Software Requirements Specification

for

Smart Student Attendance System

Version 1.0 approved

Prepared by Jordyn Brosemer, Duc Nguyen, Bowang Yang, Yongqiang Zuo

Team 3

November 7, 2022

Table of Contents

1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 1
1.5 References 1
2. Overall Description 1
2.1 Product Users 1
2.2 Operating Environment 2
2.3 Assumptions and Dependencies 2
3. Functional Requirements 2
4. Nonfunctional Requirements 2
5. Other Requirements 2

Revision History

Name	Date	Reason For Changes	Version
Jordyn Brosemer, Duc Nguyen, Bowang Yang, Yongqiang Zuo	11/7/22	First SSRS Implementation	1.0

1. Introduction

1.1 Purpose

This document is designed to describe the performance of a simple class attendance interfacing system so professors can easily mark if students are in lecture or not. This attendance system will be implemented on an android app in Java. In this document the requirements nonfunctional/functional that determine the system's operation.

1.2 Document Conventions

This document is written in Arial/Times fonts. The headers are bold and written in times and the paragraph sections are written in Arial. Every requirement listed in this document will be bolded, and sub-requirements will make up priority of larger higher level requirements.

1.3 Intended Audience and Reading Suggestions

This document should be read by developers who are intending on implementing a similar system, testers and users who want to know how the app works, PM's who need to develop their own requirements or documentation writers looking for examples.
The rest of this document contains a description of the product as well as the scope and requirements list for the app.

1.4 Product Scope

This application is designed to aid in instructors ability to take attendance of classes. Using this instructors should be easily able to mark the attendance of the students in the class without interrupting the class itself as learning always comes first. Instructors should be able to mark attendance quicky and check the overall attendance of one student across a semester or one class across a semester.

1.5 References

Referenced this app:
https://play.google.com/store/apps/details?id=com.ferid.app.classroom&hl=en_US&gl=US

2. Overall Description

2.1 Product Users

This app is designed for class instructors students should not be able to mark their own attendance.
Registrar offices potentially would use this application to add overall classes.
IT Management would need to interface with this for cloud storage or server access.

2.2 Operating Environment

This application is specifically written in Java for use on android devices.

2.3 Assumptions and Dependencies

This application will be developed using android studio and emulating an android application on a computer. Physical testing of the application on a phone itself may be unlikely.

3. Functional Requirements

1. *Instructors should be able to add classes that they are teaching to the application*
2. *Instructors should be able to add students to a class they are teaching*
3. *Instructors should not be able to add a student to a class they are not teaching*
4. *Instructors should be able to remove a student from a class they are teaching*
5. *Instructors should be able to mark a student as absent for a specific lecture*
6. *Instructors should be able to view attendance for different days through the semester for a specific class.*
7. *Instructors should be able to view the attendance for a specific student for a class they are teaching through the semester.*
8. *All application data should be stored on the user's phone.*
9. *Instructors should be able to remove a class from their course list.*
10. *Instructors should be able to revise previous lectures and mark a student as absent/present.*
11. *Instructors should be able to transfer the data file to server.*
12. *Instructo can search information using course number, course name, student name, student ID.*
13. *Instructors can see graphical data for each student.*
14. *All the data can be transfer to other mobile device via Bluetooth.*
15. *The system must give the instructor the ability to check the absence date for specified period.*
16. *If the instructor is trying to check the absence date that is out of the attending period, like checking dates during the holidays, the system needs to give users error messages that indicate that the input dates are not valid.*
17. *If the students are not signed up for the class or on the attendance sheet, the system needs to give the user an error message that indicates that the students are not on the list*

4. Nonfunctional Requirements

1. *If a student role exists, they should not be able to access attendance database.*
2. *Instructors should not be able to view other instructor's classes.*
3. *If 100 instructors use the app simultaneously the performance of the app should not be dramatically affected.*
4. *Application should work successfully 99% of the time. MTBF should be millions of years.*
5. *Application code should be modular and self-documenting to ease in viewing and maintainability.*
6. *Ensure data security and prevent data or privacy breaches because he system should have an authentication process that only let authorized individuals check and use their data.*
7. *The app takes up as little space as possible.*

8. *The app should have good testability.*
9. *The system should have a user-friendly interface and platform in which users of any age or expertise can use.*
10. *The app should have good scalability.*
11. *The app should be easy to use without guide.*
12. *The system should have a feedback system that ask the user for improvements.*
13. *The system should always be ready to use on mobile as there is no need for laptop or computer in every class.*
14. *Data between students and faculty can be synchronized within 1 second.*
15. *The system should be compatible with Android.*
16. *After the end of the course, the system will save the relevant data for at least one month before it can be cleared.*
17. *Users do not need knowledge of database or programming to use the system.*

5. Other Requirements

1. *Mobile Phone with Android system.*
2. *A laptop or PC with platform for Java programming.*