Student Information Management System

Specification Document

`Brian Campos, Ben Herrera, Christina Havel and Mark Stenmark

CS3420 Software Engineering

Table of Contents

[Introduction 3](#_Toc479831823)

[*Purpose* 3](#_Toc479831824)

[*Document Conventions* 3](#_Toc479831825)

[*Scope* 3](#_Toc479831826)

[*References* 3](#_Toc479831827)

[Overview 4](#_Toc479831828)

[*Product Perspective* 4](#_Toc479831829)

[*Product Features* 4](#_Toc479831830)

[*User Classes and Characteristics* 4](#_Toc479831831)

[*Operating Environment* 4](#_Toc479831832)

[*Design and Implementation Constraints* 4](#_Toc479831833)

[*User Documentation* 5](#_Toc479831834)

[*Assumptions and Dependencies* 5](#_Toc479831835)

[Glossary 5](#_Toc479831836)

[System Features 5](#_Toc479831837)

[Interface Requirements 7](#_Toc479831838)

[*User Interfaces* 7](#_Toc479831839)

[*Hardware Interfaces* 7](#_Toc479831840)

[*Software Interfaces* 7](#_Toc479831841)

[*Communications Interfaces* 7](#_Toc479831842)

[Non-Functional Requirements 8](#_Toc479831843)

[*Performance Requirements* 8](#_Toc479831844)

[*Safety Requirements* 8](#_Toc479831845)

[*Security Requirements* 8](#_Toc479831846)

[*Software Requirements* 8](#_Toc479831847)

[Analysis Models 8](#_Toc479831848)

# Introduction

## Purpose

The intended purpose of the Student Information Management System is to provide a simple interface for university students and administrators to access and modify student information. Student information refers to name, IDs, courses, exam grades and registered courses. In addition to that, the SIMS will hold account and password data for the student and administrator access.

## Document Conventions

Place holder

## Scope

Students, teachers, and university administrators will use the SIMS to store important and private information belonging to the students. The Teachers/administrators will be able to add and modify student exam grades and ID/names as well as calculate each student’s GPA. Students will be able to view exam grades, and calculate their GPA as well as view a list of their registered courses. Students will NOT be able to view another student’s information, or make changes to data. This is a security requirement that will be emphasized later in this document.

## References

Use case docs, textbooks

# Overview

## Product Perspective

This software product is to be used as a replacement for the prospective school’s previous system. The software will be a unit within itself, and will be accessed via a web host.

## Product Features

Students, and administrators will be able to easily manage a student’s information. Administrators will be able to add and modify test grades of their students, as well as calculate the student’s GPA. Administrators will be able to modify student information including their name, and ID if it is found to be incorrect. Students will have access to only their information. Students will have viewing rights only. Students will be able to view their classes that they are registered for as well as their test grades in each class and a cumulative GPA.

Insert data flow diagram

## User Classes and Characteristics

The Student Information Management System will consist of two separate user classes—student and administrator. Student will consist of all users that are individuals that are registered in classes at the school. Administrators will consist of both teachers and the school faculty. Students will have viewing access only, and will only be able to view their own information. Administrators will have view, add, and modify abilities. Each class of users will be required to log in to the system to ensure security and privacy of the information.

## Operating Environment

The SIMS will reside on a web server in order for it to be accessed from anywhere. The database that holds all of the information, will also reside on a web server. This will allow for both components to communicate over the web as well as create fluidity by allowing anyone access to the software no matter what type of computer they have. In the future, there may be a need to create a mobile application for the SIMS, in which case Android and Apple operating systems would be used to host.

## Design and Implementation Constraints

Place Holder

## User Documentation

We have not created a user manual at this time

## Assumptions and Dependencies

Since the SIMS will be hosted on webservers, it will be dependent on the outside service functionality. The hostname and IP address will need to stay static to ensure all of the components have a reliable communication path.

# Glossary

|  |  |
| --- | --- |
| SIMS | Student Information Management System |
| Software |  |
| GUI | Graphical User Interface |
| Database |  |
| User | Any person, student or teacher that actively engages with the software |

# System Features

1. Administrator Login
   1. Access log n page
   2. Admin types in login credentials
   3. Clicks ‘Login’
   4. Admin’s account is now accessible
   5. Exit
2. Modify Student Name
   1. Access login page
   2. Student types in login credentials
   3. Clicks ‘Login’
   4. Student’s account is now accessible
   5. Exit
3. Modify Student ID
   1. Admin logs into account
   2. Admin clicks on button ‘students’
   3. Admin searches student by ID/name
   4. Once found, admin clicks ‘edit’
   5. Changes to student’s name can now be made
   6. Make specified changes
   7. Save
   8. Exit
4. Modify Student Courses
   1. Admin logs into account
   2. Admin clicks on button ‘courses’
   3. Admin searches student by ID
   4. Once found, admin clicks ‘edit’
   5. Changes to student’s courses can now be made
   6. Make specified changes
   7. Save
   8. Exit
5. Modify Exam Scores
   1. Admin logs into account
   2. Admin clicks on button ‘courses’
   3. Admin searches student by ID
   4. Once found, admin clicks ‘view’
   5. Exams button is clicked
   6. Edit is now clicked
   7. Make specified changes
   8. Save
   9. Exit
6. Modify GPA Calculation
   1. Admin logs into account
   2. Admin clicks on button ‘students’
   3. Admin searches student by ID/name
   4. Once found, admin clicks ‘edit’
   5. GPA can now be accessed for editing
   6. Make specified changes
   7. Save
   8. Exit
7. Retrieve Student Name
   1. Log into system with student or admin ID and password
   2. Admin/Student selects ‘Students’ on menu
   3. Type in student ID
   4. Student name is displayed
   5. Exit
8. Retrieve Student ID
   1. Log into system with student or admin ID and password
   2. Admin/Student selects ‘Students’ on menu
   3. Type in student name
   4. Student ID is displayed
   5. Exit
9. Retrieve Registered Courses
   1. Log into system with student or admin ID and password
   2. Admin/Student selects ‘Courses’ on menu
   3. Type in student ID
   4. Student’s registered courses are displayed
   5. Exit
10. Retrieve Exam Scores
    1. Log into system with student or admin ID and password
    2. Admin/Student selects ‘Courses’ on menu
    3. Type in student ID
    4. Type in course ID
    5. Student exam courses are displayed
    6. Exit
11. Retrieve GPA Calculation
    1. Log into system with student or admin ID and password
    2. Admin/Student selects ‘Courses’ on menu
    3. Type in student ID
    4. GPA is displayed
    5. Exit

# Interface Requirements

## User Interfaces

## Hardware Interfaces

## Software Interfaces

Database requirements🡪MySQL Server 2016

The SIMS will be programmed in C++ in Microsoft Visual Studio

## Communications Interfaces

Access to the SIMS will be given through the HTTPS protocol, to ensure security of student information. A web browser will be required.

# Non-Functional Requirements

## Performance Requirements

The SIMS is expected to be available at least 90% of the time. In order to ensure minimal downtime, the robustness of the software will be tested in many circumstances including large data uploads.

## Safety Requirements

## Security Requirements

By using secure servers and encryption, we can ensure that the software safeguards the student’s information. Passwords will be required to be complex, requiring a minimal of 8 characters, 1 number, 1 capital, and one symbol. Passwords will be required to be reset every 3 months.

## Software Requirements

Robustness will be ensured by utilizing servers with redundancy models, to minimalize down time.

# Analysis Models