Easy Attendance

Software Requirements Specification

for

S

Version 1.0 approved

Prepared by Liam Strausser, Austin Feuerman, Mitchell Kelly

LAM Software LLC.

October 31, 2016

Table of Contents

Table of Contents..........................................................................................................................ii

Revision History.............................................................................................................................ii

1.   Introduction..............................................................................................................................1

1.1   Purpose.................................................................................................................................1

1.2   Document Conventions........................................................................................................ 1

1.3   Intended Audience and Reading Suggestions......................................................................1

1.4   Product Scope.......................................................................................................................1

1.5   References............................................................................................................................1

2.   Overall Description..................................................................................................................2

2.1   Product Perspective..............................................................................................................2

2.2   Product Functions................................................................................................................. 2

2.3   User Classes and Characteristics.........................................................................................2

2.4   Operating Environment......................................................................................................... 2

2.5   Design and Implementation Constraints................................................................................3

2.6   User Documentation............................................................................................................. 3

2.7   Assumptions and Dependencies...........................................................................................3

3.   External Interface Requirements........................................................................................... 3

3.1   User Interfaces..................................................................................................................... 3

3.2   Hardware Interfaces............................................................................................................. 4

3.3   Software Interfaces............................................................................................................... 4

3.4   Communications Interfaces.................................................................................................. 5

4.   System Features.....................................................................................................................5

4.1   System Feature 1..................................................................................................................5

4.2   System Feature 2……………................................................................................................5

4.2   System Feature 3……………................................................................................................6

5.   Other Nonfunctional Requirements....................................................................................... 6

5.1   Performance Requirements...................................................................................................6

5.2   Safety Requirements.............................................................................................................7

5.3   Security Requirements..........................................................................................................7

5.4   Software Quality Attributes.................................................................................................. 7

5.5   Business Rules......................................................................................................................8

6.   Other Requirements................................................................................................................8

Appendix A: Glossary................................................................................................................... 8

Appendix B: Analysis Models....................................................................................................... 8

Appendix C: To Be Determined List........................................................................................... 8

Use Cases………………………………………………………………………………………………...9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Liam, Mitchell, Austin | 11/2/2016 | First Draft | 1.0 |
|  |  |  |  |

**1.** **Introduction**

**1.1** **Purpose**

The purpose of this document is to describe Easy Attendance. It will cover how the system works, the interfaces of the system, how the system reacts to user input, and how the system is supposed to be used.

**1.2** **Document Conventions**

There are no document conventions used in this write-up.

**1.3** **Intended Audience and Reading Suggestions**

This document is intended for developers, marketing staff, users, testers, and documentation writers. This document contains an overview of the project, how the interface will work, the features of the project, and requirements of the system. Developers should look at the how the interface will work and the requirements, so the system is created to the intended design. Marketing staff should be familiar with the overview of the system and its features, so they know what to promote. Testers should be familiar with the overview of the system and know all the requirements to make sure the system works correctly. Documentation writers should know the overview, interface and features in order to create a substantial manual.

**1.4** **Product Scope**

Easy Attendance is an automatic attendance system that will use Bluetooth to check proximity. It will also store all attendance records and give the option to see the records in a grid format, like Excel. The system will make it easier for institutions that require participants to attend events, such as university classes. The system will be able to determine who is present based on whether the system can detect a Bluetooth signal from the participant’s device or not. The goal would be to create a system that simplifies the attendance checking process, which will improve the efficiency of the event.

In a university setting, a moderator could run Easy Attendance, and it will automatically count the students who enter as present once they turn their Bluetooth on. This will make taking attendance easier for teachers, and create an incentive for students to go to class because the system will not overlook absences. In a business setting, this system could be used to confirm employees log times easily without taking up a lot of time.

**1.5** **References**

Easy Attendance will be using Bluecove JSR-82 and Netbeans to utilize the Bluetooth functionality.

**2.** **Overall Description**

**2.1** **Product Perspective**

This system is a new, self-contained product. The major component of the system is a device for the moderator to run the program on, such as a computer. Participants will also need a device that have Bluetooth capability, but there will be an option to mark participants as not having a device and an additional web-based check-in.

**2.2** **Product Functions**

* Let the user create events
* Let the user edit events they have created
* Let the user add participants to an event
* Let the user edit participants they have added to an event
* Let the user add a participant’s Bluetooth device
* The system will check for the device
* The system will mark participants as “Present” if their device is found before the event starts
* The system will mark participants as “Late” if their device is found after the event starts and before it ends
* The system will mark participants as “Absent” if their device is not found before the event ends
* Let the user check for devices
* Let the user manually change attendance status
* Manual changes take priority
* Let the moderator get a report of attendance statuses
* Let participants of the event use a web-based check in to mark their status

Use Cases – See Page 9

**2.3** **User Classes and Characteristics**

The users involved are moderators, who check the attendance for the events, and participants, who attend the events and need their attendance status recorded.

The moderator is expected to understand how attendance works. Moderators must have basic knowledge of computer applications in order to effectively use Easy Attendance. The moderator should have the knowledge of how Bluetooth devices are paired.

The participant must be experienced with the Internet in order to utilize the web-based check-in if they need to. The participant must know how to turn Bluetooth on if they have a Bluetooth device.

**2.4** **Operating Environment**

The system will operate on computers using Windows 7 (or above). The web-based check in option will be available on any web browser.

**2.5** **Design and Implementation Constraints (Optional)**

The Bluetooth connection should only check if the devices are in range of the moderator’s computer. No participant data should be able to be received from a participant device.

**2.6** **User Documentation**

A user manual will be available in the application for moderators. It will give the users the information necessary to create their own events with participants, pair devices with the system, manually check attendance, and receive a report on attendance records.

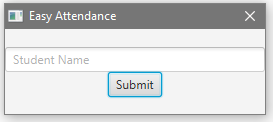
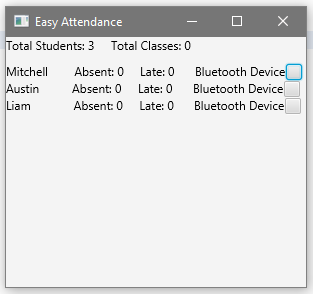
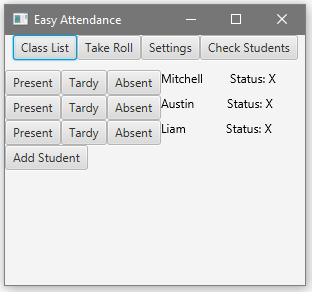
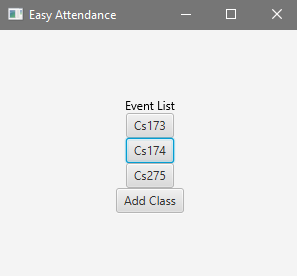
**2.7** **Assumptions and Dependencies**

Assume the moderator computer has Bluetooth capability. Assume participant devices will be found by the system. Assume the user understands how to use the system correctly and effectively.

**3.** **External Interface Requirements**

**3.1** **User Interfaces**

The GUI will consist of three main scenes and a web interface for students who do not have Bluetooth enabled devices. The first scene is for selecting and adding classes. Each class will be a button on this scene and when pressed will take the instructor to a list of students in the specified class with their current attendance status. In the student screen there will be buttons to return to the previous class list screen, a button to begin taking roll button, a button to edit the settings of the class, and a button to move to the check student’s scene. The check student’s scene will show students in the class and their attendance records as well as if they have a Bluetooth device. This scene will allow for the instructor to check or uncheck if the student has a Bluetooth device. The web interface will have a student name input and a verification setting to prove they are in the class. The GUI will have a pop up window for adding classes and students. Rough GUI scenes have been attached.



**3.2** **Hardware Interfaces**

The hardware used will be the system the moderator runs Easy Attendance on and the Bluetooth devices the students use to check in. The moderator’s system will have to be running Windows 7 or greater and the student's Bluetooth devices will need to support at least Bluetooth 2.0.

**3.3** **Software Interfaces**

Easy Attendance will utilize Bluetooth (2.0 or higher) through the Bluecove JSR-82 implementation. Easy Attendance will search for Bluetooth signals from mobile devices to determine if a student's device is in range. If Bluetooth signals are found by the instructors system, Easy Attendance will mark the student as present or tardy. Easy Attendance will also have to receive attendance info through web form submission to mark a student as present or tardy.

**3.4** **Communications Interfaces**

Easy Attendance will use a web interface to check attendants in. This will be done electronic form web submission. A student's attendance records will need to be secured while the data is transferred to an instructor's system.

**4.** **System Features**

**4.1** **Graphical User Interface**

4.1.1   Description and Priority

Easy Attendance will provide a graphical user interface for navigating functionality and viewing the program. This is a high priority feature due to it being necessary for using the program.  
Benefit 9, Penalty 1, Cost 6, Risk 3

4.1.2   Stimulus/Response Sequences

The GUI will start when the program does. The GUI will allow the instructor to view, add and remove classes; view add and remove students as well as view their attendance records.

4.1.3   Functional Requirements

REQ-1: The system shall have a graphical user interface.

REQ-2: The system shall provide functionality to view classes.

REQ-3: The system shall provide functionality to add classes.

REQ-4: The system shall provide functionality to remove classes.

REQ-5: The system shall provide functionality to view student lists

REQ-6: The system shall provide functionality to add students

REQ-7: The system shall provide functionality to remove students

REQ-8: The system shall provide functionality to view student’s attendance

**4.2** **Bluetooth Functionality**

4.2.1   Description and Priority

The device will be able to search for Bluetooth signals from devices to determine if a student is present. This feature is high priority due to it being a major feature of Easy Attendance.  
Benefit 9, Penalty 1, Cost 7, Risk 3

4.2.2   Stimulus/Response Sequences

A Bluetooth search will start when the instructor presses the “Take Roll” button on the GUI. When a student enters the classroom and the device finds their Bluetooth signal it will mark them as present or tardy.

4.2.3   Functional Requirements

REQ-9:   The system shall provide functionality for Bluetooth connection

REQ-10: The system shall record student’s attendance according to time

**4.3** **Alternative Check-in**

4.3.1   Description and Priority

Easy Attendance will provide an alternative check in for students who do not have a Bluetooth enabled device  
Benefit 7, Penalty 4, Cost 8, Risk 5

4.3.2   Stimulus/Response Sequences

If a student does not have a Bluetooth enabled device, they will be able to use a web interface to check in to class. This interface will ask a user to verify that they are actually present in class.

4.3.3   Functional Requirements

REQ-1: The system shall have a graphical user interface.

REQ-10: The system shall record student’s attendance according to time.

REQ-11: The system shall provide functionality for an alternative check-in.

**5.** **Other Nonfunctional Requirements**

**5.1**     **Performance Requirements**

REQ: The system shall look for phones from a class list using Bluetooth.

This nonfunctional requirement must be put in to show how the system is going to check for student attendance, they will use a class list which includes a Bluetooth id for each student and the system will check to see if that student's id is present on the Bluetooth list.

REQ: The system shall check through the class list once every minute from 5 min before to 5 min after the start of class (present check period).

This nonfunctional requirements must be put in to create a present period for the student. The system will refresh every minute and will see if the student's Bluetooth id is present during that time.

REQ: The system shall check that the student is present for at least three class list checks.

This check will make sure that the student is present for at least three refreshes, this is so the student will be unable to stop outside of a classroom and get marked as present and leave.

REQ: The system shall check through the class list once every five minutes after the present check period.

This nonfunctional requirement is to check to see if the student comes late to class. The system will refresh every five minutes and mark the student as late during this period. Anyone not checked in after this late period will be marked as absent.

REQ: The system shall allow for alternative check in methods, which include moderator non-automated functionality.

This nonfunctional requirement will allow for the moderator to go through and change attendance log if needed as well as allow for the moderator to check in the student if they do not have a Bluetooth capable device.

REQ: The system shall generate a csv file with updated class attendance after each class.

This nonfunctional requirement will make it easier for the moderator to see after each class where the student stands attendance wise. The system will generate a csv file and that class and all previous classes to show where the student stands in an easy to read file for the moderator.

REQ: The system shall give moderator a warning list when student starts to miss too many classes.

This nonfunctional requirement will allow the moderator to give a number of absences when the moderator wants to be warned if that number gets too high. The system will generate a list and show the list as students get on it.

**5.2** **Safety Requirements**

Will add later as become more readily available.

**5.3** **Security Requirements**

REQ: The system shall ensure that user data is not accessed beyond the student's Bluetooth id.

This will ensure that a user does not have any information stolen from using this system.

REQ: The system shall have safety in place to ensure a student cannot access the moderators account to change lateness’s and absences.

This is to make sure that a student does not stand outside of a classroom, turn on their Bluetooth and get counted as present.

**5.4** **Software Quality Attributes**

REQ: Our system shall strive for maintainability, testability, usability, portability, reusability and correctness.

These are not the only software quality attributes, these are just the main ones we plan to strive for.

**5.5** **Business Rules**

The system shall allow the moderator to use the moderator system, which includes checking to see student’s attendance and changing attendance as they see necessary.

The system shall only allow students to use a Bluetooth capable device and use Bluetooth id to check in for class.

**6.** **Other Requirements**

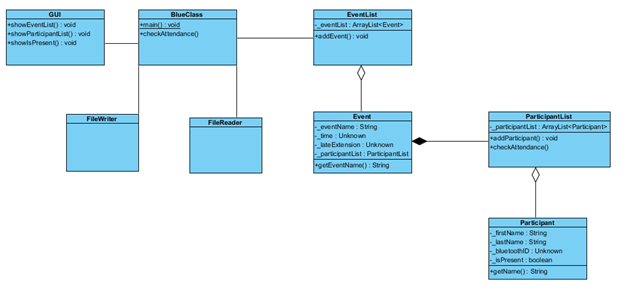
For future growth of Easy Attendance we would like to take the outline of the code and create an Easy Work Login. This will allow for the company to expand using a lot of legacy code.

**Appendix A: Glossary**

Use Case Diagram:

Attn - Attendance

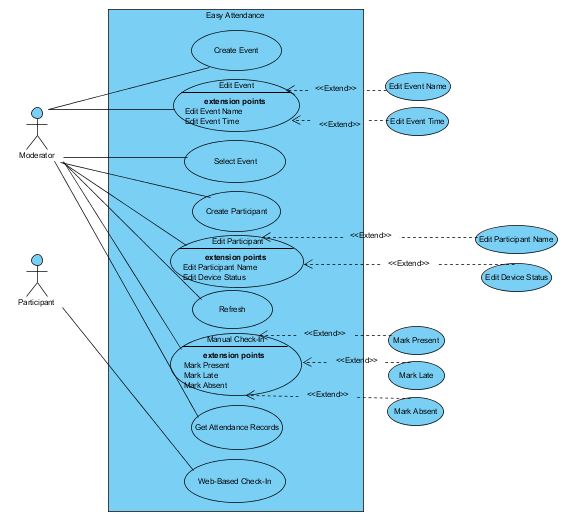
**Appendix B: Analysis Models**



**Appendix C: To Be Determined List**

This is a future update section, we would like to scale this into a work related concept. Using it to sign in and out hourly workers.

Easy Attendance Use Cases



#### **2.2.1 Moderator Use Cases**

#### **Use case:** Create Event

**Diagram:**

Moderator

Create Event

**Brief Description**

The moderator creates an Event that prompts for an Event Name, Start Time, End Time, and Late Factor.

**Initial Step-By-Step Description**

Before this use case can be initiated, the moderator has started the Easy Attendance Program.

1. The moderator presses the Create Event button.
2. The interface shows the fields that need to be filled in.
3. The moderator fills in the required fields.
4. The moderator presses the Apply button.

The system creates the Event and adds it to the Event List.

#### **Use case:** Edit Event

**Diagram:**

Moderator

Edit Event

**Brief Description**

The moderator can change the information of an Event.

**Initial Step-By-Step Description**

There must be an existing Event to edit.

1. The moderator presses the Edit Event button

2. The interface shows the fields that hold the current Event information.

1. The moderator changes the information in these fields.
2. The moderator presses the Apply button.

The system changes the Event information to the new input.

#### **Use case:** Edit Event Name

#### **This use case extends the Edit Event use case**

**Diagram:**

Moderator

Edit Event Name

**Brief Description**

The moderator edits the Event Name.

**Initial Step-By-Step Description**

The user must be in the Edit Event menu in order to edit the event name.

1. The moderator clicks on the text box labeled “Event Name”.
2. The moderator enters the desired name.

The new name will be set if the user clicks Apply in the Edit Event window.

#### **Use case:** Edit Event Time

#### **This use case extends the Edit Event use case**

**Diagram:**

Moderator

Edit Event Time

**Brief Description**

The moderator edits the Event Time, which includes start and end time and late factor.

**Initial Step-By-Step Description**

The user must be in the Edit Event menu in order to edit the Event name.

1. The moderator chooses the time from the drop down menu that the Event starts.
2. The moderator chooses the time from the drop down menu that the Event ends.
3. The moderator enters how many minutes the Late Factor will be.

The new time information will be set if the user clicks Apply in the Edit Event window.

#### **Use case:** Select Event

**Diagram:**

Moderator

Select Event

**Brief Description**

The moderator chooses the Event to open.

**Initial Step-By-Step Description**

There must be an existing Event to open.

1. The moderator presses on the Event they would like to view.

2. The interface shows the participants in the Event.

3. The interface shows the time information of the Event.

4. The interface shows the manual attendance options next to participants.

5. The interface shows the Refresh button

The system shows user a window with the Event window and all the options listed above.

#### **Use case:** Create Participant

**Diagram:**

Moderator

Create Participant

**Brief Description**

The moderator creates a Participant to add to the Event.

**Initial Step-By-Step Description**

There must be an Event for the participant to be added to.

1. The moderator presses the Create Participant button.
2. The interface shows the fields that need to be filled in.
3. The moderator fills in the required fields.
4. The moderator pairs the participants Bluetooth device with the system.
5. The moderator presses the Apply button.

The system creates a participant with the entered information and adds them to the participant list in the Event.

#### **Use case:** Edit Participant

**Diagram:**

Moderator

Edit Participant

**Brief Description**

The moderator can edit the information of a participant.

**Initial Step-By-Step Description**

There must be a participant to edit.

1. The moderator presses the Edit Participant button.
2. The interface shows the fields that need to be filled in.
3. The moderator fills in the required fields.
4. The moderator presses the Apply button.

The system will update the new participant information

#### **Use case:** Edit Participant Name

#### **This use case extends the Edit Participant use case.**

**Diagram:**

Moderator

Edit Participant Name

**Brief Description**

The moderator can edit the name of the participant

**Initial Step-By-Step Description**

There must be a participant to edit.

1. The moderator click in the text box labeled Participant Name.
2. The moderator enters the desired name.

The system will update the new participant name if the moderator clicks Apply.

#### **Use case:** Edit Device Status

#### **This use case extends the Edit Participant use case.**

**Diagram:**

Moderator

Edit Device Status

**Brief Description**

The moderator can edit device that is paired or set the participant as not having a device.

**Initial Step-By-Step Description**

There must be a participant to edit.

1. The moderator clicks the Pair Device button.
2. The moderator pairs a new device.

The system will update the new participant device, or set the participant as not having one if there is no device paired, if the moderator clicks Apply.

**Use case:**  Refresh

**Diagram:**

Moderator

Refresh

**Brief Description**

The moderator can start the device search process.

**Initial Step-By-Step Description**

There must be an Event to refresh and participants to be looked for.

1. The moderator presses the refresh button.
2. The system goes through the device checking process.

The Event has been refreshed and the attendance status will change for each participant if they have been discovered.

#### **Use case:** Manual Check-In

**Diagram:**

Moderator

Manual Check-In

**Brief Description**

The moderator can manually check in participants.

**Initial Step-By-Step Description**

There must be participants of an Event to be marked.

1. The moderator chooses the status they would like to set to a participant.

The participant’s status is set to what they moderator chose.

#### **Use case:** Mark [Status] (Present, Late, Absent)

#### **This use case extends the Manual Check-In use case**

**Diagram:**

Moderator

Mark [Status}

**Brief Description**

The moderator can choose what status they would like to set a participant to.

**Initial Step-By-Step Description**

There must be an Event and a participant in that Event to set the status to

1. The moderator presses Mark Present, Late, or Absent button.

The participant’s status is set to whatever the moderator has selected.

#### **Use case:** Get Attendance Records

**Diagram:**

Moderator

Get Attn. Records

**Brief Description**

The moderator can get the records of all Event days and attendance status of participants for those days.

**Initial Step-By-Step Description**

There must be an Event to get the records from, and at least one Event day to have attendance statuses of.

1. The moderator presses the Get Attendance Records button.
2. The system creates a file for the spreadsheet.
3. The system asks the user where they would like to save the file.

The system creates a spreadsheet of the information

2.2.2 Participant Use Cases

#### **Use case:** Web-Based Check-In

**Diagram:**

Participant

Web Check-In

**Brief Description**

The participant can sign in from a web-based application.

**Initial Step-By-Step Description**

The participant has to be added to the event they are trying to mark their attendance for.

1. The participant opens the web-application in a browser.
2. The participant enters their credentials.
3. The student clicks Present

The system marks the student as Present