://www.instructables.com/id/Attendance-System-Using-MYSQL-With-Raspberry-Pi-an/

Emmet. (2019). Build your own Raspberry Pi RFID Attendance system. Retrieved March 12, 2020, at https://pimylifeup.com/raspberry-pi-rfid-attendance-system/

History of RFID. (n.d). Retrieved April 6, 2020, at https://www.globalventurelabels.com/history-of-rfid/

Innovative Architects. (n.d). The seven phases of the system-development life cycle. Retrieved April 9, 2020, at https://www.innovativearchitects.com/KnowledgeCenter/basic-IT-systems/system-development-life-cycle.aspx

Landt, J. (2001). Shrouds of Time, The history of RFID. Retrieved April 10, 2020,at https://web.archive.org/web/20090327005501/http://www.transcore.com/pdf/AIM%20shrouds_of_time.pdf

Last Minute Engineers. (n.d). What is RFID? How It Works? Interface RC522 RFID Module with Arduino. Retrieved 11, 2020, at https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/

Magrassi, P. & Berg, T. (2002). A world of smart objects: The role of auto identification technologies. Retrieved April 12, 2020, at https://www.gartner.com/en/documents/366151/a-world-of-smart-objects-the-role-of-auto-identification-

MATT. (2015). Using I2C Enabled LCD Screens with the Raspberry Pi. Retrieved March 7, 2020, at https://www.raspberrypi-spy.co.uk/2015/05/using-an-i2c-enabled-lcd-screen-with-the-raspberry-pi/

MATT. (2018). Using an I2C OLED Display Module with the Raspberry Pi. Retrieved March 7, 2020 at https://www.raspberrypi-spy.co.uk/2018/04/i2c-oled-display-module-with-raspberry-pi/

Meints, M. (2007). D3.7: A Structured Collection on Information and Literature on Technological and Usability Aspects of Radio Frequency Identification (RFID). Retrieved April 15, 2020 at http://www.fidis.net/resources/deliverables/hightechid/#c1782

Phillips, G. (2017). How does RFID Technology Work? Retrieved April 5, 2020, at https://www.makeuseof.com/tag/technology-explained-how-do-rfid-tags-work/

Raspberry Pi Org. (n.d 1). Downloading Raspbian. Retrieved March 5, 2020, at https://www.raspberrypi.org/downloads/

Raspberry Pi Org. (n.d 2). How to get and install NOOBS. Retrieved March 5, 2020, at https://www.raspberrypi.org/help/noobs-setup/2/

Raspberry Pi Org. (n.d 3). NOOBS installation. Retrieved March 5, 2020, at https://www.raspberrypi.org/documentation/installation/noobs.md

Raspberry Pi Org. (n.d 4). Documentation of Raspberry Pi 4. Retrieved March 6, 2020, at https://www.raspberrypi.org/documentation/usage/gpio/

Rehkopf, M. (n.d). User stories with Examples and Template. Retrieved April 14, 2020, at https://www.atlassian.com/agile/project-management/user-stories

T2informatik. (n.d). User Story. Retrieved April 15, 2020, at https://t2informatik.de/en/smartpedia/user-story/

TechTerms. (2020). System Requirements. Retrieved April 9, 2020, at https://techterms.com/definition/system_requirements

TheFIRMA. (n.d). Services. Retrieved April 5, 2020, at https://thefirma.fi/?page_id=749

Thepihut. (2015). Turning on an LED with your Raspberry Pi’s GPIO Pins. Retrieved March 10, 2020, at https://thepihut.com/blogs/raspberry-pi-tutorials/27968772-turning-on-an-led-with-your-raspberry-pis-gpio-pins