

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

Student Attendance Control

Version 0.5approved

Prepared by Arif Ahmet Balik, Huseyin  
Bayraktar, Omer Faruk Can

Istanbul Arel University

May 8, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Purpose . . . . .	5
1.2	Document Conventions . . . . .	5
1.3	Intended Audience and Reading Suggestions . . . . .	5
1.4	Project Scope . . . . .	6
1.5	References . . . . .	6
<b>2</b>	<b>Overall Description</b>	<b>6</b>
2.1	Product Perspective . . . . .	6
2.2	Product Features . . . . .	6
2.2.1	Web Application . . . . .	6
2.2.2	Device . . . . .	6
2.3	User Classes and Characteristics . . . . .	7
2.4	Operating Environment . . . . .	7
2.5	Design and Implementation Constraints . . . . .	7
2.6	User Documentation . . . . .	8
2.7	Assumptions and Dependencies . . . . .	8
<b>3</b>	<b>System Features</b>	<b>8</b>
3.1	Data and application hosting on Amazon web servers . . . . .	8
3.1.1	Description and Priority . . . . .	8
3.1.2	Stimulus/Response Sequences . . . . .	8
3.2	Add, remove or edit class, student and user information . . . . .	9
3.2.1	Description and Priority . . . . .	9
3.2.2	Stimulus/Response Sequences . . . . .	9
3.3	Generate, print and export reports . . . . .	10
3.3.1	Description and Priority . . . . .	10
3.3.2	Stimulus/Response Sequences . . . . .	10
3.4	Monitor students individually . . . . .	10
3.4.1	Description and Priority . . . . .	10
3.4.2	Stimulus/Response Sequences . . . . .	10
3.5	Reading RFID cards . . . . .	11
3.5.1	Description and Priority . . . . .	11
3.5.2	Stimulus/Response Sequences . . . . .	11
3.6	LED and sound indicators . . . . .	11
3.6.1	Description and Priority . . . . .	11
3.6.2	Stimulus/Response Sequences . . . . .	11

3.7	More than one week with battery life . . . . .	11
3.7.1	Description and Priority . . . . .	11
3.7.2	Stimulus/Response Sequences . . . . .	12
3.8	Functional and Non-Functional Requirements . . . . .	12
3.9	System Specifications . . . . .	13
3.10	Block Diagrams . . . . .	15
3.11	Use Cases . . . . .	16
<b>4</b>	<b>External Interface Requirements</b>	<b>21</b>
4.1	User Interfaces . . . . .	21
4.2	Hardware Interfaces . . . . .	21
4.3	Software Interfaces . . . . .	22
4.4	Communications Interfaces . . . . .	23
<b>5</b>	<b>Appendix</b>	<b>24</b>
5.1	Appendix A: To Be Determined List . . . . .	24
5.2	Appendix B: Lists . . . . .	24
	<b>List of Figures</b>	<b>24</b>
	<b>List of Tables</b>	<b>25</b>

# Revision History

Name	Date	Reason For Changes	Version
1	13.04.2019	Initial	0.1
2	17.04.2019	Adding use case tables and functional requirements	0.2
3	21.04.2019	Update functional requirenments and chapters redesigned	0.3
4	23.04.2019	Specifications added	0.4
5	05.05.2019	Block Diagrams added	0.5

Table 0.1: Revision History

# 1 Introduction

## 1.1 Purpose

The product which will be covered throughout the document is a student attendance control system using RFID technology. This document includes all the specification for prototype versions. The document covers SRS for both embedded and web platform.

## 1.2 Document Conventions

When the name 'device' is mentioned, it means document points to the hardware that will read the RFID card, but in some cases device point to something else, in that case reader is notified.

When 'report' is mentioned in the report, it means SRS refers to attendance lists or information of an individual student.

<b>RFID</b>	Radio Frequency Identification
<b>SRS</b>	Software Requirements Spesification
<b>CMSIS</b>	Cortex Microcontroller Software Interface Standart
<b>HAL</b>	Hardware Abstraction Library
<b>RTC</b>	Real Time Clock
<b>RGB</b>	Red, green, blue
<b>SoC</b>	System On Chip
<b>PCB</b>	Printed Circuit Board
<b>PWM</b>	Pulse Width Modulation
<b>DAC</b>	Digital to Analog Converter
<b>ADC</b>	Analog to Digital Converter
<b>IC</b>	Integrated Circuit
<b>TH</b>	Through Hole

Table 1.1: Glossary

## 1.3 Intended Audience and Reading Suggestions

This document's reader is the instructor and as well as the students of the Software Engineering class. Document begins with introducing the project and its internals, challenges and design considerations.

## 1.4 Project Scope

Attendance is a job that is still being done by the instructor manually. This project aims to change this with a low cost device and help university to switch a fully autonomous attendance monitoring system with great benefits. This project will also inspire next generation under graduate students to contribute their environment and making it better.

## 1.5 References

*ARM CMSIS* <https://www.arm.com/why-arm/technologies/cmsis>

*HAL Libraries* [https://www.st.com/content/st\\_com/en/products/embedded-software](https://www.st.com/content/st_com/en/products/embedded-software)

*RFID* [https://en.wikipedia.org/wiki/Radio-frequency\\_identification](https://en.wikipedia.org/wiki/Radio-frequency_identification)

*MFRC522 Datasheet* <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>

*Software Engineering, Ian Somerville, 6th Edition*

# 2 Overall Description

## 2.1 Product Perspective

Arel University already uses an RFID system, this self-contained product will extend the power of this system into the classrooms. Product contains a device and a database system that will keep track student movements, and will generate reports.

## 2.2 Product Features

### 2.2.1 Web Application

- FE-1: Data and application hosting on Amazon web servers
- FE-2: Add, remove or edit class, student and user information
- FE-3: Generate, print and export reports
- FE-4: Monitor students individually

### 2.2.2 Device

- FE-1: Reading RFID cards
- FE-2: LED and sound indicators
- FE-3: More than one week with battery life

## 2.3 User Classes and Characteristics

Instructor (favored)	The Instructor is the person or people who monitors attendances of students individually or class based. The Instructor uses web server to track student's or class's information. Instructor can also create or edit new classes or students.
Student (favored)	The students is the main participant in the system. Students use the device and their student cards to log themselves in to the class on their schedule. Student can not log in to web server in the prototype version.
University staff	University staff is responsible for data entry into the system, such as classes and students.

Table 2.1: User Classes and Characteristics

## 2.4 Operating Environment

OE-1:	System is dependent on geographical areas. Timezone shall be set before operation.
OE-2:	Web application shall use latest web browsers.
OE-3:	Any data generated online shall be stored on the server with MySQL database systems.
OE-4:	.NET libraries will be needed for web server to run
OE-5:	RFID reader (a.k.a The Device) needs an external device that will provide internet connection to process any data.
OE-6:	The device should be charged or plugged into a power source to carry any kind of operation.

Table 2.2: Operating Environments

## 2.5 Design and Implementation Constraints

Since device contains a microcontroller, it is hard to store large amounts of data on it, options such as SD card or flash memory means more cost and complexity to the product. And then again WiFi protocol is a big issue in embedded systems since it also requires large amount of data structures and complex communication protocols, thus in the development process it might be a problem to communicate to a web server, fluently and securely.

Also device can be designed with plug-n-play solutions such as Arduino, but it is a

'plan b' because first the project must be professional and second it has to be easily mass produceable.

Web server is funded by the owner of the project which will be hard to maintain by them, therefore money is another constraint.

## 2.6 User Documentation

The device and web server will be provided with a 'Quick Start Guide' and with hands-on training, also device support will be available until students graduate.

Also for developers, project will release a 'Communication Protocol Guide'.

## 2.7 Assumptions and Dependencies

AS-1:	No more than 100 bytes of data sent at one http request
AS-2:	No more than 10.000 page views monthly
AS-3:	Users has PDF file reader either installed on the device they use or on the web browser.
AS-4:	Internet is accessible in every classroom.
DE-1:	WiFi Router for the device
DE-2:	The device should be charged or plugged into a power source to carry any kind of operation.

Table 2.3: Assumptions and Dependencies

# 3 System Features

## 3.1 Data and application hosting on Amazon web servers

### 3.1.1 Description and Priority

Application will need to be hosted on Amazon Web Services

Priority: high.

### 3.1.2 Stimulus/Response Sequences

Not applicable.



## 3.2 Add, remove or edit class, student and user information

### 3.2.1 Description and Priority

Instructor or university staff (a.k.a user) needs to be able to add new students every year and also add new classes and new users to access the system. These features will be accessible on the web application after log in operation.

Priority: high.

### 3.2.2 Stimulus/Response Sequences

Stimulus: User requests to add or edit information of a student

Response: If user requests to edit, then user shall provide student number or any other parameter that is on the DB, then system recalls the data from the DB and fills and displays a form, otherwise system displays an empty form for following informations to be filled by the user:

1. Student No
2. Name & Last Name
3. Department
4. RFID No

Stimulus: User requests to add or edit information of a class

Response: If user requests to edit, then user shall provide class number or any other parameter that is on the DB, then system recalls the data from the DB and fills and displays a form, otherwise system displays an empty form for following informations to be filled by the user:

1. Class No
2. Name
3. Day of Week
4. Start
5. End
6. Students

Stimulus: User requests to add or edit information of a user

Response: If user requests to edit, then user shall provide username, then system recalls the data from the DB and fills and displays a form, otherwise system displays an empty form for following informations to be filled by the user:

1. User Name
2. Password

Stimulus: User requests to delete a student, class or user

Response: User types the name or id of student, class or user and clicks delete button. System deletes the record from DB

### 3.3 Generate, print and export reports

#### 3.3.1 Description and Priority

Instructor or university staff (a.k.a user) needs to be able to create reports for classes, those are the reports that will show which class has how many attended students in the conventional attendance lists format. These features will be accessible on the web application after log in operation.

Priority: high.

#### 3.3.2 Stimulus/Response Sequences

Stimulus: User requests to generate attendance list in either form of a directly printable file by clicking 'print' button or downloadable PDF file by clicking 'export' button after entering name or id of the class in the 'generate report' section.

Response: System copies all the data related to attendance into a template and either opens a print screen or creates a downloadable link and saves the file in the DB for later use

Stimulus: User requests to view an existing attendance list and wants it in either form of a directly printable file by clicking 'print' button or downloadable PDF file by clicking 'export' button after entering name or id of the class in the 'view reports' section.

Response: System searches for files related to given class and list every one of them in an order and shows an option for both printing and exporting for each one of them

### 3.4 Monitor students individually

#### 3.4.1 Description and Priority

Instructor or university staff (a.k.a user) needs to be able to track each student in detail individually, this informations will show which student attended which class at which time and how often. These features will be accessible on the web application after log in operation.

Priority: normal.

#### 3.4.2 Stimulus/Response Sequences

Stimulus: User requests to view student attendance information by entering the student id or name of the student in 'student  $i$  view' section. Optionally user can export or print those reports

Response: System copies all the data related to attendance into a template and displays it in the web page, system adds two button at the bottom of the page for exporting and printing the information

## **3.5 Reading RFID cards**

### **3.5.1 Description and Priority**

Device shall be able to read student cards. The timezone shall be set and on/off shall be opened before operation.

Priority: high.

### **3.5.2 Stimulus/Response Sequences**

Stimulus: Student brings his/her student card near to the device

Response: Device reads and sends the ID of the card to the web server and turns it LEDs and sound on.

## **3.6 LED and sound indicators**

### **3.6.1 Description and Priority**

Device shall inform the student about the process that is ongoing.

Priority: normal.

### **3.6.2 Stimulus/Response Sequences**

Stimulus: Student brings his/her student card near to the device

Response: Device turns on an RGB led depending on the status of the process and plays a certain tone. Following list shows the possible status and color codes:

1. Success - Green
2. Unknown Card - Red
3. No Internet Connection - Blue
4. Web Server Not Responding - White
5. Unknown Problem (HALT) - Flashing red
6. Low Battery - Flashing white

## **3.7 More than one week with battery life**

### **3.7.1 Description and Priority**

Device shall keep operational without a power plug for at least one week.

Priority: normal.

### 3.7.2 Stimulus/Response Sequences

Not applicable.

## 3.8 Functional and Non-Functional Requirements

### Functional Requirements

- FR-1: Device shall distinguish students by their student card
- FR-2: Device shall inform its environment by sound and indication LEDs when it detects an RFID card.
- FR-3: Device shall work with WiFi and stand alone when there is no WiFi present
- FR-4: Device shall establish a secure connection with web server and send data through HTTP
- FR-5: Device shall read and generate JSON format
- FR-6: Device shall maintain time information within the system with an internal backup battery
- FR-7: When on battery device shall monitor its battery status and inform its environment when battery is running out.
- FR-8: Web server shall have a DB to keep record students, classes and other things.
- FR-9: Web server shall generate attendance reports individually for every class that is defined in the system.
- FR-10: Web server shall generate detailed report for every student
- FR-11: Web server shall be only accessible to instructor and staff of university with a log in page.
- FR-12: Web server shall maintain its databases locally on the server, that is it shall not be remote to web server.

### Non-Functional Product Requirements

- NFP-1: The time between reading an RFID card and processing it in web server shall be less than 5 seconds.
- NFP-2: It shall be possible to store every data that has been sent in 100 bytes long variable.
- NFP-3: Web server shall be usable by every device who knows the keys and communication protocols.
- NFP-4: Device shall establish a communication channel with %95 percent success rate.
- NFP-5: Device shall be affordable enough.
- NFP-6: Device shall have small form factor

### Non-Functional Organisational Requirements

- NFO-1: First prototype shall be operational before final exams, that is in Arel University.

- NFO-2: Documentation of any kind shall be prepared with great care.
- NFO-3: Standards for any kind of document shall be compatible with the book 'Software Engineering' which is used in the class.
- NFO-4: Device shall be delivered with 1 hour training
- NFO-5: Every option in web application shall be accessible by the instructor with no more than 3 clicks.

### Non-Functional External Requirements

- NFE-1: The project shall not disclose any personnel information that is stored in the databases.

## 3.9 System Specifications

### Web Server

- Data Storage Platform: SQL Server 2019
- Data Query Language: T-SQL
- Web Service: XML
- Language: ASP.NET
- Data Encryption: Hash

### The Device

- MCU: ST STM32F0
- WiFi SoC: Espressif ESP8266
- MIFARE RFID Front End: NXP MFRC522
- RTC: Dallas Semi. DS1307
- Battery: 7.4V 2C Lithium Polymer
- Interface: 5mm TH RGB led and Buzzer.

The device uses RFID technology to meet with *FR-1*. Firstly project needs a database so it can match the id inside an RFID card to a student, for this SQL Server 2019 will be used (*FR-8* and *FR-12*). This database containing students RFID ID's and student informations will be crated manually by the project owners based on Appendix A.

Device will create an electrical field around it and when a card approaches it will energize the card and read the content of card with MFRC522 IC (According to datasheet IC is capable to meet with *NFP-1*) and format it with JSON and Hash encryption along with time information using DS1307 and send it (this mets with *NFP-2* see 4.3 for more detail) to web servers with HTTP requests (every device can send HTTP request so *NFP-2* is met) using ESP8266 SoC (*FR-3*, *FR-4*, *FR-5* and *FR-6*), then web server will compare that data with its database to see if there is a match with using ASP.NET *Data* libraries and SQL Sever DBs. Then web server will send status to the device using again ASP.NET and HTTP requests and device will act accordingly (*FR-2*).

Device also will monitor its battery status with ADC modules of MCU to inform user when its about to run out (*FR-7*).

There will be a login page at the web server again made with ASP.NET, no feature will be available until user logs in. Every data of user will be recorded on DB (*FR-11*).

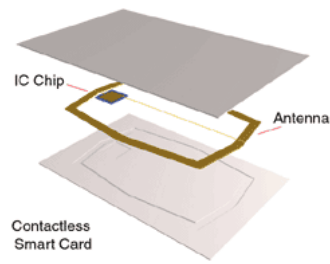
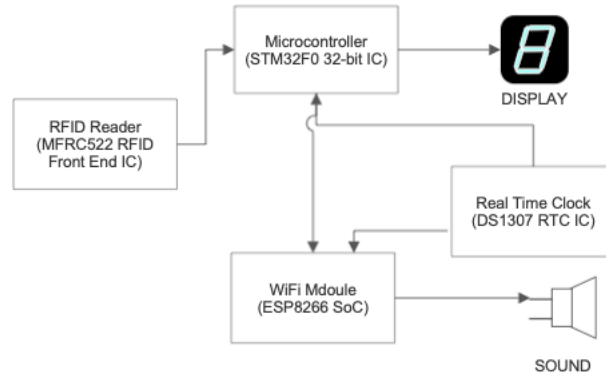


Figure 3.1: Contactless Smart Card



Figure 3.2: Typical Buzzer

### 3.10 Block Diagrams



<b>BLOCK DIAGRAM - SAC</b>	DRAWN BY	CHECKED	DATE	SCALE	SHEET NO.
	Arif A. Balik		3/5/19	Not to Scale	1/2

Figure 3.3: Overall Block Diagram for The Device

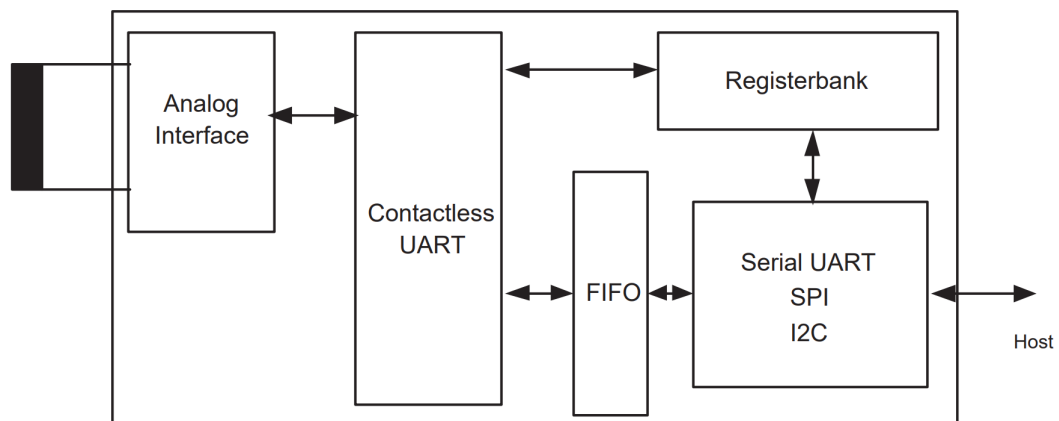


Figure 3.4: Simplified MFRC522 IC Diagram

### 3.11 Use Cases

The following use cases gives a general scenario on how student and instructor may interact with the project. If reader seeks more detailed information, please check functional requirements and system features.

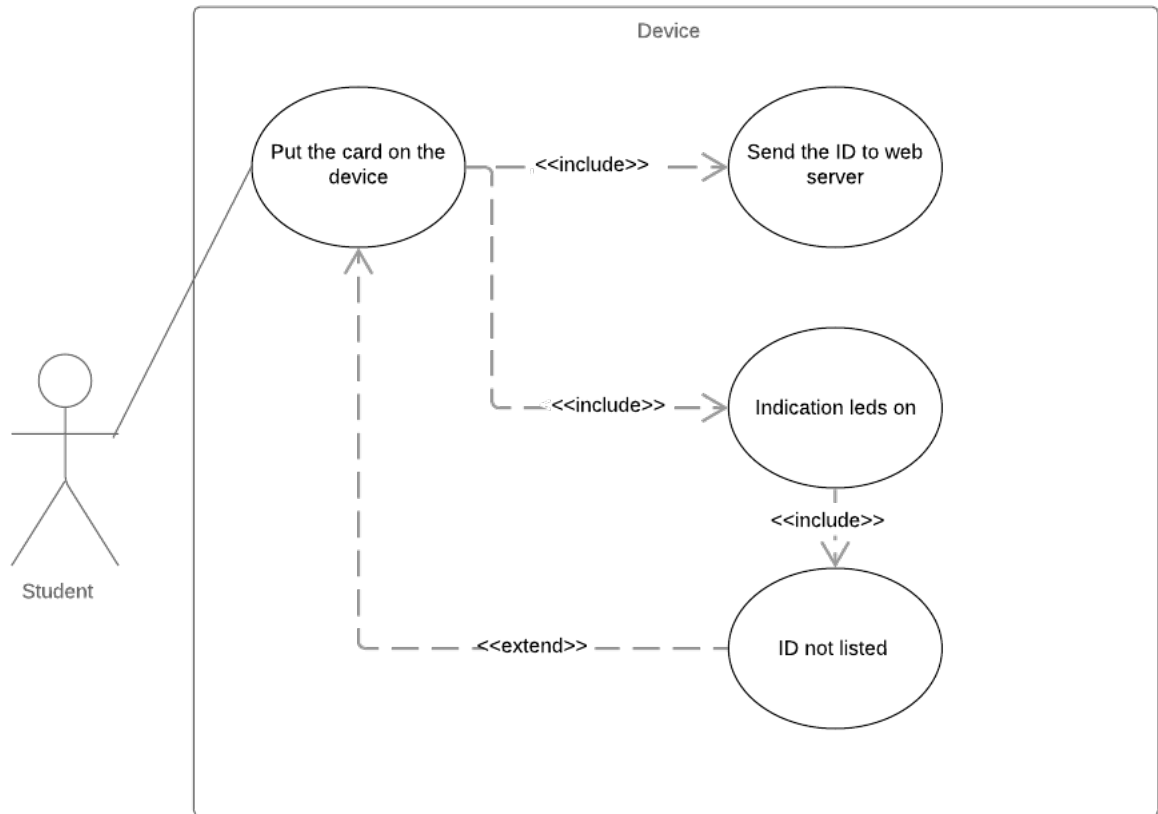


Figure 3.5: Student and Device Interaction



<b>Name</b>	Put the card on the device
<b>Actors</b>	Student
<b>Description</b>	Student use the card to log in to the class
<b>Succesful Completion</b>	<ol style="list-style-type: none"> <li>1. Device sends the ID to server</li> <li>2. Server looks for a match</li> <li>3. Server returns success</li> <li>4. Indication LEDs turn on</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. Device sends the ID to server</li> <li>2. Server looks for a match</li> <li>3. Server returns failure</li> <li>4. Indication LEDs turn on</li> </ol>
<b>Pre-condition</b>	Student must have a valid RFID card
<b>Post-Conditions</b>	None
<b>Assumptions</b>	None

Table 3.1: Scenario : Put the card on the device

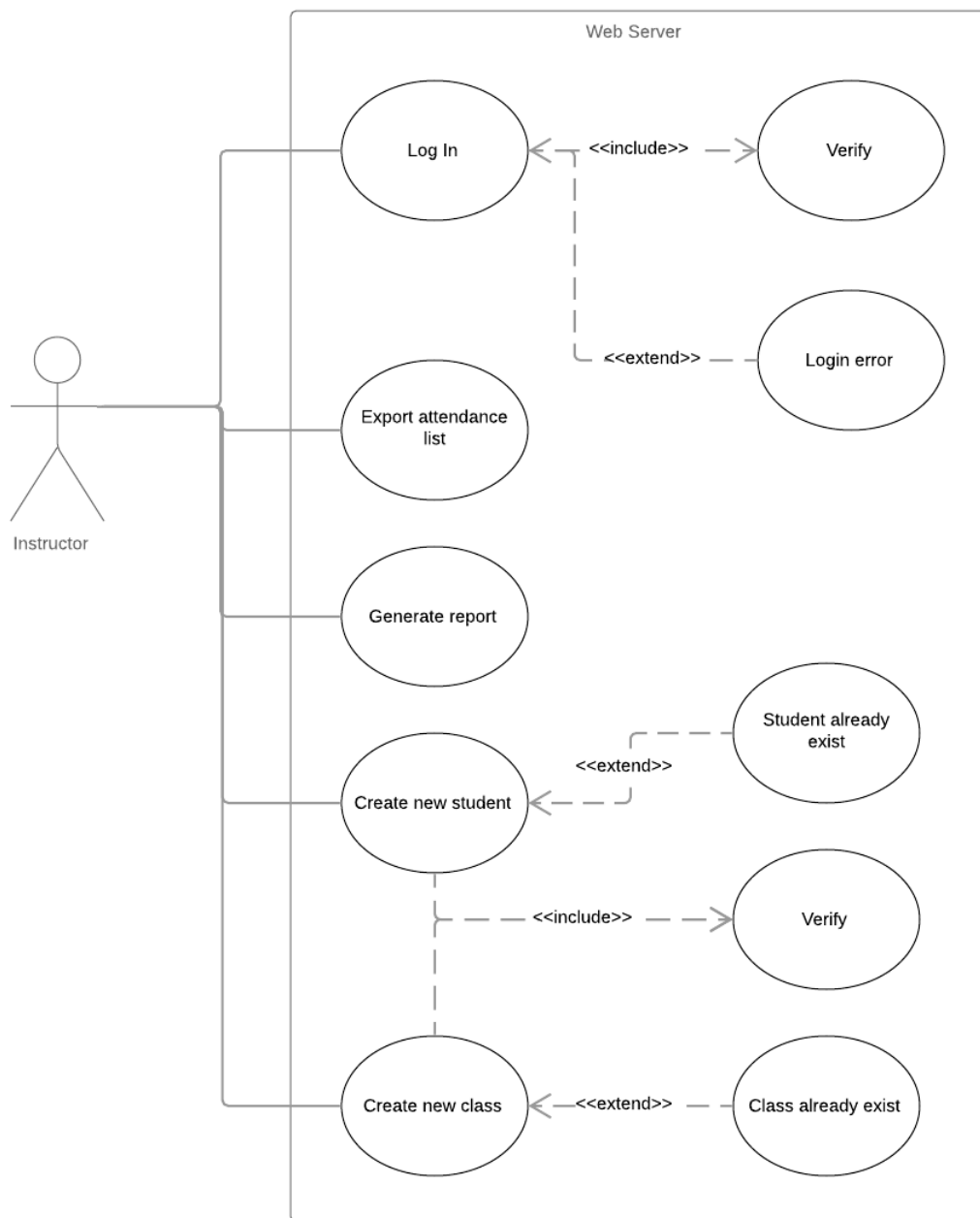


Figure 3.6: Instructor and Web Server Interaction

<b>Name</b>	Log in
<b>Actors</b>	Instructor
<b>Description</b>	Instructor uses its id and password to log in the web server interface
<b>Succesful Completion</b>	<ol style="list-style-type: none"> <li>1. Instructor types id and password</li> <li>2. Server looks for a match in the database</li> <li>3. Server finds a match</li> <li>4. Server starts the session</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. Instructor types id and password</li> <li>2. Server looks for a match in the database</li> <li>3. Server can not find a match</li> <li>4. Server prints an error message</li> </ol>
<b>Pre-condition</b>	Instructor should have a computer and internet connection
<b>Post-Conditions</b>	None
<b>Assumptions</b>	Instructor has id and password

Table 3.2: Scenario : Log in

<b>Name</b>	Export Attendance List
<b>Actors</b>	Instructor
<b>Description</b>	Instructor uses web server interface to export a list of students who attended to class
<b>Succesful Completion</b>	<ol style="list-style-type: none"> <li>1. Instructor selects the class in the main menu</li> <li>2. Instructor uses the button located in the main page named 'export'</li> <li>3. Server collects the data and creates a .pdf file</li> <li>4. Instructor downloads the file</li> </ol>
<b>Alternative</b>	None
<b>Pre-condition</b>	Instructor should have a computer and internet connection
<b>Post-Conditions</b>	None
<b>Assumptions</b>	Instructor is logged in

Table 3.3: Scenario : Export Attendance List

<b>Name</b>	Generate Report
<b>Actors</b>	Instructor
<b>Description</b>	Instructor uses web server interface to generate statistics
<b>Succesful Completion</b>	<ol style="list-style-type: none"> <li>1. Instructor selects the class in the main menu</li> <li>2. Instructor uses the button located in the main page named 'generate report'</li> <li>3. Server collects the data and generates statistic as a .pdf file</li> <li>4. Instructor downloads the file</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. Instructor selects the class in the main menu</li> <li>2. Instructor uses the button located in the main page named 'generate report'</li> <li>'3. Server can not find enough data to generate statistic as a .pdf file</li> <li>4. Server prints error message</li> </ol>
<b>Pre-condition</b>	Instructor should have a computer and internet connection
<b>Post-Conditions</b>	None
<b>Assumptions</b>	Instructor is logged in

Table 3.4: Scenario : Generate Report

<b>Name</b>	Create New Student
<b>Actors</b>	Instructor
<b>Description</b>	Instructor uses web server interface to add new student into system
<b>Succesful Completion</b>	<ol style="list-style-type: none"> <li>1. Instructor clicks 'add student' button in the main page</li> <li>2. Instructor fills the information required, such as name, student id, card id, department and presses 'save' button.</li> <li>3. Server checks if student already exist</li> <li>4. Student dont exist, server records the data in DB</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. Instructor clicks 'add student' button in the main page</li> <li>2. Instructor fills the information required, such as name, student id, card id, department and presses 'save' button.</li> <li>3. Server checks if student already exist</li> <li>4. Student already exist, server prints error message</li> </ol>
<b>Pre-condition</b>	Instructor should have a computer and internet connection
<b>Post-Conditions</b>	None
<b>Assumptions</b>	Instructor is logged in

Table 3.5: Scenario : Create New Student

<b>Name</b>	Create New Class
<b>Actors</b>	Instructor
<b>Description</b>	Instructor uses web server interface to add new class into system
<b>Succesful Completion</b>	<ol style="list-style-type: none"> <li>1. Instructor clicks 'add class' button in the main page</li> <li>2. Instructor fills the information required, such as name, department, students and presses 'save' button.</li> <li>3. Server checks if class already exist</li> <li>4. Class dont exist, server records the data in DB</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. Instructor clicks 'add class' button in the main page</li> <li>2. Instructor fills the information required, such as name, department, students and presses 'save' button.</li> <li>3. Server checks if class already exist</li> <li>4. Class already exist, server prints an error message</li> </ol>
<b>Pre-condition</b>	Instructor should have a computer and internet connection
<b>Post-Conditions</b>	None
<b>Assumptions</b>	Instructor is logged in

Table 3.6: Scenario : Create New Class

## 4 External Interface Requirements

### 4.1 User Interfaces

UI-1: Web application shall allow user to generate reports using mouse alone.

UI-2: All features of web application shall be accessible trough one page after successful log in.

### 4.2 Hardware Interfaces

HI-1: Student shall be able to initiate the process of attendance using the device with their student cards alone.

HI-2: Device shall inform the student about the attendance process in case of either success or failure.

HI-3: Device shall warn the instructor before battery runs out.

Also following figure shows general description of hardware of project.

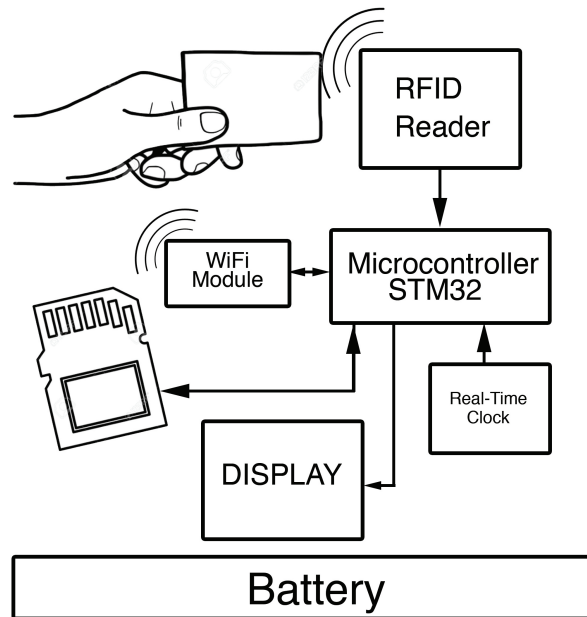


Figure 4.1: Internal modules of hardware

### 4.3 Software Interfaces

#### SI-1: Web Server

SI-1.1: Web server shall communicate with the device

SI-1.2: Upon successful identification web server shall return success to the device

SI-1.3: Web server shall control every log in operation and print a related messages according to who logged in

SI-1.4: Web server shall handle every document to be created

SI-1.5: Web server shall store every process on the MySQL DB and inform the user about status of process

#### SI-2: Database - The system shall communicate with the database for following operations:

SI-2.1: For every log in operation, checking whether user exists or not, and password is correct.

SI-2.2: To store attendance information real time.

SI-2.3: To add, delete or edit students, classes and users

SI-2.4: To generate reports

#### SI-3: Device

SI-3.1: Device shall communicate with the web server

SI-3.2: Device shall send every RFID card it detects along with other informa-

tions as the following JSON format

```
{
  "Date": "13.04.2019 12:20:48 ",
  "ID": 18383400,
  "Key": 12345677890
}
```

SI-3.3: Device shall listen server after sending any data, to inform student whether operation is successful or not.

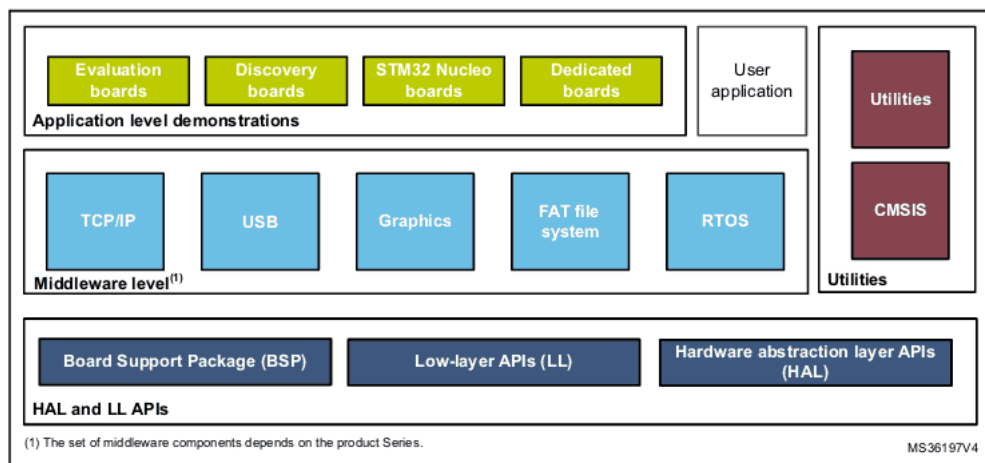


Figure 4.2: Software layers for microcontroller

## 4.4 Communications Interfaces

UI-1: Web application shall send monthly reports automatically through email.

UI-2: Web application shall send an email for every revision or update on the software or terms.

# 5 Appendix

## 5.1 Appendix A: To Be Determined List

- How to communicate bidirectionally
- Solve the problem when student has courses which are overlapping
- Test the DB structre

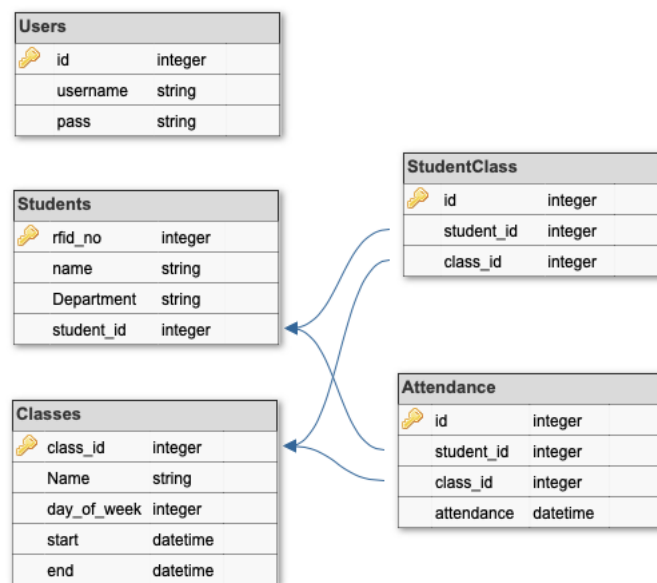


Figure 5.1: DB Structure

## 5.2 Appendix B: Lists

### List of Figures

3.1	Contactless Smart Card . . . . .	14
3.2	Typical Buzzer . . . . .	14
3.3	Overall Block Diagram for The Device . . . . .	15
3.4	Simplified MFRC522 IC Diagram . . . . .	15
3.5	Student and Device Interaction . . . . .	16



3.6	Instructor and Web Server Interaction . . . . .	18
4.1	Internal modules of hardware . . . . .	22
4.2	Software layers for microcontroller . . . . .	23
5.1	DB Structure . . . . .	24

## List of Tables

0.1	Revision History . . . . .	4
1.1	Glossary . . . . .	5
2.1	User Classes and Characteristics . . . . .	7
2.2	Operating Environments . . . . .	7
2.3	Assumptions and Dependencies . . . . .	8
3.1	Scenario : Put the card on the device . . . . .	17
3.2	Scenario : Log in . . . . .	19
3.3	Scenario : Export Attendance List . . . . .	19
3.4	Scenario : Generate Report . . . . .	20
3.5	Scenario : Create New Student . . . . .	20
3.6	Scenario : Create New Class . . . . .	21