Technical Design Document

AIMS - Apprentice Internship Management System



# Version History

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Description** |
| 2016/05/25 | Cole Chambers | Initial Creation |
| 2016/06/15 | Jeff Pickett | Initial Creation |
| 2016/06/16 | James Keim | Added server details |
| 2016/06/23 | Jeff Pickett | Incorporate initial feedback |
| 2016/07/20 | Jeff Pickett | Incorporate additional feedback |
| 2016/08/18 | Tiffany Trusty | Small change to registration and updated data model |
| 2016/12/05 | Tiffany Trusty | Updated Data model image to reflect actual web admin portal model |

# Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| SOW | Statement of Work |
| TDD | Technical Design Document |
| AIMS | Apprentice Internship Management System |
|  |  |
|  |  |
|  |  |

# References

1. AIMS SOW, EFC Salesforce / Tinderbox
2. AIMS Web Admin Wireframes

# Table of Contents

[1 Version History 2](#_Toc456779486)

[2 Acronyms 2](#_Toc456779487)

[3 References 2](#_Toc456779488)

[4 Table of Contents 3](#_Toc456779489)

[5 Introduction 4](#_Toc456779490)

[6 Scope 5](#_Toc456779491)

[6.1 Mobile App Features 5](#_Toc456779492)

[6.2 Web App Features 5](#_Toc456779493)

[7 Mobile Wireframes 6](#_Toc456779494)

[7.1 Login 6](#_Toc456779495)

[7.2 Register 7](#_Toc456779496)

[7.3 Calendar 8](#_Toc456779497)

[7.4 Daily Check In 9](#_Toc456779498)

[7.5 Weekly Survey 10](#_Toc456779499)

[7.6 Raise Hand / Ask Question 11](#_Toc456779500)

[7.7 Profile 12](#_Toc456779501)

[8 Web App Wireframes 13](#_Toc456779502)

[9 Data Model 14](#_Toc456779503)

[9.1 Data Model Details 14](#_Toc456779504)

[10 Technical Architecture 15](#_Toc456779505)

# Introduction

The purpose of this document is to expand upon the SOW and define the technical architecture and high level design required to fulfill project scope. Anything not included in this document is considered out of scope and may incur additional time and costs.

The Apprentice Internship Management System (AIMS) is an application that works to resolve pain points that plague the academy administrators, instructors, and students. The system will provide for tracking of apprentice profile information, attendance and hours, course scheduling with times and locations, and provide a mechanism for a student to ask questions anonymously within a classroom setting.

Upon completion, the participating apprentices and all future apprentices will have a functional architecture that demonstrates best practices in communicating with a back-end API powered by a custom relational database with multiple user interfaces for the web, iOS, and Android. The completed project will be an adaptable legacy product that benefits all parties involved. Future apprentices can expand upon this foundation to address future needs of the academy while greatly enhancing their portfolio with a full stack product.

# Scope

This document is structured as follows:

* Mobile Wireframes
* Web Wireframes
* Database Diagram
* API Specifications
* Technical Architecture

## Mobile App Features

The native mobile apps will include the following features:

* Login / Forgot Password / Logout
* Registration - Basic email address / password only
* Course Schedule Calendar Interface with times & locations
* Daily Punch Card - Utilize geofences to check people in / out
* Weekly Score Card – Allow employers or apprentices to be sent a template survey to gather feedback
* Raise Hand / Ask Question - Allows users (apprentices, students, etc.) to ask questions, anonymously if desired
* User Profile – View and update user information
* Notifications - Receive event notifications from the server

## Web App Features

The administrative web app will include the following features:

* Creation, assignment, general management of users, groups, organizations and roles
* Creation, assignment, general management of events
* Creation, assignment, general management of surveys
* Administrative reports
* System configuration

# Mobile Wireframes

Below are proposed wireframes for the mobile application

## Login

The login page is a standard login page that passes the username and credentials to the server via SSL returning an authentication token.

Also on the page is a Forgot Password button that will send the server a API call that will initiate a password change email.



## Register

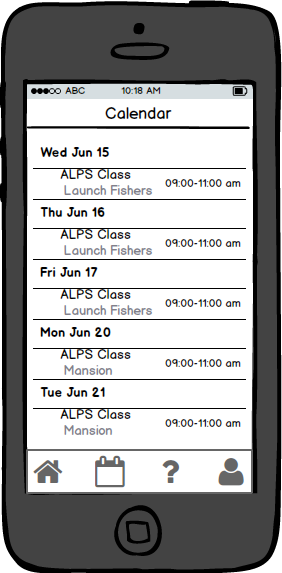
The register page is the page that allows a new user to request an account on the server. While this does create the account, it is expected that an administrator will still need to assign roles to the user after the user has registered and created their account.

Note that when the user successfully submits this page, he or she will get a message that says, “We’ve received your request to join this AIMS group, you will be receiving a follow up email with instructions later.”



## Calendar

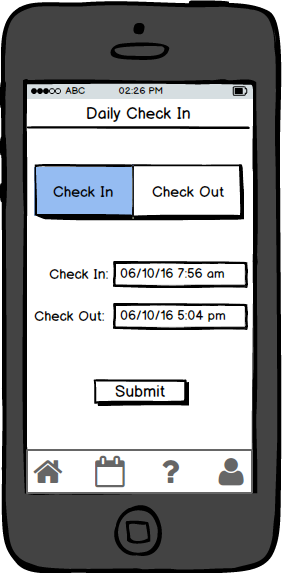
The calendar screen will allow the user to view the date, time and location of each of the events in their calendar stored on the server.

****

## Daily Check In

The daily check in will allow the user to submit when they arrive and leave for the day. This will automatically log the time and location of each event. Additionally, the entry and exit of the established geofences will be logged as well.

Note that the first time this screen is used, the user will be asked for permission to access their location.



## Weekly Survey

This page will allow a user to take a quick survey on a regular basis. The questions are stored on the server which will allow for the questions to be dynamic. The answers will be in two parts. The first part is a score (e.g. 1-5 stars) and the second will be an optional note if the user desires to provide context around the scores.

The icons in the survey screen will be fetched from the server. For the initial release, a single set of smiley icons will be used.



## Raise Hand / Ask Question

This screen allows a user (student, apprentice, other) to ask a question which will be seen by the admin for a given group. On the Question page, a user will only be able to see their own questions. An admin will be able to see the questions for their group or groups.

In the question list, an individual's view (e.g. Student) will only have the student's questions, but other views (e.g. Teacher) can have views of a group.



## Profile

This screen is where a user can view and edit their profile information. This screen also allows for the user to change their avatar and password. Additionally, there are read-only icons at the bottom that are shown only if indicated by the server.

At the bottom of the profile screen, there is a section for badges that show specific items that have been completed (e.g. code badges, tests that have been taken, background checks, etc.). For each of these items, clicking on one can take the user to an external web view to a URL specified by the server. The icon, title, description, and URL are all returned from the server.



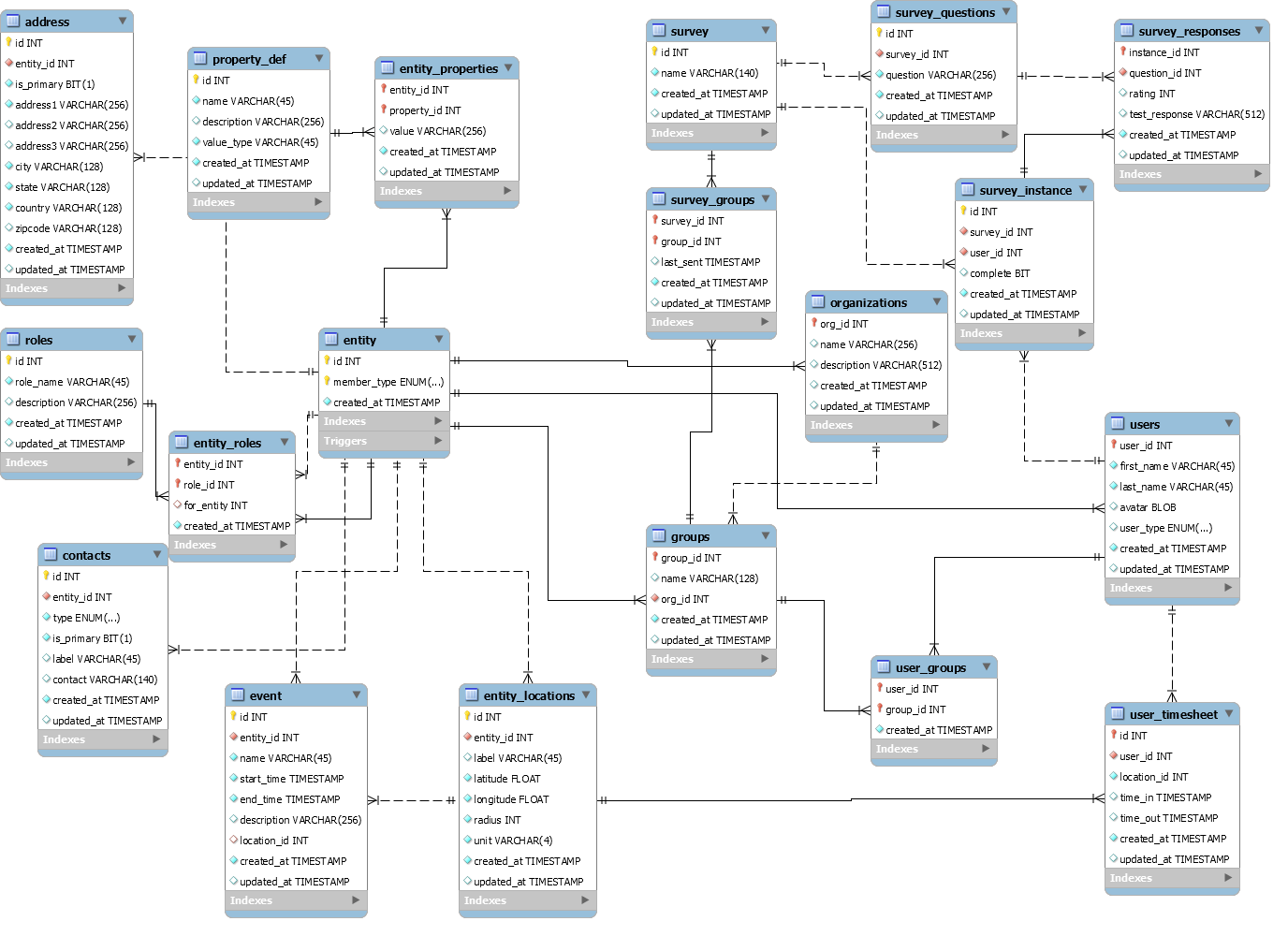
# Web App Wireframes

See companion document with web admin wireframes

# Data Model

## Data Model Details

Users, group, and organizations will all be considered entities. A trigger on the entity table will adjust the id ranges of the various types to ensure they are recognizable. This will avoid duplication of data and lead to a more normalized solution.



# Technical Architecture

Below is a generic technical architecture that could vary based on the implementation details. For example, if it is decided that the high load would require adding multiple servers, there would need to be a load balancer in front of the server components.

Also it is assumed that the server(s) will have a firewall in front of it, and all network communications will be TLS/SSL encrypted.

