**Software Design Specification**

**For**

**Attendance Auditor +**

**Powered by:**

**Adrian Haro**

**Robert Airth**

**Skye Gibney**

**Cristian Puac**

**Last Revised: October 17, 2020**

**Table of Contents**

1. **Introduction …………………………………………………………………………. 3**

**1.1 Overview and Purpose ……………………………………………………….. 3**

**1.2 Functional Requirements ……………………………………………………. 3**

1. **Interface Design (3 functional areas) …………………………………………... 4**

**2.1 Homepage …………………………………………………………………….... 4**

**2.2 Inventory Page ………………………………………………………………… 5**

**2.3 Event details page ……………………………………………………………. 6**

**2.4 Profile ………………………………………………………………………….. 7**

**2.5 Member sign-in page ………………………………………………………… 8**

**2.6 Login page ……………………………………………………………………… 9**

1. **Data Design (3 functional areas) ……………………………………………….. 10**

**3.1 Database schema for oracle SQL**

1. **Testing Plan Table (for at least 1 functional area) ……………………………. 11**

**4.1 Overview**

**4.2 Create Event TEST**

**1. Introduction**

**1.1 Overview and Purpose**

This document is intended to describe the design for Attendance Auditor Plus (AA+). AA+ is a web application that allows event organizers to create events and keep track of the attendees planning to attend these events. This software is also designed to provide the different events that attendees can choose to attend, while also listing past events that the attendees have attended.

**1.2 Functional Requirements**

- Organizer Event Management ( Ref: 4.1.3.1.)

- Member Event Registration (Ref: 4.1.3.2.)

- Inventory Management (Ref: 4.1.3.3.)

**2. Interface Design**

**2.1 Homepage**

2.1.1 Calendar

2.1.2 Upcoming Events

2.1.3 Create event(org only)

**2.2 Inventory Page**

2.2.1 Inventory list

2.2.2 Search inventory

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”

**2.3 Event details page**

2.3.1 Event details

2.3.2 RSVP button

2.3.3. Event management (org only)

2.3.3.1 Edit

2.3.3.2 Delete

2.3.3.3 Sign-in members

**2.4 Profile page**

2.4.1 Member/organizer info

2.4.2 View past events attended

2.4.3 View future RSVP’d events

2.4.4 Edit profile button

2.4.5 List items currently checked out

**2.5 Event sign-in page**

2.5.1. List of RSVPs with checkboxes to mark them as present

2.5.2. Button to sign-in member

2.5.2.1. Search members

2.5.3. Button to sign-in guest

2.5.4. Separate checkbox list for non-RSVPs

**2.6 Login page**

2.6.1 Username

2.6.2 Password

2.6.3 Submit

**3. Database description**

**3.1 Entity-Relationship Diagram Model**

Below is a diagram which describes the relationship between the entities in the database.

**4. Appendix A: Testing Plan Tables**

**4.1 Overview**

This section describes the test cases that should be carried out by the Attendance Auditor Plus application. Each test is represented by a different functional requirement that is to be carried out through various actions. Each case also details the inputs needed to carry out the test case as well as the expected outputs.

**4.2 Inventory Management (Reference 4.1.3.4)**

|  |  |
| --- | --- |
| **Test ID** | **Inventory1.1** |
| Title | Member -- View Checked Out Items |
| Objective | A member can search what items are checked out under their name. |
| Setup | Member is logged into system |
| Test Data | Assume instance exists |
| Test Actions | Click-view-inventory, cancel |
| Expected  Result | “Apple Laptop #12, description, checked out by: Cristian Puac,  Checked out date: 11/11/2021 , Intended return date: 11/18/2021” |
| Actual  Result |  |

|  |  |
| --- | --- |
| **Test ID** | **Inventory1.2** |
| Title | Member -- Search Inventory |
| Objective | A member can view items to check out. |
| Setup | A logged in member can view a list of items to check out. |
| Test Data | “” |
| Test Actions | Search, cancel |
| Expected  Result | “Laptops: pc#1, pc#2, pc#3; Calculators: calc#1, calc#2, calc#3;  Camcorders: cmc#1, cmc#2, cmc#3;” |
| Actual  Result |  |

|  |  |
| --- | --- |
| **Test ID** | **Inventory1.3** |
| Title | Editing an Item |
| Objective | Change the description of Item |
| Setup | Member is logged into system can search the inventory, search a specific item to edit |
| Test Data | Type in the item to be searched and find item, click on item description and to edit it out |
| Test Actions | Type item name that should be searched (e.g. Laptop) -> click search button -> click laptop to be edited -> click edit description -> change the edits -> save new description |
| Expected  Result | “Success item description changed  from ‘Apple Ipad\_2ndGen’  To ‘Apple Ipad\_4thGen’ |
| Actual  Result |  |

|  |  |
| --- | --- |
| **Test ID** | **Inventory1.4** |
| Title | Check-In/Return an item |
| Objective | Member can return checked out item |
| Setup | Need to have instance under unavailable |
| Test Data | Expects Item, Member, and the date of return |
| Test Actions | Submit, Cancel |
| Expected  Result | “Success! ‘Andrew Martinez’ returned ‘WindowsXP’ “ |
| Actual  Result |  |