SOFTWARE REQUIREMENTS SPECIFICATION

for

Student Attendance Control

Version 0.1 approved

Prepared by Arif Ahmet Balik, Huseyin Bayraktar, Omer Can

Istanbul Arel University

April 17, 2019

Contents

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

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 version 0.1, that is prototype. The document covers SRS for both embedded and web platform.

1.2 Glossary

RFID	Radio Frequency Identification
SRS	Software Requirements Spesification
CMSIS	Cortex Microcontroller Software Interface Standart
HAL	Hardware Abstraction Library
RTC	Real Time Clock

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
RFID https://en.wikipedia.org/wiki/Radio-frequency_identification

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 Functions

Web platform of the product expected to have following functions;

- User Friendly Web Interface
- Adding or removing students with just a few clicks
- Extensive reporting
- Secure Databses

and for the hardware;

- WiFi and SD Card capability
- LED, LCD and sound indicator
- More than one week with battery
- Small form factor
- Cheap

2.3 Operating Environment

Product will operate in a typical classroom and does not need any additional hardware or software except the device itself and a WiFi connection. Software will run on a microcontroller and will talk to a web server which will be located in one of the major providers such as Amazon or Google with MySQL database.

Hardware contains ARM Cortex-M architecture and requires CMSIS to run a software on it. Web server will be written with ASP.NET and databases are managed with SQL.

2.4 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.

2.5 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'.

2.6 Assumptions and Dependencies

It is still unknown that device can communicate with the web server bidirectionally and sustainably. This is still an assumption in the early phase of prototyping, developers will test and update the SRS accordingly.

Also it is assumed that both processes (web and embedded development) will go fluently and at the same time, any bottleneck can cause project to fail due to the short deadline.

3 External Interface Requirements

3.1 User Interfaces

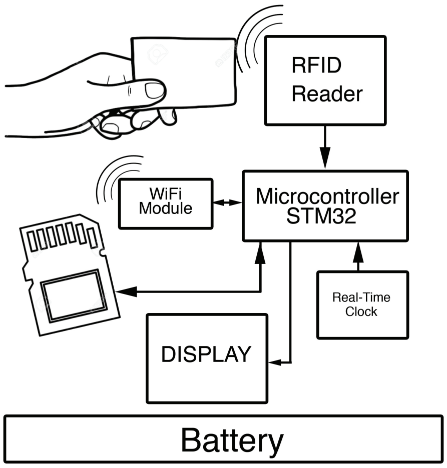
Product has two interfaces. One is device (hardware) and web server.

There will be indication lights on the device to show whether card is accepted or not (also there will be optional LCD). On the web server there will be a page to create classes and students and a page for report generation. Those pages are expected to have very simple and dummy proof.

3.2 Hardware Interfaces

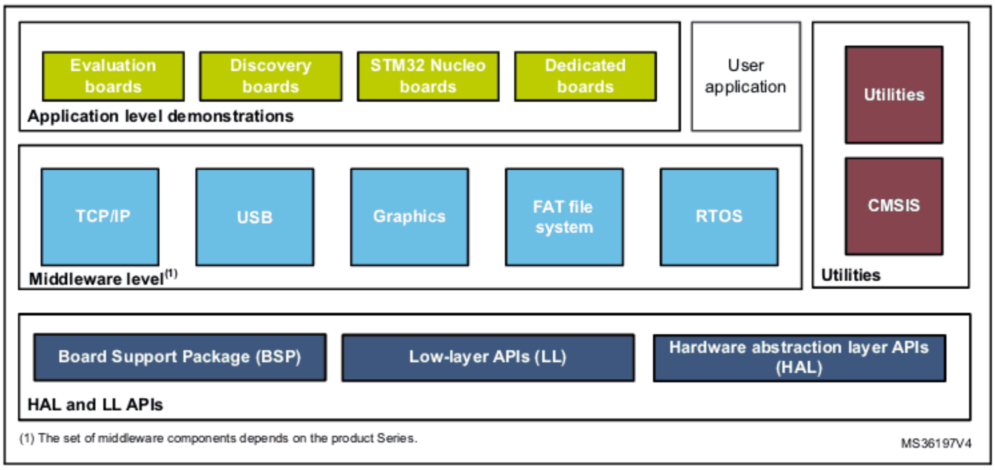
The only interaction between device and web software is communication via HTTP.

Also following figure shows general description of hardware of project.



3.3 Software Interfaces

There are two different platforms the softwares will run on. One is hardware which contains nothing but bare-metal C, but it is worth to mention, developers will use standart libraries that vendors of microcontrollers will provide, such as HAL for STM32 microcontrollers, RFID libraries etc.



Web server will run on Windows and will be written in .NET 4.7. And databases will be on MySQL.

3.4 Communications Interfaces

One major required communication interface needed for the project is a WiFi connection for the device, it can not operate otherwise. Also students has to use ID cards which supports RFID protocol.

4 System Features

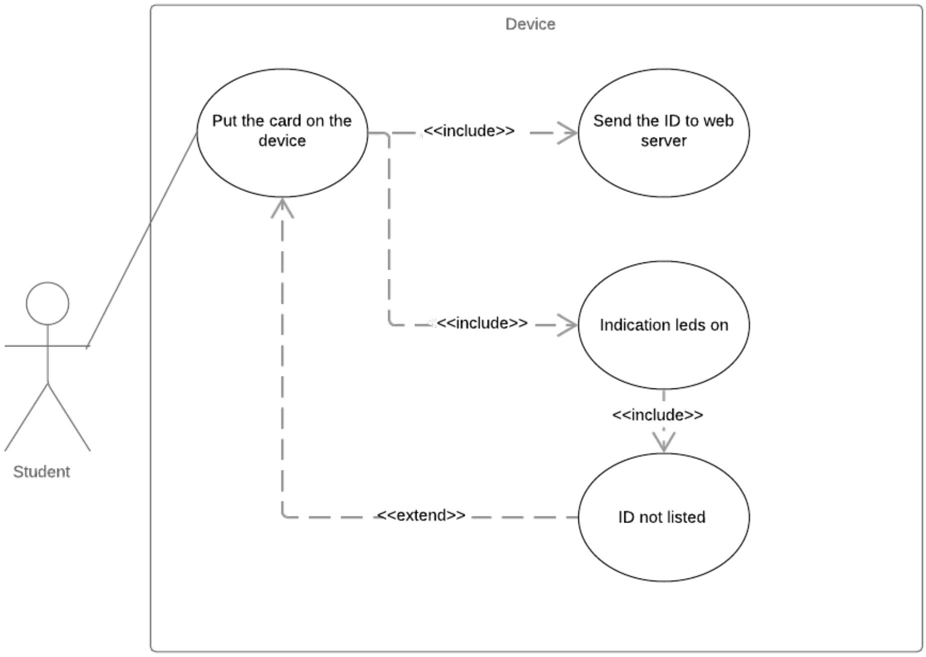
4.1 Tracking Individual Students

4.1.1 Description and Priority

Maybe the most important feature is the tracking ability. In conventional method, it is very hard to track the info for every student, but with this device it will be possible to log every movement of the student in university that is related to attendance. When student entered the class, how often student attends a preticular class etc.

4.1.2 Stimulus/Response Sequences

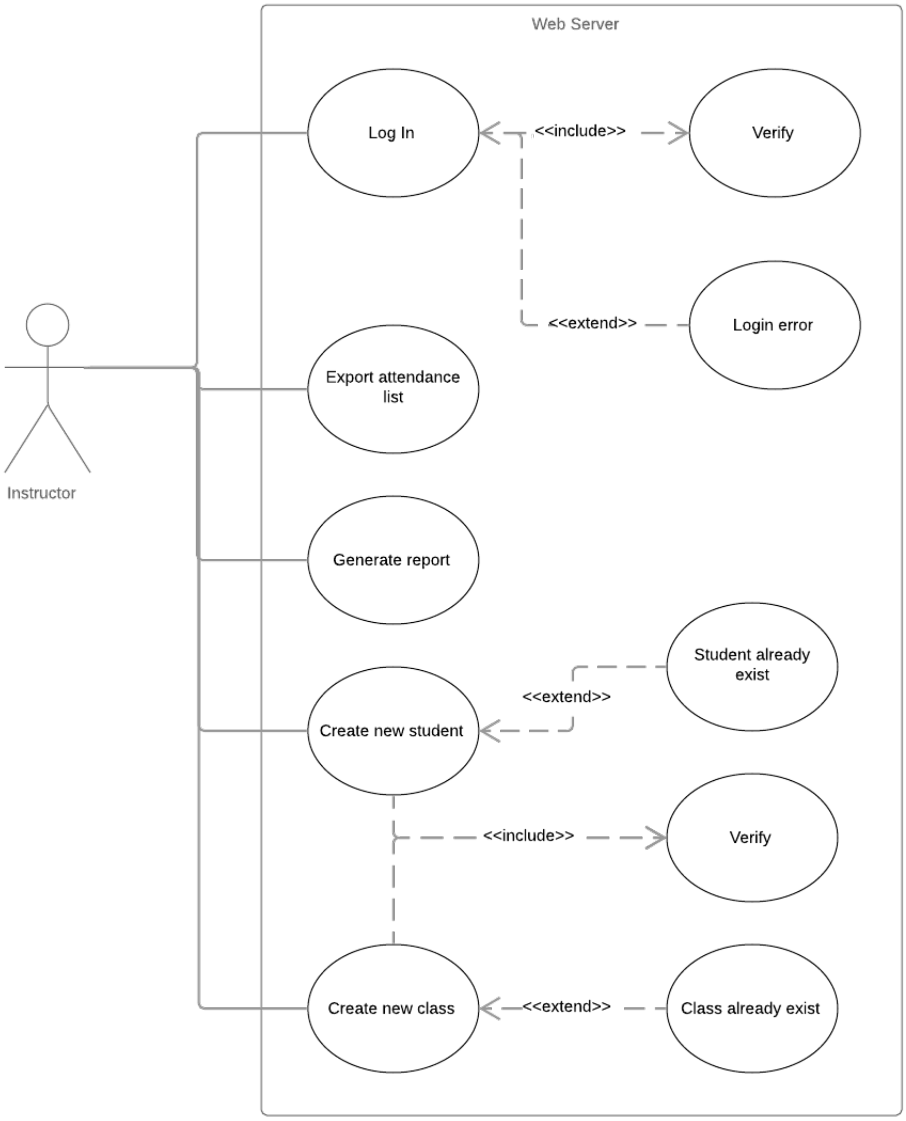
Following use case shows how a student and device interacts.



Following table shows the scenario for the use case above.

Name	Put the card on the device
Actors	Student
Description	Student use the card to log in to the class
Succesful Completion	1. Device sends the ID to server 2. Server looks for a match 3. Server returns success 4. Indication LEDs turn on
Alternative	1. Device sends the ID to server 2. Server looks for a match 3. Server returns failure 4. Indication LEDs turn on
Pre-condition	Student must have a valid RFID card
Post-Conditions	None
Assumptions	None

Following figure shows how instructor interacts with the web server.



Following tables defines scenarios for use cases above.

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

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

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

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

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

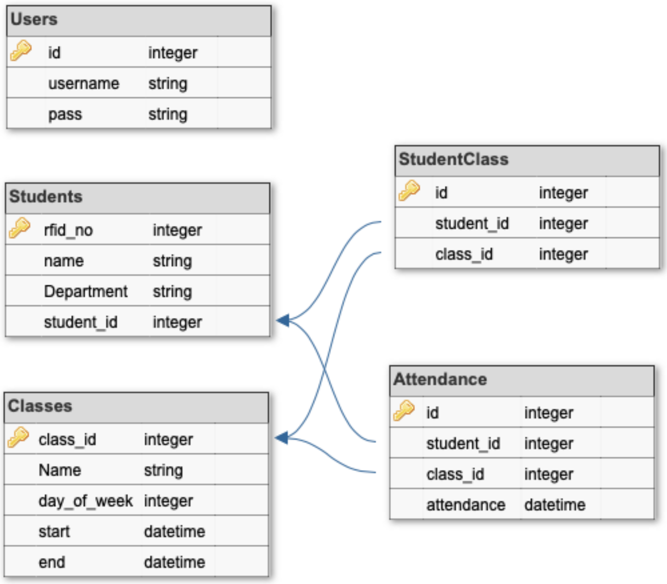
4.1.3 Functional and Non-Functional Requirements

No	Functional Requirement
1	Ability to distinguish students by using their student card
2	Ability to create readable meaningful reports
3	Affordable enough

No	Non-Functional Requirement
1	Should have long battery life
2	Simple user interface must be provided
3	Should have small form factor
4	Should consume as less as possible system resources
5	Data flow should be secure
6	Should be highly standalone
7	Should have long term support

4.2 Efficient Data Structure

Developers tried to come up with simplest DB for prototype. Following figure shows the structure.



Users	
id	integer
username	string
pass	string

Students	
rfid_no	integer
name	string
Department	string
student_id	integer

Classes	
class_id	integer
Name	string
day_of_week	integer
start	datetime
end	datetime

StudentClass	
id	integer
student_id	integer
class_id	integer

Attendance	
id	integer
student_id	integer
class_id	integer
attendance	datetime

5 Other Nonfunctional Requirements

5.1 Safety Requirements

Device should be kept away from hot and cold places since it has a LiPo battery inside and it can be dangerous when explodes.

5.2 Software Quality Attributes

Web server can be adapted to any device that can send ID of an RFID card. Therefore developers can make IoT devices to enhance the system.

With our 'Communication Protocol Guide' developers can contribute product's software and hardware easily.

6 Other Requirements

For project to work with university's automation system a special integration sprint is required since every system has its own different internals.

This prototype has no intention to integrate itself as such. But for the first commercial release, it is possible.

6.1 Appendix A: To Be Determined List

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