**Software Requirements Specification**

**for**

**Attendance Auditor +**

**Version 1.0.0 approved**

**Prepared by:  
 Robert Airth  
Skye Gibney  
Adrian Haro  
Cristian Puac**

**GAHP Dev Group**

**Last Edited:12/10/2020**

**Table of Contents**

Table of Contents

[**1. Introduction** 1](#_Toc58594808)

[**1.1 Purpose** 1](#_Toc58594809)

[**1.2 Intended Audience and Reading Suggestions** 1](#_Toc58594810)

[**1.3 Product Scope** 1](#_Toc58594811)

[**2. Overall Description** 1](#_Toc58594812)

[**2.1 Product Perspective** 1](#_Toc58594813)

[**2.2 Product Functions** 1](#_Toc58594814)

[**2.3 User Classes and Characteristics** 2](#_Toc58594815)

[**2.4 Operating Environment** 2](#_Toc58594816)

[**2.5 Design and Implementation Constraints** 2](#_Toc58594817)

[**3. External Interface Requirements** 2](#_Toc58594818)

[**3.1 User Interfaces** 2](#_Toc58594819)

[**3.2 Hardware Interfaces** 2](#_Toc58594820)

[**4.** **System Features** 3](#_Toc58594821)

[**4.1 Admin/Organizer Event Management** 3](#_Toc58594822)

[**4.3 Member Events/Items functionality** 4](#_Toc58594823)

[**5.** **Other Nonfunctional Requirements** 4](#_Toc58594824)

[**5.1 Performance Requirements** 4](#_Toc58594825)

[1. Fast response time under few seconds 4](#_Toc58594826)

[2. Handles errors gracefully 4](#_Toc58594827)

[**5.2 Safety Requirements** 4](#_Toc58594828)

[**5.3 Software Quality Attributes** 5](#_Toc58594829)

[1. Robust implementation/modular and integrable 5](#_Toc58594830)

[2. Cheap to implement 5](#_Toc58594831)

[3. High acceptability/availability (web-browser implementation) IE, Firefox, Chrome 5](#_Toc58594832)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **1. Introduction**

## **1.1 Purpose**

The product described in this SRS is the Attendance Auditor Plus (v1.0.0). This SRS pertains to the entirety of the system.

## **1.2 Intended Audience and Reading Suggestions**

This document is intended for developers, testers, and project managers. It contains information on user and system requirements as it pertains to creating the system itself and subsequent testing for system correctness. The document will be used during the development process, and so there is no single way to proceed through this document. Instead, it will be used as a reference and a goal marker to guide development.

## **1.3 Product Scope**

The software described here is intended for event organizers looking for a simple, easy-to-implement solution for event management that allows them to keep track of event attendees. The software should provide a way of generating reports for specific attendees, listing the workshops they have attended during a given time period.

# **2. Overall Description**

## **2.1 Product Perspective**

The motivational scenario for creating this software was as a solution for attendance auditing for professional development workshops at a community college. Event organizers needed a way to measure interest in a workshop, register who actually attended, and later would need to create reports for the attendance of a specific faculty member to give to the dean of their department. The software is intended to be a standalone system that can be easily hosted on the client’s servers.

## **2.2 Product Functions**

1. User RSVPs for an event to show interest
2. Event organizer registers attendees
3. Event organizer compiles a report of a specific attendee and all the workshops they have visited during a given time period
4. Event organizer compiles a report of attendees for a set of workshops and/or series of workshops
5. Attendee views calendar of upcoming events that have been input by the event organizer
6. Attendee views report of past events attended

## **2.3 User Classes and Characteristics**

1. Admin
   1. Most privileges, able to view logs of actions made by individual organizers
2. Event organizers
   1. Have higher privileges to be able to manage events and audit attendees
3. Event attendees/members
   1. Can view past and future general event information and view their own past-attendance data.

## **2.4 Operating Environment**

Backend will be hosted on the client’s servers using NodeJS and Express, and the database is hosted using MongoDB Atlas. Front-end will be implemented in EJS. Client’s server environment will vary.

## **2.5 Design and Implementation Constraints**

Client organization will maintain the delivered software once implemented.

# **3. External Interface Requirements**

## **3.1 User Interfaces**

1. Logical Characteristics(projected) -- Hosts and Attendees
   1. Hosts will receive special registration links
   2. Hosts will use web browser to access the web application
   3. Hosts will be presented with a GUI
   4. GUI will take user input from BIOS
   5. GUI will be able to display audit info or a form for Attendees
   6. GUI will have big but decent sizes buttons
   7. GUI error messages will be displayed from invalid links or access

## **3.2 Hardware Interfaces**

1. Hosts can email links to possible Attendees
2. Will be multi-platform to work with browsers on computers or smartphones

**3.3 Communications Interfaces**

1. HTTP and or HTTPS between all GET’s and POSTs

# **System Features**

## **4.1 Admin/Organizer Event Management**

4.1.1 Description and Priority

1. Admin will be able to view logs of event organizer activities (high priority)
2. Admin will be able to view past events (medium priority)
3. Admin can audit the event attendance data (medium priority)

4.1.2 Stimulus/Response Sequences

When an Admin decides to host an event, the system will ask the user to enter the name of the event, the time the event will happen, and also the place that this event will take. Once this information is entered, the system will ask the Admin if everything they have entered is accurate to the information the event is supposed to have.

Once the event is published onto the system, the Admin will be able to view the event, as well as any past events. For the current event, Admin are able to audit the attendance data to see who is coming, as well as getting information of who exactly attended on the day of the event.

4.1.3 Functional Requirements

4.1.3.1. Event Create and Delete

4.1.3.2. View Event Data

4.1.3.3. Event Registration Check In/Out

4.1.3.4 Manage Profile’s and Registrations

**4.2 Inventory Functional Requirements**

4.2.0 Functional Requirements

4.2.1 Inventory list; view the available inventory

2.2.2 Search inventory to see if avail or not

2.2.3 Inventory management (org only)

2.2.3.1 Add item

2.2.3.2 Edit item

2.2.3.3 View serial number

2.2.3.4 Checkout/return items

2.2.4 Checkout details

2.2.4.1 For members, this will take the value “In Stock” or “Checked Out”

## **4.3 Member Events/Items functionality**

4.3.1. Description and Priority

1. Through registration, members can sign into system (medium priority)
2. Members can view past events, whether they attended or declined
3. Members can RSVP to an event and attend
4. Members can browse and look at Items

4.3.2. Stimulus/Response Sequences

When members decide to sign up for an event, they are presented with different upcoming events they could sign up for. Members will be able to sign-in by providing an email school address and their unique campus identification number. Once signed up, members will be reserved a space in the event. Members can view any past events that they have attended as well, or events that they did not attend even though they had said they were.

# **Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

## Fast response time under few seconds

## Handles errors gracefully

1. Can’t be offline longer than 5 seconds

## **5.2 Safety Requirements**

1. *Sensitive information can be kept safe by limiting some access to information for users to take a look at.* 
   1. *Attendee has the most limited view of the database, they can only see their own past-attendance information, future-attendance information, and general event information*
   2. *Event Organizer has access to event management functions, future-attendance information, attendance check-in functions, and general event information.*
   3. *Lead Event Organizer has access to everything the Event Organizer has access to plus the ability to view past-event attendance information and attendee attendance information.*

## **5.3 Software Quality Attributes**

## Robust implementation/modular and integrable

## Cheap to implement

## High acceptability/availability (web-browser implementation) IE, Firefox, Chrome

**Appendix A: Software Process Model**

*The development of the software will be implemented using an* ***Agile*** *methodology.*

**Appendix B: Roles for Sprint 1**

1. *Team Leader (Cristian & Robert)*
   * *coordinating team effort during sprints*
   * *clarifying project sprint objectives*
   * *maintaining morale*
   * *scheduling team meetings*
2. *Lead Programmer (Skye)*
   * *Coordinating coding efforts*
   * *Refining design approaches*
   * *Coordinating software development*
   * *Oversight of software-interface implementations*
   * *Oversight of hardware-interface implementations*
3. *Lead Designer(Cristian)*
   * *Designing program structure*
   * *Designing user interfaces*
4. *Quality Assurance Leader(Testing)(Robert)*
   * *Frequent review of all software-engineering processes*
   * *Ensuring software meets SRS requirements before, during, and after each sprint.*
   * *Creating testing methods to verify usability and functionality of each software functional area.*
   * *Creating testing methods to verify usability and functionality of the completed software.*
5. *Assistant QA (Skye)*
6. *Analyst/Requirements Engineer(Adrian)*
   * *Oversees the development of the SRS and milestones*
   * *providing reports on your analysis to the lead engineer or other stakeholders in the project.*
7. *They/them (Skye)*