Technical Design Document

**Apprentice Internship Management System**

 

# Version History

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Description** |
| 2016/5/25 | Cole Chambers | Initial Creation |
|  |  |  |
|  |  |  |

# Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| TDD | Technical Design Document |
| SOW | Statement of Work |
|  |  |
|  |  |
|  |  |
|  |  |

# References

1. <document title>, <document location>

# Table of Contents

[1 Version History 2](#h.gjdgxs)

[2 Acronyms 2](#h.30j0zll)

[3 References 2](#h.1fob9te)

[4 Table of Contents 3](#h.2et92p0)

[5 Introduction 4](#h.tyjcwt)

[6 Scope](#h.3dy6vkm) 5

6.1 Features 5

6.2 Exclusions [5](#h.3dy6vkm)

[7 Technical Architecture 6](#h.1t3h5sf)

[7.1 Use Cases 6](#h.4d34og8)

[8 Technical Design](#h.2s8eyo1)

[8.1 Design Constraints](#h.17dp8vu)

[8.2 Data Model](#h.3rdcrjn)

[8.3 User Interface](#h.26in1rg)

[9 Platform Specific Technical Information](#h.lnxbz9)

# 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 throughout the remaining documentation) is an application that works to resolve numerous 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 classroom messaging with chat and anonymous question submission.

Upon completion, the participating apprentices and all future apprentices will have a functional template architecture that demonstrates best practices in communicating with a back-end API powered by a custom relational database, with multiple user interfaces for 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

**6.1 Features**

The native mobile apps will include the following features:

* Registration / Account Creation - Basic username / password only
* Login / Forgot Password / Logout
* Course Schedule Calendar Interface with times & locations
* Raise Hand - Allows students to ask questions anonymously
* Location Punch Card - Utilize geofences to check people in / out
* Classroom List / Detail used to power messaging features
* Chat Room - Basic messaging features for the class / cohort
* User Profile - Typical user profile

**6.2 Exclusions**

The following items are specifically excluded from the scope of this project:

* Features not specifically defined in the SOW
* Android tablet mobile app
* iOS tablet mobile app
* Windows Phone mobile app
* Localization support
* Deployment to Production
* Development of the Database, API, Student Portal, and Admin Portal.
* *These will be developed and delivered by Eleven Fifty Academy apprentices*

# Technical Architecture

AIMS_Architectual_Diagram.png

**SQL Database**: This component is the database where backend data will be stored.

**Data API**: This component is the data API that references the database tables. See 8.2 Data API.

**API Access Layer**: This component exposes some aspects of the Data API to external users.

See 8.2 API Access Layer.

**Services**: These components form the assembly that encapsulates the business logic of the apps.

**Models**: These components are the provide data representations within the apps.

**User/Admin Portal**: These components represent the main UI of the apps.

See 8.3 User Interface.

## Use Cases

<This section is optional and should only be used for high level, architecturally significant use cases>

### Login Page

### AIMS CRUD Pages

# Technical Design

## Design Constraints

Browser, Android and iOS support will be needed across separate versions of the application. Development will not include specialized support for Windows phone, Android tablet or iOS tablet. As a result from utilizing multiple language support, API separation is required in order to avoid pitfalls during development (by allowing different methodology in different applications) while simultaneously increasing application security.

Confidential personal data including characteristics such as employment contracts, remuneration details, qualifications and absence information should all be considered private and these data points should not be accessible by third parties. Personal identifiers such as names, email addresses and phone numbers should be considered public domain for the intents of this project.

User authentication will be processed through Oauth2 across all platforms.

## Data API

The Data API will be the only service with direct database access. The service will use a Code First strategy with Data Migrations enabled to facilitate ease of deployment and incremental database development. The service will return jSON data responses to all subsequent services.

### <Identity Framework Entities>

These are auto-generated by the Identity Framework (AspNetUser, AspNetRole, etc.). ----TBD

### User

|  |  |  |
| --- | --- | --- |
| Id (key) | Int | Unique Id |
| UserId (foreign key) | Int | Relates to a User |
| Name | String | Name of user |
| Email | String | Contact email |
| Address | DateTime | Utc created date |
| Role | String | Apprentice/Instructor/Senior/Administrator |
| Class (foreign key) | ClassId | Relates to a Class |
| StartDate | DateTime | Apprenticeship/Hire start date |
| EndDate | DateTime | Apprenticeship/Employment end date |
| IsAdmin | Bool | Is/Is not an admin |
| IsDeleted | Bool | Is/is not deleted |

### Class

|  |  |  |
| --- | --- | --- |
| Id (key) | Int | Unique Id |
| ClassId (foreign key) | Int | Relates to a class |
| Title | String | Title of the class |
| Instructor | UserId | Instructor for class |
| Schedule | Array | List of events |
| StartDate | DateTime | Class start date |
| EndDate | DateTime | Class end date |

### 8.2.4 Event

|  |  |  |
| --- | --- | --- |
| Id (key) | Int | Unique Id |
| EventId (foreign key) | Int | Relates to an event |
| Title | String | Title of the event |
| Coordinator | UserId | The event poster |
| Description | String | Description of the event |
| StartDate | DateTime | Event start date |
| EndDate | DateTime | Event end date |

## User Interface

Mockups of the main pages of the app

### Add page

### 

### 

### 

# Platform Specific Technical Information

-Subject to change

### 9.1 Web App (Web/Web Admin)

This will be a C# .NET MVC Web application.

### 9.2 Android App

TBD

### 9.3 iOS App

TBD

### 9.4 API Access Layer (Mobile API services)

TBD. C# .NET MVC API.

### 9.5 Data API

C# .NET MVC API that provides extensive API access for the API Access Layers.