**Requirements Analysis Document**

**CITS3200 Group 27 – UWAttend Project**

**Revision History:**

Version R0.1 8/08/24 Isabella Rowe. Created

Version R0.2 [date] [name]. revised template

**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 database that is not connected to UWA’s systems.

**3.0 Proposed System**

**3.1 Overview**

The proposed UWAttend web app will streamline the process of attendance tracking by providing a web-based solution that integrates with the university's Calista database. The app will include pages for login, session management, and class lists. It will also feature functionality for signing in/out students, updating student information, submitting grades, and syncing data with a remote database.

**3.2 Functional Requirements**

* Login page with a potential remembering function.
* Facilitator session login for unit and semester-based sessions.
* Sign students in/out of lab sessions
* Ability to change the status of students’ consent for photography
* Ability to sync the information with UWA’s central Calista 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: Essential to prevent and handle user errors.
* Input/output devices: Designed for use on mobile, tablet, and desktop devices.

**3.3.2 Documentation**

* Types of documentation: User manuals, technical documentation.
* Audience: Facilitators, 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: ??

**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.

**3.3.6 System Interfacing**

* External input: Integration with Calista (UWA 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: Regular updates for new features and security patches.

**3.3.9 Physical Environment**

* Operation locations: University premises, remote locations.
* Environmental conditions: Standard indoor environments.

**3.3.10 Security Issues**

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

**3.3.11 Resource Issues**

* Backup frequency: Regular automated backups.
* Responsible parties: IT staff for backup, installation, and maintenance.

**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.