Requirements Analysis Document

CITS3200 Group 27 – UWAttend Project

Preface:

This document addresses the requirements of the UWAttend web app project. The intended audience for this document is the designers and the clients of the project.

Group members:

Adam Wyatt, Delta Oliver, Isabella Rowe, James Dixon, Olivia Hanly, Zac Morris

1.0 General Goals

The primary goal of the UWAttend web app is to facilitate efficient and secure attendance tracking for university sessions. The app aims to provide facilitators with an intuitive interface to sign in, configure sessions, manage student information, and sync data with the central UWA database. The web app also ensures data security and accessibility across various devices and screen sizes.

2.0 Current System

The current system is a simple webapp with functionality for configuring session settings, signing students in/out, viewing basic class list information, and syncing the information to a remote database.

3.0 Proposed System

3.1 Overview

The proposed UWAttend web app will enhance attendance tracking for labs, tutorials, and other university session. The app will support multiple concurrent sessions, allowing students to be signed into different sessions by separate facilitators, with the unit coordinator deciding on the accuracy of sign ins. It will track student arrival and departure times to ensure they receive safety briefings and allow facilitators to manage student information, including session attendance, grades, and consent for photography. Security measures, such as linking facilitators to specific units and using Pheme API for authentication, will ensure that only authorized personnel have access to sensitive information.

3.2 Functional Requirements

- User (session facilitator) account login
- Session configuration
- Student sign in/sign out by student number, name and/or preferred name
- Sign out entire class capability
- Record of student consent for photography, marks, comments
- Support for multiple units

- Ability to export data to csv file
- Direct communication with Calista database
- Student sign in/sign out by student ID card barcodes
- Sync functionality to sync web app's local data with remote database

3.3 Nonfunctional Requirements

3.3.1 User Interface and Human Factors

- Types of users: Facilitators, professors, and other university staff.
- Training: Minimal training required, focusing on ease of use.
- Ease of learning: High priority for an intuitive interface.
- Error protection: High priority, the application must robustly handle and gracefully recover from exceptions, ensuring stability and continuity.
- Input/output devices: Designed for use primarily on mobile devices.

3.3.2 Documentation

- Types of documentation: User manuals, technical documentation.
- Audience: Professors/Unit coordinators, IT staff, and developers.

3.3.3 Hardware Consideration

- Target hardware: Devices running iOS, Android, and desktops.
- Characteristics: Compatibility with varying screen sizes and storage capacities.

3.3.4 Performance Characteristics

- Speed, throughput, and response time: Optimized for quick data entry and retrieval.
- Size/capacity constraints: Optimized for limited storage and high user concurrency.

3.3.5 Error Handling and Extreme Conditions

- Input errors: User-friendly error messages and guidance.
- Extreme conditions: Offline mode with local data storage.
- Cybersecurity: Protection against SQL/database injection, encryption of private student information.

3.3.6 System Interfacing

- External input: Integration with external database.
- Output: Exportable CSV-style databases for unit coordinators.
- Format restrictions: Standardized data formats for interoperability.

3.3.7 Quality Issues

- Reliability: High reliability with fault trapping.
- Restart time: Minimal acceptable restart time after failure.
- Downtime: Low acceptable downtime per 24-hour period.
- Portability: Compatibility across different hardware and operating systems.

3.3.8 System Modifications

- Expected modifications: No updates required. The application is designed for static use by Professor Adrian Keating without the need for ongoing maintenance or security patches.

3.3.9 Physical Environment

- Operation locations: University premises.
- Environmental conditions: Suitable for standard indoor settings, laboratory environments, and outdoor usage.

3.3.10 Security Issues

- Data/system access control: Secure login and session management.
- Physical security: Ensure secure access to devices and data.

3.4 Constraints

- Programming language: Constraints based on existing infrastructure and team expertise.
- Development environment: Constraints due to available tools and technologies.
- Library usage: Preference for widely used and well-supported libraries.
- Legacy systems: Integration with existing university systems.