Software Requirements Specification

for

GWA Verifier

Version 1.0 approved

Prepared by CMSC 128 C-3L

June 06, 2022

Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

GWA Verifier Version 1.0 is a technology geared to automate the filtering process of the CAS Scholarships, Honors, and Awards Committee (SHAC) in its responsibility to nominate honorable students eligible for CAS Awards. The application aims to eliminate the tedious process of SHAC's manual grade assessment.

1.2 Document Conventions

The title page of this document is written in Arial font, sizes 32 and 20 for the title, and 14 for the subtitles. Headers are written in bold Times, with section headers at size 18 and subheaders at size 14. The rest of the text in this document is in Arial, size 11.

1.3 Intended Audience and Reading Suggestions

This document is intended for the project developers and users (CAS Scholarships, Honors, and Awards Committee members). Before reading this document, it is highly recommended that the reader familiarize themselves with the grading system of UP. It is also suggested that the reader become knowledgeable of the 2018 Curricula for degree programs in CAS.

1.4 Project Scope

The product aims to provide a means to store and manage a record of students' information and grades in the courses that they have taken. In the process, the product is able to check the validity of the data being stored. This includes the ability to verify students' GWA and the ability to check if a student has qualified for honors. It would not cover complicated cases involving the need for human consideration such as, but not limited to, student involvement in academic dishonesty, civil or criminal cases, etc.

The product's current iteration is only compatible with CAS degree programs in their 2018 curriculum. The product only supports document types of .csv, .xlsx, and .pdf for file uploads. It also does not have the ability to verify if a specific course code is a valid or existing course code within UPLB. In addition, it cannot handle degree programs with specialization such as BS Agricultural Chemistry, BA Communication Arts, etc.

1.5 References

The style and formatting of this document uses a template copyrighted by Karl E. Wiegers, permitting use, modification, and distribution of the document.

2. Overall Description

2.1 Product Perspective

This system was selected and assigned to CMSC 128 C-3L by Asst. Prof. Reginald Recario and Asst. Prof. Miyah Queliste. This is designed as a tool to aid SHAC in its corporate responsibilities. The product is a first of its own and is not a follow-on member of any product family.

2.2 Product Features

The system will:

- *Allow user authentication,*
- *Allow authorized users to create and manage users,*
- *Allow authenticated users to view student records from the system's database,*
- *Allow authenticated users to upload student records to the system's database,*
- *Checks the accuracy of the computations related to the student's grades,*
- *Checks the eligibility of the student for latin honors,*
- *Allow authenticated users to download a summary of students eligible for latin honors,*
- *Allow authenticated users to modify student records,*
- *Allow authenticated users to remove student records,*
- *Log all activities made by authenticated users,*
- *Allow authorized users to view activities made by users, and*
- *Allow authorized users to view edit history of users.*

2.3 User Classes and Characteristics

The members of the Scholarships, Honors, and Awards Committee (SHAC) will be the people who will make use of this system. The classes of users that will be using it are the standard class, and the administrator class of users.

The administrator class consists of a chair or head from one of the unit representatives, a representative from the College of Arts and Sciences - Office of the College Secretary (CAS-OCS), and the Assistant College Secretary (ACS), as the ex-officio. This class has the most privileges when they make use of this software. The standard class of users are able to add to the list of students, view the entire list including those who are qualified for graduation, and add, edit, and view student records.The administrator class can do all of these, but they can also view the full list of users of the product, view a full log of activities done by the users, and delete existing users.

2.4 Operating Environment

The operating environment of this software requires the installation and the setting up of NodeJS, the React javascript library, npm, and MariaDB, in order to function as intended. It can work in a laptop or a computer with any of the three main Operating Systems; namely Windows, Linux, and iOS. There are no other software requirements that are needed in order to run this software.

2.5 Design and Implementation Constraints

The product is implemented as a web application whose core components are the backend, frontend and database. The React javascript library is used to implement the user interface. The NodeJS runtime environment is used to enable product features such as file parsing, database queries, User sign-up etc. The MariaDB database management system is used to create the database component of the product. The product is expected to run optimally as long as the above-mentioned components are up to date.

2.6 User Documentation

There is currently no available user manual for the system.

2.7 Assumptions and Dependencies

The system will be web-based and will be deployed to a local network. It can currently only be accessed locally, but may become available on the Internet. It is also assumed that the devices used can handle file uploads and downloads. For the uploaded files, it is assumed that every course being processed has a Course Number, Grade, Units, Weight, Cumulative and Term except for special cases such as LOA and AWOL. Comments regarding the actual number of units per semester is optional.

3. System Features

3.1 User Authentication

3.1.1 Description and Priority

High priority is given to this feature. It includes basic user login and logout functionalities.

3.1.2 Stimulus/Response Sequences

- *The user enters valid credentials and clicks the login button to be redirected to the main page of the system.*
- *The user clicks the logout button to be redirected to the login page.*

3.1.3 Functional Requirements

REQ-1: *There must be form validation.*
REQ-2: *A warning message must appear to inform the user of an unsuccessful login.*

3.2 File Upload and Processing

3.2.1 Description and Priority

High priority is given to this feature. It includes the uploading and processing of a given file into data to be added to the product's database.

3.2.2 Stimulus/Response Sequences

- *The user selects a file (or multiple files) to be uploaded. Files that are in xlsx, csv, and pdf formats are able to be uploaded and processed, as long as they follow a specific format.*
- *The user is able to see if the files that they have uploaded resulted in any warnings or errors.*
- *The user can now refresh the student list, to see the students that they have added to the database.*

3.2.3 Functional Requirements

REQ-1: *File upload must work properly.*
REQ-2: *Warnings and errors must appear and inform the user if there are issues encountered during file processing.*

3.3 List of Students View, Sort, & Search

3.3.1 Description and Priority

High priority is given to this feature. The feature allows a user to view the students in separated tabs along with their names, student numbers, degree programs, and GWA. The students can be sorted by their last name, student numbers, and GWA. It also includes a search bar to display only the student(s) applicable to the content of the search input. The feature also allows the user to click on the 'Show Qualified' button to display only the students eligible for latin honors based on the courses they took and their GWA. The

feature also allows the user to click on a student tab's 'View Record' button to view the record of the selected student.

3.3.2 Stimulus/Response Sequences

- *The user can see the students sorted by their last names by default which were uploaded into the database, as well as sort and search options along the top portion of the display.*
- *The user clicks a sort option to change how students are sorted in the display.*
- *The user clicks the 'Default' button to revert any display changes.*
- *The user clicks the 'Show Qualified' button to display all students who are eligible for latin honors.*

3.3.3 Functional Requirements

REQ-1: *There must be student data in the database.*
REQ-2: *Student data in the database must be complete and correct.*
REQ-3: *Clicking the 'Qualified' button with no eligible students in the database will not display any student tab.*

3.4 Student Record View, Edit, & Delete

3.4.1 Description and Priority

High priority is given to this feature. The feature allows a user to view a student's record of courses taken along with the grades, units, and resulting weights separated into tables by terms/semesters. The student record page also displays a student's name, student number, degree program, and GWA. It also allows a user to modify a student's record by changing parts of the record or adding to the record. The feature also allows a user to delete the student's record from the database.

3.4.2 Stimulus/Response Sequences

- *The user can see the student's record.*
- *The user can click on the 'Edit' button for the whole record to start editing the student's record.*
- *The user can continue to click on a table or row's 'Edit' button to edit courses taken by a student.*
- *The user can click on a table's 'Add' button to add a single row for the term/semester.*
- *The user can click on the 'Add Table' button to add a new term/semester for the student record.*
- *The user can click on the 'Delete' buttons to delete a row, table, or the whole student record.*
- *The user can click on the 'Cancel' buttons to undo certain changes made by the user.*
- *The user can click on the 'Submit' buttons to confirm changes made by the user, but not yet saving the changes to the database.*
- *The user can click on the 'Save' button to upload the student's record along with the changes made by the user to the database.*

3.4.3 Functional Requirements

REQ-1: *The student's records must be in the database.*
REQ-2: *The student's records must be complete and correct.*

3.5 Download Summary

3.5.1 Description and Priority

Low priority was given to this task. It takes all qualified students from the database table 'qualified' and appends them to a pdf file. The pdf's data is arranged in table format.

3.5.2 Stimulus/Response Sequences

- *The user clicks download and is given a pdf of all qualified students.*

3.5.3 Functional Requirements

REQ-1: *Database must have data in it.*
REQ-2: *Nothing will be stored in the pdf if the students are all unqualified.*

3.6 Admin Dashboard

3.6.1 Description and Priority

High priority is given to this feature. This feature is only accessible to users of the "admin" type. In this page, an admin user can create new users or accounts that could access the app, and manage the existing accounts accessing the app. Also, in this page, an admin user could view all his/her own activities, and edit history, as well as, view all the activities, and edit history of all the accounts.

3.6.2 Stimulus/Response Sequences

- *By default, when visiting the admin page, it shows the 'manage users' tab where the user can view, and manage the accounts that have access to the app.*
- *The user can click the 'Activity History' button of an account to view the activity history of the selected account.*
- *The user can click the 'Edit History' button of an account to view the edit history of the selected account.*
- *The user can click the 'Update' button of an account to change the type of an account either from 'admin' to 'member' or 'member' to 'admin'.*
- *The user can click the 'Delete' button of an account to delete the selected account.*
- *The user can click the 'View all activities' option in the navigation bar to view all the activities of all users.*
- *The user can click the 'View all edit history' option in the navigation bar to view the edit history of all users.*
- *The user can click the 'View my activities' option in the navigation bar to view the user's list of activities.*
- *The user can click the 'View my edit history' option in the navigation bar to view the user's list of edit history.*

3.6.3 Functional Requirements

REQ-1: *The users' accounts must be in the database.*
REQ-2: *The users' activities must be reflected in the database.*
REQ-3: *The users' activities in the edit page must be reflected in the database.*

4. External Interface Requirements

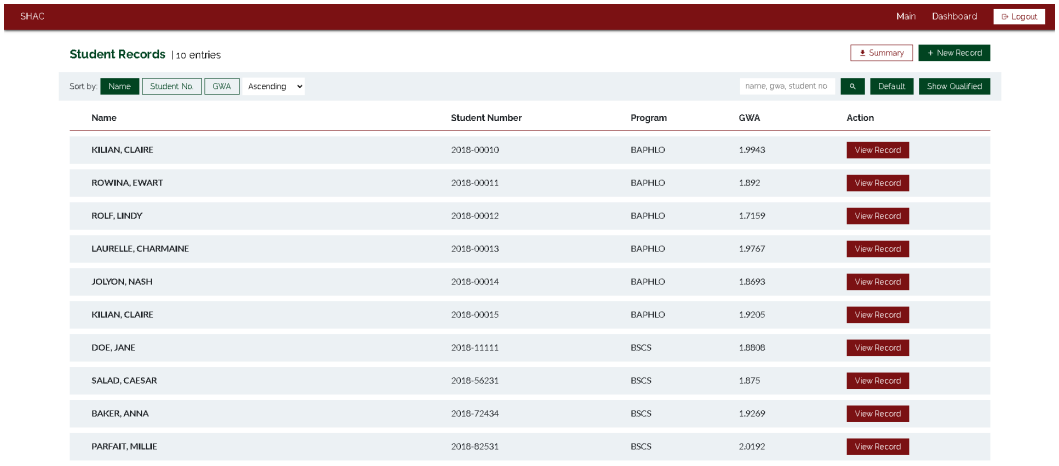
4.1 User Interfaces

The software will have a minimalistic design and UI elements will be displayed in a straightforward manner. It will use React Bootstrap design templates for modals, toasts, navigation, and other interface components. It will also make use of React Icons to enhance the user interface and user experience.

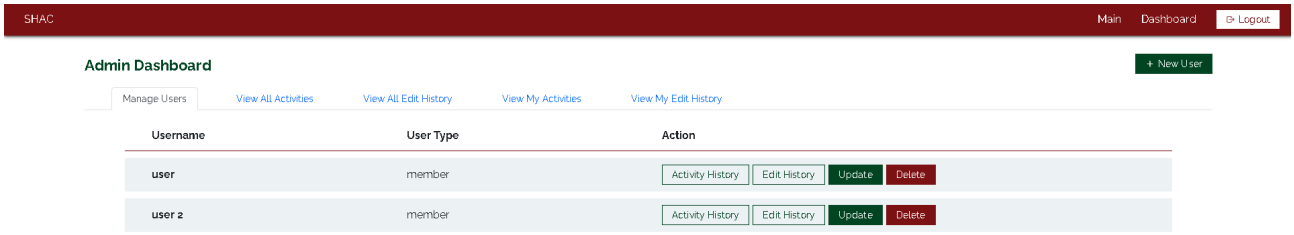
Shown below are screenshots for the different pages of the system:

Login Page

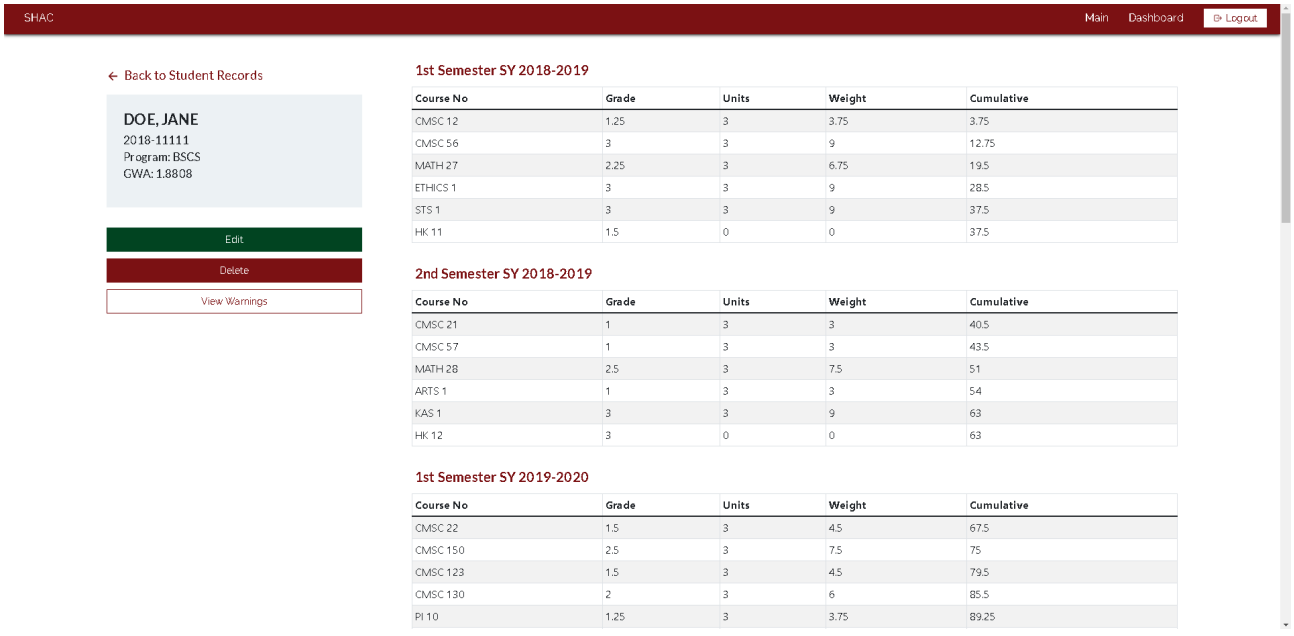
Main Page



Admin Dashboard



Student Record



4.2 Hardware Interfaces

The back-end and database components are both hosted in the same server computer. Client computers can connect to the network and use a standard browser to access the system. A two-way communication using JSON-formatted data is set up between the server and client.

4.3 Software Interfaces

For the project, MySQL is used as the database management system, while Express is used as the server framework. Moreover, the Node.js backend communicates with ReactJS via JSON.The front-end and back-end components are not stand-alone and are expected to be run together.

4.4 Communications Interfaces

There are no communication functions needed for this product.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must be accessible in the local network. It must perform tasks quickly, make accurate computations, and provide correct data.

5.2 Safety Requirements

There are no safety requirements needed for this product.

5.3 Security Requirements

Security is implemented for user authentication. MD5 encryption is used to safely store passwords in the database.

5.4 Software Quality Attributes

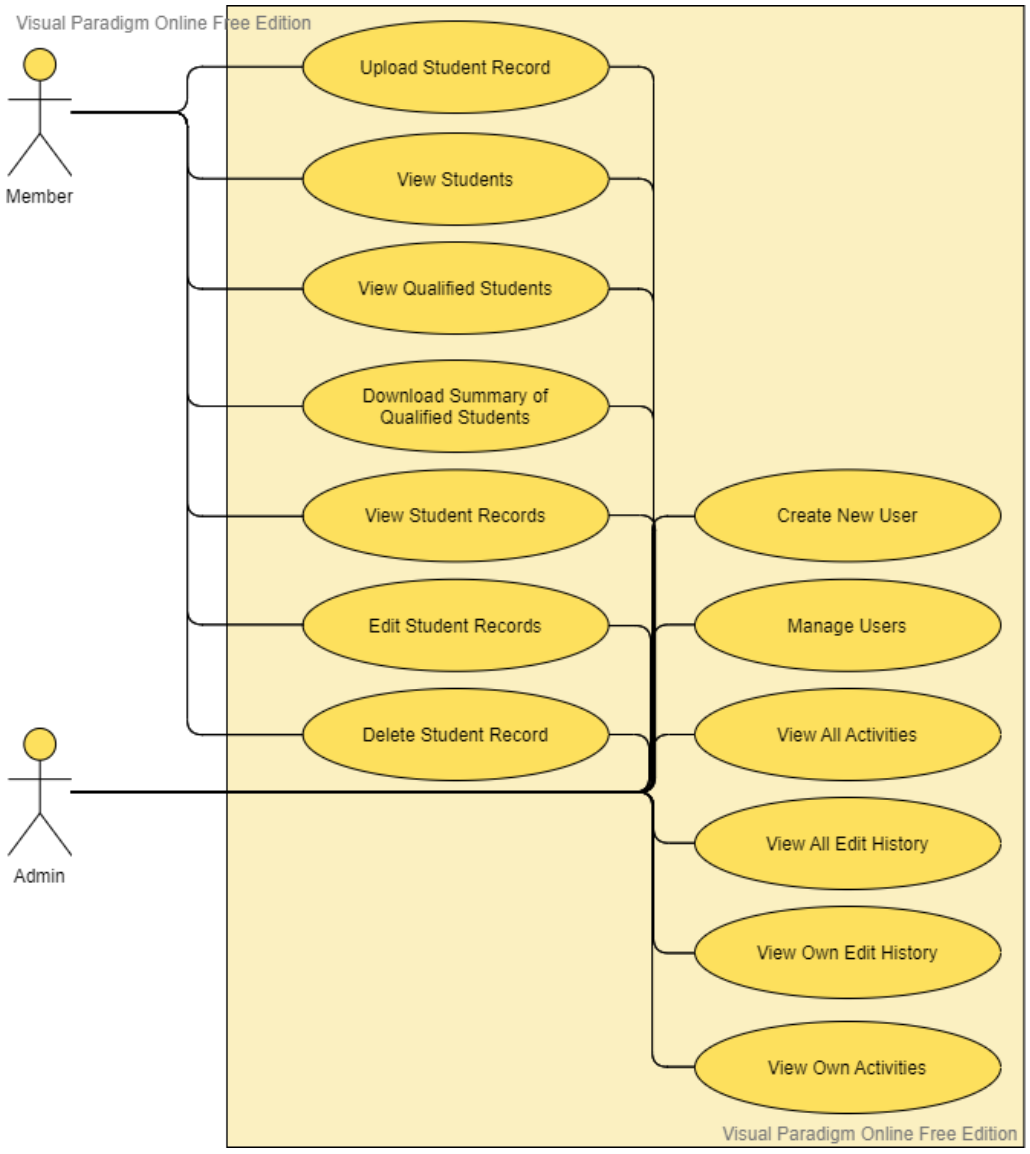
Key attributes for this system are: correctness, usability, and reusability. The system must process the data correctly at all times to provide accurate information for the users. Usability is valued in order to provide our clients with the best user-experience in terms of ease of usage and design. It is also imperative that the system can be used repeatedly given numerous batches of graduating students that are produced per year. Other notable attributes for this system are its adaptability and maintainability considering the long-term necessity of this system for the users and the potential addition of future courses and degree programs.

Appendix A: Glossary

- *GWA - General Weighted Average*
- *SHAC - Scholarships, Honors, and Awards Committee*
- *ICS - Institute of Computer Science*
- *CAS - College of Arts and Sciences*
- *UPLB - University of the Philippines Los Baños*
- *LOA - Leave of Absence*
- *AWOL - Absent without Leave*

Appendix B: Analysis Models

Use Case Diagram

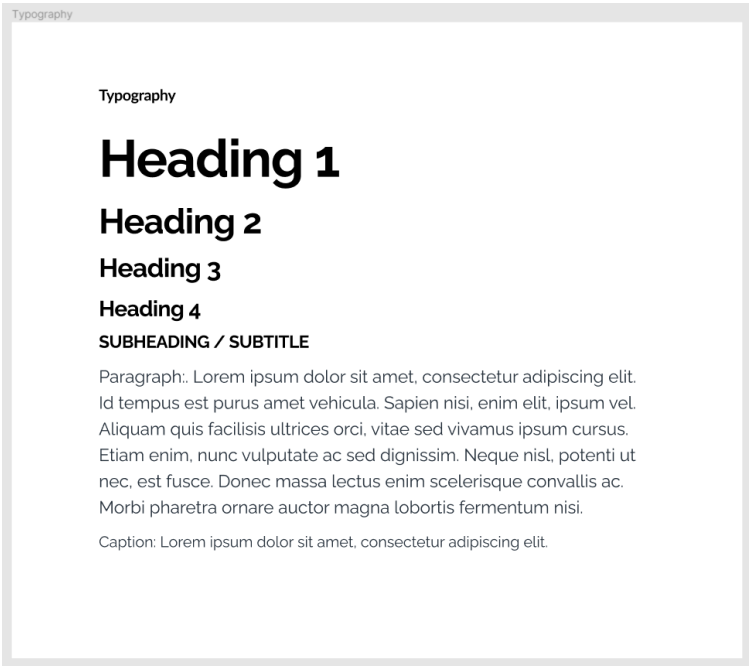


Appendix C: Issues List

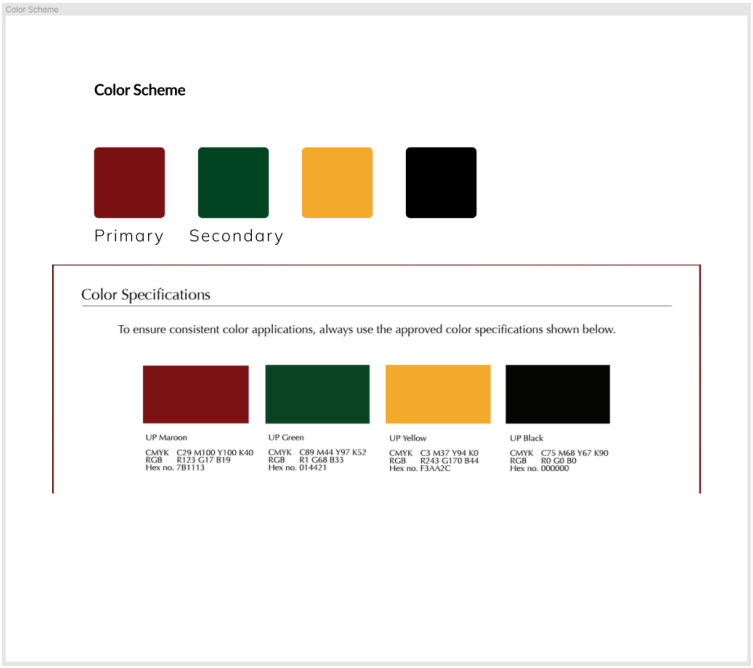
1. *PDF files are converted into excel files using an npm package called pdf-to-excel. The conversion, however, is not accurate when empty cells exist. This results in PDF becoming more prone to errors compared to .csv and .xlsx files.*
2. *In PDF files, text that is centered in a cell and merged cells are not read properly.*
3. *For .csv files, student numbers from 20XX-00001 to 20XX-00012*
4. *After uploading files, changes are not reflected until the user refreshes the page.*
5. *After editing a student file, the GWA calculation presented on the page does not reflect the actual GWA of the student. GWA will be reflected properly once the user goes back to the Main page.*
6. *Newly created users cannot change their registered passwords.*
7. *For adding a new table in a student record, the user needs to add another row for the new table to be able to input data in the Grade and Units column.*
8. *On the admin dashboard, viewing the edit history and activity history of a certain user does not display the expected results.*

Appendix D: Wireframes and Design

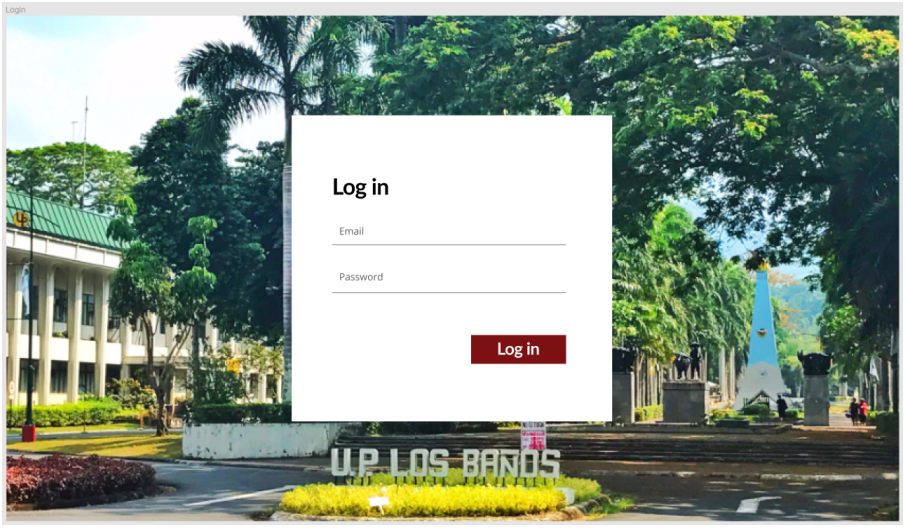
Typography



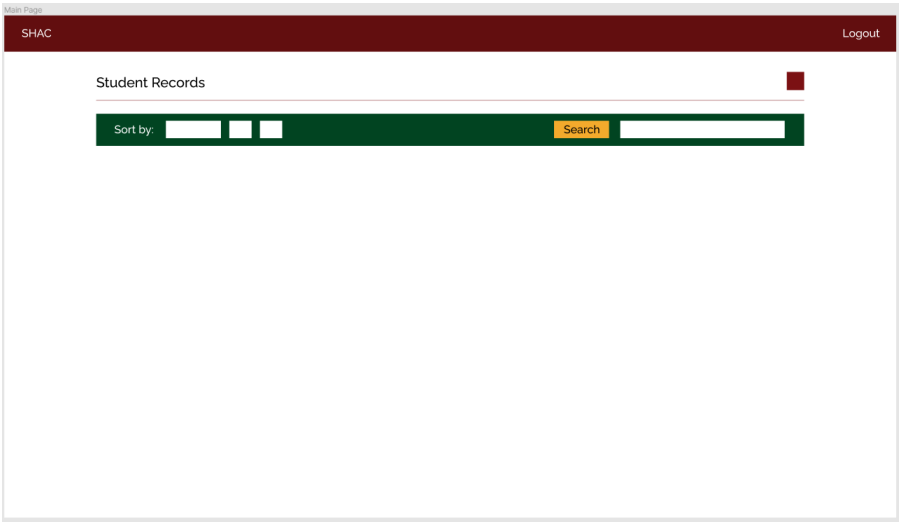
Color Scheme



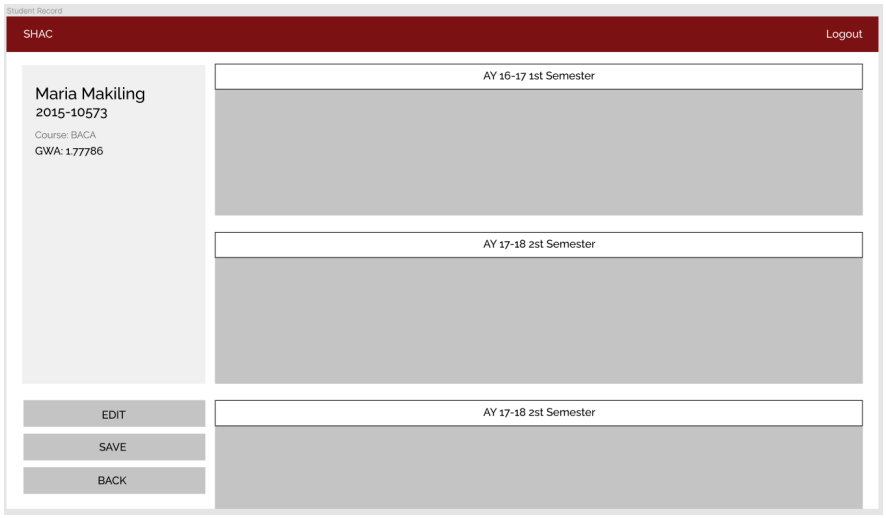
Login Page



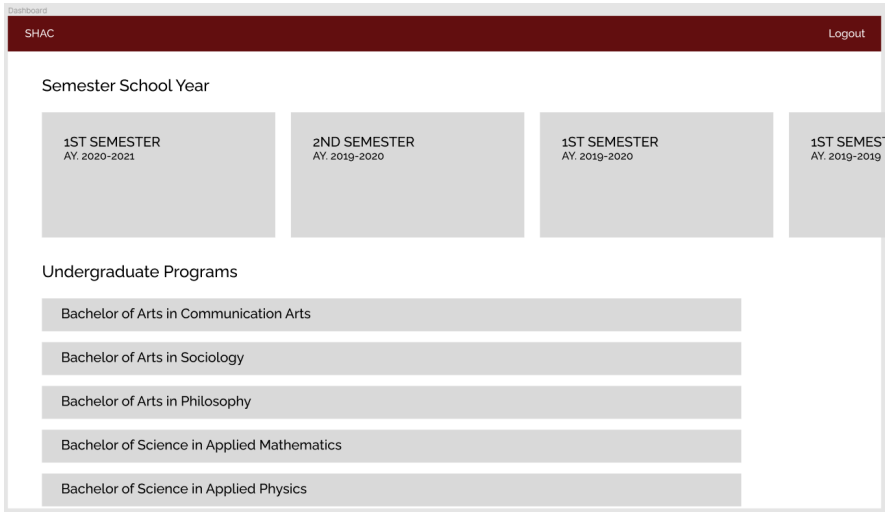
Main Page



Student Record Page



Dashboard Page - This is where the user should be redirected to after logging in, however, it was not implemented.



No wireframe/design made for admin page

Appendix E: Database Schema

Database Tables

Tables_in_final_db
activities
edit_history
students
taken_courses
users

users table

Field	Type	Null	Key	Default	Extra
Username	varchar(30)	NO	PRI	NULL	
Password	char(32)	YES		NULL	
Type	varchar(10)	YES		NULL	

students table

Field	Type	Null	Key	Default	Extra
ID	varchar(10)	NO	PRI	NULL	
First_Name	varchar(20)	YES		NULL	
Last_Name	varchar(20)	YES		NULL	
Program	varchar(10)	YES		NULL	
GWA	decimal(5,4)	YES		NULL	
Qualified	int(1)	YES		NULL	
Warnings	text	YES		NULL	

taken_courses table

Field	Type	Null	Key	Default	Extra
ID	int(5)	NO	PRI	NULL	
Student_ID	varchar(10)	NO	PRI	NULL	
Course_Code	varchar(20)	YES		NULL	
Course_Type	varchar(10)	YES		NULL	
Grade	varchar(4)	YES		NULL	
Units	varchar(4)	YES		NULL	
Weight	decimal(4,2)	YES		NULL	
Term	varchar(20)	YES		NULL	

activities table

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Username	varchar(30)	NO	PRI	NULL	
Action	text	YES		NULL	
Date	timestamp	NO		CURRENT_TIMESTAMP	

edit_history table

Field	Type	Null	Key	Default	Extra
ID	int(5)	NO	PRI	NULL	
Username	varchar(30)	NO	PRI	NULL	
Student_ID	varchar(10)	YES		NULL	
Datetime_of_edit	datetime	YES		NULL	
Edit_notes	text	YES		NULL	