iAttend:

Software Requirements Specification (SRS) Document, Draft iii

Three Furious Locomotives

March 1, 2020

iAttend Software Requirements Specification (SRS) Document

SRS-ThreeFuriousLocomotives.tex
Draft iii
March 1, 2020
Three Furious Locomotives

Revisions

Version	Primary Author	Description of Version	Date Completed
i	TFL	Initial creation of document.	1/29/20
ii	Michael Jarman	Modifications as suggested by Professor Cindric	2/6/20
iii	Michael Jarman	Proofreading, converting to LaTeX	3/1/20

Contents

1 Introduction

1.1 Purpose

The purpose of this document is explain the workings of iAttend, and the functions it will provide The User in language understandable by both parties.

1.2 Description

This software product (iAttend) will serve as a more efficient means of recording attendance for the Music department at the University of Mount Union. The system will use QR codes identify students and scan and track their attendance events for recital credits. The iAttend system will consist of three main components: the database, the web service, and the mobile application. The three components will work seamlessly provide a quick and easy method for the department chair/professor create a course, its rules and roster, and administer attendance. It will allow students receive credit for their attendance and view their progress. Beneficial features of this product will include its security, as well as its simplicity and ease of use for those who may be technologically apprehensive.

1.3 Overview

This document will contain different sections describing various aspects of the software. These sections include a glossary of terms, descriptions of the product when related interfaces, user operations, functions of the product, lists of requirements, and discussion of a software development lifecycle model.

1.4 Glossary of Terms

- "The Software" shall refer the iAttend system in its entirety, including all devices involved.
- "The Team" shall refer the software development team in charge of creating The Software, known as Three Furious Locomotives.
- "The Administrator (Admin)" shall refer the owner of an iAttend Administrator account, which will have permission change configuration of their course as set up in the iAttend portal. For example, administrators would be department chairs or a professor configuring iAttend for their own classroom use.
- "The Moderator (Mod)" shall refer a user with elevated permissions. This user does not have permission change configuration of an established course but is able scan users and manually change attendance records. Professors assisting The Administrator with scanning would be examples of moderators.
- "The User (User)" shall refer a user with an ID that has no permissions beyond viewing their own attendance records. E.g. students, attendees.
- "The Portal" shall refer the web site that is accessible via a web browser. It will allow all types of users log in and view/edit records depending on their permission levels.
- "Course" shall refer the environment set up by The Administrator. It includes a roster of users record attendance for and has its own set of attendance records for each user for each event that takes place in that course. This could be a Math 101 course in Spring of 2020, for instance.
- "Check-In" shall refer the act of a user being scanned by a moderator or administrator, thereby recording attendance. This process involves the scanner scanning the QR code, identifying The User, and recording that attendance affirmation the database for the course.
- "Event" shall refer the specific event within the course for which a set of check-ins are grouped by. This could be attendance for the third lecture of a course, for instance.

- "The App" shall refer the mobile application for iOS primarily responsible for scanning QR codes and sending the check-in the database.
- Personally Identifiable Information (PII) refers information about a user that makes her/him identifiable in the real world, such as name, address, birthday, etc.
- Secure Sockets Layer (SSL) refers a way that data is securely sent across the Internet using encryption.
- Public Key Encryption refers a way of encrypting data where one key is publicly available while another key is privately held by the software owner. Together, these keys allow for encryption and decryption of data.
- Simple Mail Transfer Protocol (SMTP) is a way of transmitting emails.

1.5 Business Context

The Music department at the University of Mount Union has freely elected have The Team develop iAttend for their use in recording attendance at student showcases, recitals, and other events that mandate attendance.

2 Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

- Web Interface (The Portal) (Admin, Mod, and User logins).
- App Interface (The App) (Admin and Mod login, iOS).
- Export Attendance Reports Excel format

2.1.2 Memory Constraints

The application should require a minimal amount of space on a user's device and the web interface is not expected use much storage. A database server will store attendance records, and it may require a large amount of storage if the application is used by a large number of users. The application should not require a device with much memory, but the server may need a decent amount of memory keep up with possible high volume.

2.1.3 Operations

Normal and special operations required by The User:

- Registration for all types of users on the web portal
- Creation of course by admin on the portal
- Configuration by admin
- Scanning of user QR codes at an event by mod or admin
- Export of records by admin
- Manual record entry/correction by mod or admin

2.2 Product Functions

Functions performed by The Software:

- Take input of user scans
- Generate attendance reports for any and all events for any and all users

2.3 Similar Systems

There do exist some similar systems for tracking attendance, but many are complicated by complex login paradigms and unnecessary integration with other educational technology systems. The goal of iAttend is remain simple and not overextend its role, but perform the task of recording attendance reliably while avoiding becoming a technical nuisance.

2.4 User Characteristics

The intended users of The Software may be novice computer users but shall be able easily understand the functions of The Software and interact with it fulfill their needs.

2.5 User Objectives

iAttend will allow a moderator scan users in for various events that they expect users attend. The Moderator will scan codes that The Users provide, which will store their attendance records in a database. Users can view their own attendance, and The Moderators and admins can view all attendance records.

2.6 Constraints

Constraints on The Software include:

- Reliability requirements
- Must work every time a user is scanned
- Must generate correct reports on the portal
- Must reflect manual record changes globally
- Criticality of the application
- The Software must work as intended, as its output directly affects student grades and any credit for user attendance.
- Safety and security considerations/regulatory policies
- Must handle user information securely (Personally Identifiable Information, or PII, such as names, emails, and passwords)
- Must handle all transactions via an encrypted and secure connection (via Secure Socket Layer or SSL)
- Must be able run smoothly in a browser
- Interfaces other applications
- Should be able export an Excel compatible format
- Audit functions
- Reports generated will be visible admin and only the user's own records will be visible users
- Control functions
- Configure permissions and login information for users and admins
- Signal handshake protocols
- SSL, Public Key Encryption

3 Specific Requirements

Priority Scale: Low (1) – Medium (2) – High (3)

1. Low: Items that can be eliminated should the need arise, without adversely affecting the product. These items are not urgent and not as important the final product.
2. Medium: Items that are desired by the customer and/or users of the system, but that may be postponed until a future release. These items are not urgent and but are important parts of the final product.
3. High: Items that are mission critical and without which the system cannot function in a manner that is satisfactory the customer. These items are urgently needed and important the success of the final product.

3.1 Functional Requirements

- User Registration via Web Portal
  - Users shall be able register a new user account with the online web portal and associate with a course identifier
  - Priority rating: 3
  - Challenges: Email verification, credential storage, QR code generation
  - Risks: PII
  - Dependencies: Existing database for sign-ins, a stored procedure for registering a user, network connection, web portal
- User View
  - Users must be able view only their own records
  - Priority rating: 3
  - Challenges: Login, error retrieval
  - Risks: seeing another user's record
  - Dependencies: An existing database for sign-ins, stored procedure for accessing a single user's data from the database, network connection
- User Attendance Requirements
  - Users must be able be sorted by degree track for the purposes of differing attendance requirements
  - Priority rating: 3
  - Challenges: Administrator must validate this
  - Risks: user on the wrong set of requirements, admin doesn't update it
  - Dependencies: An existing database of degree tracks and requirements, an existing database of user attendance records, network connection
- Manually Changing Attendance
  - Moderator or Administrator must be able manually change attendance records
  - Priority rating: 3
  - Challenges: updating the record change in the database for all views
  - Risks: updating the record with invalid or incorrect information
  - Dependencies: An existing database of user attendance records, a web portal for admins or moderators edit records, a stored procedure for editing a user's attendance record, network connection

- Admin must be able set up and configure courses
  - Administrator must be able change requirements for user groups and edit configuration for a course
  - Priority rating: 3
  - Challenges: creating user groups that have different attendance requirements
  - Risks: corrupting existing data after editing
  - Dependencies: An existing database of degree tracks and requirements, network connection
- Multi-tier permissions system
  - Admin must be able select 2 or 3 tiers of permissions and add moderators if necessary. Otherwise, simply have a course with users and one admin.
  - Priority rating: 3
  - Challenges: allowing a course function with an Admin as the only scanner
  - Risks: giving the wrong user data, admin not sharing permissions
  - Dependencies: A variable in the user table denote permissions levels
- Notification system
  - Users shall be notified when their attendance isn't meeting the intended requirement for their attendance in the course via email
  - Priority rating: 2
  - Challenges: sending email, determining when the notification should happen
  - Risks: Not sending notifications in time leading users not knowing their situation
  - Dependencies: Network connection, SMTP connection
- Like button/feedback
  - Users shall be able give feedback on certain events via the web portal
  - Priority rating: 1
  - Challenges: creating an anonymous feedback system per event, storing feedback in database
  - Risks: losing data transmitting messages
  - Dependencies: A like count variable in the event table, an existing database of comments

3.2 User Interface Requirements

3.2.1 User Interface: Graphical (GUI) or Command-Line (CLI)

- Simple Graphical User Interfaces
  - The GUIs must be simple enough for the least technologically capable.
  - Priority rating: 3
  - Challenges: creating/finding the perfect user environment using feedback
  - Risks: User frustration and resistance adoption of the system
- Consistent Graphical User Interfaces
  - The GUIs must use a similar design language.
  - Priority rating: 2
  - Challenges: keeping the look and feel of the system consistent across platforms.
  - Risks: User frustration and resistance adoption of the system, multiple screens that are too similar may confuse some users

3.2.2 Application Programming Interface (API)

The User is not expected perform any programming. The Admin or other party wh- is in charge of setting up and maintaining the database server will need enter some commands initially create the database. These commands would include statements that set up the database tables and stored procedures, which would be provided in a file that could easily be copied and pasted, or possibly even an automated process.

3.2.3 Diagnostics (Error Reporting and Usage Logs)

Logs and error reporting will be collected in the server logs accessible by The Team. Their purpose is assist The Team in supporting the clients should something g- wrong.

3.3 System Requirements

- A desktop, laptop, or smart phone capable of running a modern web browser
- An iPhone that is capable of running up-to-date iOS

3.3.1 Hardware Interfaces

The phone application will interact with the phone itself store some data, such as encryption keys or a way remember the signed in user. This should be possible on any device, but since the application will first be designed for iOS, it must work on that hardware. The web interface will interact with the hardware device by downloading Microsoft Excel files. This should be possible on any device. Both the phone application and web interface will interact with a server storing the database. This should be possible with any server with a network connection that can run a database.

3.3.2 Communications Interfaces

The iAttend system will use Wi-Fi and ethernet for its communication. Much attention will be given security of the data all throughout the system. It will be detailed in section 3.5.3 Security.

3.3.3 Software Interfaces

A database and database management system (DBMS) would be required keep track of the attendance information of the users. The team has decided work with MySQL and the MySQL client program for these needs.

A website would need be deployed using a web host and have a connection the database. This link could be made using the ASP.NET framework.

The application would need interact with the camera feature of a moderator's device, which would require access certain features within the Android and iOS operating systems.

3.4 Domain Requirements/Constraints

The user's PII must be kept secure throughout the data transfer the server, as well as on the server itself.

3.5 Non-Functional Requirements

3.5.1 Reliability

- Must work every time a user is scanned.
- Must generate correct reports on the portal.
- Must reflect manual record changes globally.

3.5.2 Availability

Must be accessible at any time.

3.5.3 Security

- Must handle user information securely (Personally Identifiable Information, or PII, such as names, emails, and passwords).
- Must handle all transactions via an encrypted and secure connection (via Secure Socket Layer or SSL).

3.5.4 Maintainability

The Team shall be able log in the server via SSH and their own public key authentication in order push updates and bug fixes. The Team will be responsible for maintaining iAttend for the foreseeable future.

3.6 Logical Database Requirements

- Types of Data: The data that will be stored and used are names, emails, passwords, permissions, degree track, courses, events, check-ins, QR codes, and encryption keys.
- Frequency of use: This data will be used whenever the software is being used.
- Accessing capabilities: The data will be accessed by users, mods, and admins. It should be accessible at any time. The parts of the system that use the data are the app and the portal.
- Data entities and their relationships: The name, email, password, permissions, degree track, courses, events, check-ins, and QR code entities shall be associated with a user. The name, email, password, permissions, courses, events, and check-ins entities shall be associated with a mod. The name, email, password, permissions, courses, events, and check-ins entities shall be associated with an admin. All data types/entities shall be associated encryption keys, as all of the data shall be encrypted.
- Integrity constraints: Data shall not overwrite other data unless done so by the admin. Only admins, mods, or the team shall be able edit data.
- Data retention requirements: Data shall persist until deleted by the admin.

4 Software Life Cycle Model

4.1 Choice of Software Life Cycle Model

The Team shall use the Kanban SLCM. There may be slight variations the default model better keep track of who is working on what and how far along they are in the process. For example, each team member may have their own “In Progress” board, and there may be additional boards for testing or items that need additional work.

4.2 Justification for Choice of Model

Kanban was chosen due the nature of the project, where there are many different types of services that need be implemented. Therefore, The Team will be using best practices for development with microservices in order facilitate thorough testing and the best system integration. Kanban was seen as a good option where The Team can freely pull items from the backlog without worrying about sprints or other restraints put in place by other SLCMs.