Attendance Tracker

Group 8
Michael Fay, William Quade, Joel Wach

Software Requirements Specification

Document

Version: 1 **Date: (02/10/2017)**

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to describe in detail the functionality of the Attendance Tracker system. The intended audience of this SRS document includes the system's stakeholders, and the people involved in the rest of the process of building the system.

1.2 Scope

This document describes the three software products being produced, the web application, the automatic attendance tracking system, and the Canvas integration plugin.

The web application will keep class attendance records. This is the core of the system, other products require the web application to function.

The automatic attendance tracking system will be a hardware device that associates Bluetooth addresses with students, keeping track of who is in the classroom and reporting back to the web application.

The Canvas integration plugin interfaces with the Canvas API, and the web application, to sync attendance, grades, students, and classes between the two systems.

The Attendance Tracker system will be used in classrooms where attendance needs to be taken. The goal is to have a process for taking attendance that is more efficient than alternative processes like a sheet of paper, or a clicker. A system that takes less time to setup and use will result in more available class time for learning.

1.3 Definitions, Acronyms, and Abbreviations.

- BTLE - Bluetooth Low Energy.
- Canvas - A learning management system used by schools.
- FERPA - Family Educational Rights and Privacy Act.
- My.UNL - UNL account login system.
- MTBF - Mean Time Between Failures, a measure of how reliable a system is.vb
- (R1) - A feature that will be implemented in the first release.
- (R2) - A feature that will be implemented in a subsequent release.
- Raspberry Pi - A small single board computer capable of running Linux, and communication over Bluetooth and WiFi.
- SRS - Software Requirements Specification, this document.
- TCP - Transmission Control Protocol, part of the Internet protocol network communication suite.

- UDP - User Datagram Protocol, part of the Internet protocol network communication suite.
- WiFi - Technology for wireless local area networking.

1.4 References

- 1.4.1 - Canvas REST API documentation: https://canvas.instructure.com/doc/api/index.html
- 1.4.2 - Bluetooth Low Energy http://www.elinux.org/RPi_Bluetooth_LE

1.5 Overview

The rest of this document covers the overall description, and the specific requirements for the Attendance Tracker system. The overall description section describes the general use and purpose of the system, while the specific requirements section goes into detail on the system's features.

2. The Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

The primary interface that our application will utilize is the Canvas API.

2.1.2 Interfaces

The web application will have an interface for students, teachers, and administrators. Students will have the ability to do tasks like take attendance, and manage Bluetooth devices. Teachers will have the ability to manage their classes, and their students. Administrators will be able to manage administrators, teachers, classes, and students.

2.1.3 Hardware Interfaces

Hardware requirements are only applicable to our automatic attendance tracking system, or base station. The base station must contain a medium to connect to the internet and, to connect to Bluetooth enabled devices. The protocol used by the base station to determine a student's attendance is Bluetooth and Bluetooth Low Energy. The primary protocol used by the base station to submit attendance back to the web application is TCP and UDP.

The web application, and Canvas integration plugin, will not have any hardware requirements.

2.1.4 Software Interfaces

The Canvas integration plugin must automatically sync records to and from Canvas using the Canvas API. The Canvas integration plugin interfaces directly with the web application. The automatic attendance tracking system must interface with the web application to take attendance.

2.1.5 Communications Interfaces

Our system will communicate primarily using UDP and TCP. Each automatic attendance taking system will implement Bluetooth LE to communicate with student devices.

2.1.6 Memory Constraints

The web application runs on a server, so there is no memory constraint, it can be upgraded to scale with the number of users. The only part of the system with a hard memory constraint is the automatic attendance tracking system, which runs on a Raspberry Pi Model B, which has a maximum 500 MB of memory available for the application.

2.1.7 Operations

The automatic attendance taking system will only have one mode of operation. It will continuously wait for a request from web application.

2.1.8 Site Adaptation Requirements

A Bluetooth receiver, in this case a Raspberry Pi, must be installed in every room before the software can be used. The room must have a separate web enabled device for the teacher to monitor attendance through the web application..

2.2 Product Functions

The system's only function is to make taking class attendance an easier and more efficient process for students and teachers.

2.3 User Characteristics

Students are one class of user accessing the system. They all have completed high school, and are enrolled in college classes. Levels of experience, and technical expertise vary. Teachers have completed college, and have varied levels of experience and technical expertise. Administrators have varied levels of education, experience, and technical expertise.

Based on those user characteristics the system will be designed to be straightforward for a high school graduate, who has used other web applications, to use.

2.4 Constraints

One of the system's biggest constraints is the rate at which the automatic attendance tracker can scan for nearby Bluetooth devices. The system needs to be sample at a high enough rate to take attendance. The automatic attendance tracker must also connect to the web application to stay updated. The web application must also connect to third party systems, like Canvas. This requirements heavily constrains our design. The application's requirement to connect and update academic records requires our application to hardened to FERPA security standards. It also requires our application to be reliable in accordance to section three requirements.

2.6 Apportioning of Requirements.

Individual features marked as (P1) are included in the initial release of the system, while features marked as (P2) will be added to the system in future releases.

3. Specific Requirements

3.1 External Interfaces

The system shall interface with the Canvas API to synchronize attendance records, classes, students, and teachers. See 1.4.1 for Canvas API documentation.

3.2 Functions

3.2.1 Web application

3.2.1.1 Student user class

3.2.1.1.1 Account management

3.2.1.1.1.1 Introduction

Students shall be able to manage the information associated with their account.

3.2.1.1.1.2 Stimulus/Response sequence

Students shall be able to add and edit information associated with their account.

3.2.1.1.1.3 Associated functional requirements

3.2.1.1.1.3.1 Registration (R1)

Students shall be able to register for an account with the system by providing information including name, email address, and student ID.

3.2.1.1.1.3.2 Edit user information (R1)

Students shall be able to edit the information associated with their account (see 3.2.1.1.1.3.1).

3.2.1.1.1.3.3 Class registration (R1)

Students shall be able to request access to classes (see 3.2.1.2.1.3.5).

3.2.1.1.1.3.4 Bluetooth device association (R1)

Students shall be able to add and remove Bluetooth devices that are associated with their account (see 3.2.2).

3.2.1.1.1.3.5 My.UNL login (R2)

Students shall be able to associate their account with a My.UNL account and use their My.UNL account for authentication.

3.2.1.1.2 Attendance

3.2.1.1.2.1 Introduction

Students shall be able to mark their attendance for their classes.

3.2.1.1.2.2 Stimulus/Response sequence

Students shall be able to automatically mark their attendance, or have their attendance automatically recorded.

3.2.1.1.2.3 Associated functional requirements

3.2.1.1.2.3.1 Manual attendance entry (R1)

Students shall be able to manually mark their attendance (see 3.2.1.2.2.3.1).

3.2.1.1.2.3.2 Automatic attendance entry (R1)

Students shall be able to automatically mark their attendance (see 3.2.2).

3.2.1.2 Teacher user class

3.2.1.2.1 Class management

3.2.1.2.1.1 Introduction

Teachers shall be able to manage their classes.

3.2.1.2.1.2 Stimulus/Response sequence

Teachers shall be able to add, edit, and remove information relating to their classes.

3.2.1.2.1.3 Associated functional requirements

3.2.1.2.1.3.1 Add class (R1)

Teachers shall be able to add a class to the system by providing information including class name, class year, and class schedule.

3.2.1.2.1.3.2 Edit class (R1)

Teachers shall be able to edit the information associated with a class that they added (See 3.2.1.2.1.3.1).

3.2.1.2.1.3.3 Delete class (R1)

Teachers shall be able to remove a class that they added from the system.

3.2.1.2.1.3.4 Add students (R1)

Teachers shall be able to associate students with classes that they created.

3.2.1.2.1.3.5 Student approval (R1)

Teachers shall be able to approve or disapprove students who request access to classes that they created (see 3.2.1.1.1.3.3).

3.2.1.2.1.3.6 Remove students (R1)

Teachers shall be able to remove students from classes that they added.

3.2.1.2.2 Attendance taking

3.2.1.2.2.1 Introduction

Teachers shall be able to take attendance.

3.2.1.2.2.2 Stimulus/Response sequence

Teachers shall be able to manually take attendance themselves, and have students mark their attendance.

3.2.1.2.2.3 Associated functional requirements

3.2.1.2.2.3.1 Manual attendance taking student (R1)

Teachers shall be able to manually take attendance by having students input a system generated, time sensitive code into the system (see 3.2.1.1.2.3.1).

3.2.1.2.2.3.2 Manual attendance taking teacher (R1)

Teachers shall be able to manually take attendance by marking students as present.

3.2.1.2.3 Attendance export

3.2.1.2.3.1 Introduction

Teachers shall be able to access attendance records.

3.2.1.2.3.2 Stimulus/Response sequence

Teachers shall be able to export attendance records to a file, or into external systems.

3.2.1.2.3.3 Associated functional requirements

3.2.1.2.3.3.1 File export (R1)

A teacher shall be able to export attendance records in a variety of formats to a file that is downloadable.

3.2.1.2.3.3.2 Canvas export (R2)

A teacher shall be able to export attendance records into Canvas (see 3.2.3).

3.2.1.2.4 Attendance authenticity report

3.2.1.2.4.1 Introduction

Teachers shall be able to recognize attendance fraud.

3.2.1.2.4.2 Stimulus/Response sequence

Teachers shall be able to get information about the authenticity of their attendance records.

3.2.1.2.4.3 Associated functional requirements

3.2.1.2.4.3.1 Authenticity check (R2)

The system shall check attendance records for signs of fraud, for example, multiple sign ins from a single device.

3.2.1.2.4.3.2 Fraud notification (R2)

Teachers shall be notified when possible attendance fraud has been detected.

3.2.1.2.4.3.3 Authenticity report (R2)

Teachers shall be able to generate a downloadable report file with information about possible attendance fraud.

3.2.1.3 Administrator user class

3.2.1.3.1 Teacher management

3.2.1.3.1.1 Introduction

Administrators shall be able to manage teachers.

3.2.1.3.1.2 Stimulus/Response sequence

Administrators shall be able to add, edit, and remove teachers, and access their records.

3.2.1.3.1.3 Associated functional requirements

3.2.1.3.1.3.1 Add teacher (R1)

An administrator shall be able to add a new teacher to the system using information including name, and email address.

3.2.1.3.1.3.2 Edit teacher (R1)

An administrator shall be able to edit information associated with teachers.

3.2.1.3.1.3.3 Remove teacher (R1)

An administrator shall be able to remove teachers from the system.

3.2.1.3.1.3.4 Teacher functionality (R2)

An administrator shall be able to access all of the same functionality as a teacher (see 3.2.1.2).

3.2.1.3.2 Student management

3.2.1.3.2.1 Introduction

Administrators shall be able to manage students.

3.2.1.3.2.2 Stimulus/Response sequence

Administrators shall be able to add, edit, and remove students, and access their records.

3.2.1.3.2.3 Associated functional requirements

3.2.1.3.2.3.1 Add student (R1)

An administrator shall be able to add a new student to the system using information including name, student ID, and email address.

3.2.1.3.2.3.2 Edit student (R1)

An administrator shall be able to edit information associated with students.

3.2.1.3.2.3.3 Remove student (R1)

An administrator shall be able to remove students from the system.

3.2.1.3.2.3.4 Student functionality (R2)

An administrator shall be able to access all of the same functionality as a student (see 3.2.1.1).

3.2.1.3.3 Administrator management

3.2.1.3.3.1 Introduction

Administrators shall be able to manage administrators.

3.2.1.3.3.2 Stimulus/Response sequence

Administrators shall be able to add, edit, and remove administrators.

3.2.1.3.3.3 Associated functional requirements

3.2.1.3.3.3.1 Add administrator (R1)

An administrator shall be able to add a new administrator into the system using information including name, and email address.

3.2.1.3.3.3.2 Edit administrator (R1)

An administrator shall be able to edit information associated with administrators.

3.2.1.3.3.3.3 Remove administrator (R1)

An administrator shall be able to remove administrators from the system.

3.2.2 Automatic attendance tracking system

3.2.2.1 Web application Bluetooth address association

3.2.2.1.1 Introduction

The system shall support a method to link a user with a Bluetooth device address.

3.2.2.1.2 Stimulus/Response sequence

The system shall relate a user with their appropriate account when Bluetooth address is found.

3.2.2.1.3 Associated functional requirements

3.2.2.1.3.1 The system will support a method to link new users with a device Bluetooth address.

3.2.2.2 Web application hardware to room association

3.2.2.2.1 Introduction

The system shall support a method to link the hardware per room to the room’s attendance.

3.2.2.2.2 Stimulus/Response sequence

The system shall link the inroom hardware to its appropriate class when new hardware is presented.

3.2.2.2.3 Associated functional requirements

3.2.2.2.3.1 The system shall support a method to add new rooms and hardware.

3.2.2.3 Bluetooth system hardware

3.2.2.3.1 Introduction

The system shall support independent hardware on a per room basis.

3.2.2.3.2 Stimulus/Response sequence

The system hardware shall automatically or manually detect registered and non-registered devices within range.

3.2.2.3.3 Associated functional requirements

3.2.2.3.3.1 Automatic Bluetooth search

The system shall be able to automatically refresh it’s list of present users.

3.2.2.3.3.2 Web connection

The system shall require an active internet connection and thus need a WiFi card or ethernet run.

3.2.2.4 Continual Bluetooth address logging

3.2.2.4.1 Introduction

The system shall require a method to log new and existing Bluetooth addresses.

3.2.2.4.2 Stimulus/Response sequence

The system shall log a new Bluetooth address when one appears and try to place it with a registered user.

3.2.2.4.3 Associated functional requirements

3.2.2.4.3.1 Logging Location

The system shall require a location to store all of the registered users and their Bluetooth addresses.

3.2.2.4.3.2 Logging Organization

The system shall support sorting by various values.

3.2.2.5 Web application syncrhonization

3.2.2.5.1 Introduction

The system shall support updating the Web application from each room's hardware.

3.2.2.5.2 Stimulus/Response sequence

The system shall receive new data from Bluetooth clients and be able to display it in a user friendly manner.

3.2.2.5.3 Associated functional requirements

3.2.2.5.3.1 Timely updates

The system shall update automatically or manually to display logged users.

3.2.2.5.3.2 Demand updates

The system shall support the web application requesting an update of logged users.

3.2.3 Canvas integration plugin

3.2.3.1 User syncronization

3.2.3.1.1 Introduction

The system shall support synchronizing users from Canvas into the system.

3.2.3.1.2 Stimulus/Response sequence

The system shall automatically or manually sync users from Canvas into the attendance system.

3.2.3.1.3 Associated functional requirements

3.2.3.1.3.1 Automatic periodic student sync (R2)

The system shall be able to automatically sync student users from Canvas into the attendance system.

3.2.3.1.3.2 Manual student sync (R2)

The system shall be able to sync student users from Canvas into the attendance system by being manually triggered by an administrator.

3.2.3.1.3.3 Automatic periodic teacher sync (R2)

The system shall be able to automatically sync teacher users from Canvas into the attendance system.

3.2.3.1.3.4 Manual teacher sync (R2)

The system shall be able to sync teacher users from Canvas into the attendance system by being manually triggered by an administrator.

3.2.3.2 Class syncrhonization

3.2.3.2.1 Introduction

The system shall support synchronizing classes from Canvas into the system.

3.2.3.2.2 Stimulus/Response sequence

The system shall automatically or manually sync classes from Canvas into the attendance system.

3.2.3.2.3 Associated functional requirements

3.2.3.2.3.1 Automatic periodic class sync (R2)

The system shall be able to automatically sync classes from Canvas into the attendance system.

3.2.3.2.3.2 Manual class sync (R2)

The system shall be able to sync classes from Canvas into the attendance system by being manually triggered by an administrator.

3.2.3.3 Attendance grade syncrhonization

3.2.3.3.1 Introduction

The system shall support synchronizing attendance records to and from Canvas.

3.2.3.3.2 Stimulus/Response sequence

The system shall automatically or manually sync attendance records from Canvas into the attendance system, and from the attendance system to Canvas.

3.2.3.3.3 Associated functional requirements

3.2.3.3.3.1 Automatic periodic attendance sync out (R2)

The system shall be able to sync attendance records from the attendance system into Canvas automatically.

3.2.3.3.3.2 Manual attendance sync out (R2)

The system shall be able to sync attendance records from the attendance system into Canvas by being manually triggered by an administrator or a teacher.

3.2.3.3.3.3 Automatic periodic attendance sync in (R2)

The system shall be able to sync attendance records from Canvas into the attendance system automatically.

3.2.3.3.3.4 Manual attendance sync out (R2)

The system shall be able to sync attendance records from Canvas into the attendance system by being manually triggered by an administrator or a teacher.

3.3 Performance Requirements

The web application will run on a server, which can be upgraded to handle more concurrent users as the system grows a user base. Through hardware upgrades, the system should be able to handle the 25,897 students at UNL, along with teachers using the system. 95% of the interactions with the web application shall be processed in less than five seconds.

3.4 Logical Database Requirements

3.4.1 User information

The system shall store user information including name, email address, student ID, and role. This information is used primarily for authentication, and is used frequently. It is kept until deleted.

3.4.2 Class information

The system shall store class information including name, teacher, schedule, room, and associated students. This information is used primarily to keep attendance records, and is used frequently. It is kept until deleted.

3.4.3 Attendance records

The system shall store attendance records including how attendance was recorded, the class it was taken for, and a date. This information is recorded frequently, but isn't retrieved as often, it is used for tasks like the attendance report exports. It is kept until deleted.

3.4.4 Room to logging device association

The system shall store records associating the Bluetooth logging devices with specific rooms. This information is accessed frequently for tasks like turning logging device logs into attendance records. It is kept until deleted.

3.4.5 Logging device logs

The logging device shall store timestamps and Bluetooth addresses for devices that it sees. This information is continually logged, and is retrieved frequently, it is used to take automatic attendance. It is kept for one week.

3.5 Design Constraints

The automatic attendance taking functionality requires devices that have Bluetooth built in. The web application requires only an internet connection, and a modern web browser.

3.6 Software System Attributes

3.6.1 Reliability

The system should have a MTBF of 168 hours.

3.6.2 Availability

The system should be available to run at all times. Upon failure, the individual components of the system can restart and continue to function.

3.6.3 Security

The system will use an SSL certificate to encrypt user information while it is in transit on the internet. Once stored, personal information will be encrypted using SQL Server encrypted columns.

Multiple user classes including an administrator group, a teacher group, and a student group, will limit what individual users can do while using the system.

3.6.4 Maintainability

Documentation, and version control will be used to promote maintainability. The system will be built as individual components, which will also make it easier to add additional functionality to the system in future releases.

3.6.5 Portability

The web application will run on an Amazon Web Services server instance, and the central database will be a SQL based server. This will allow the application to dynamically scale to fit current demand. It also allows for the application to be cross platform.

The automatic attendance tracking system will run on a Raspberry Pi, and be designed to run on any Linux system with support for Bluetooth and WiFi.

The Canvas integration plugin will be built as a separate component, so that in the future the system can change as the Canvas API changes. Other student management systems can be added to the system.