Software Requirements Specification

for

SHACker

Version 1.0 approved

Prepared by CMSC128 - D4L

University of the Philippines - Los Baños

June 9, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Almendrala, Austria, Avila, Pascua	2022-05-02	Initial product features	0.1
Ceradoy	2022-05-17	Analysis model	0.2
Duhaylungsod, Fuentes	2022-05-27	Initial draft of section 1 and 2	0.3
Almendrala, Arguelles, Avila, Justo, Lee	2022-06-07	Writeup of section 1 to 3	0.4
All members	2022-06-08	Product features and interfaces	1.0

1. Introduction

1.1 Purpose

This document aims to introduce SHACker Version 1.0: a system intended for the College of Arts and Sciences (CAS) Scholarships, Honors, and Awards Committee (SHAC) to easily examine records of graduating students under the CAS degree programs of the University of the Philippines - Los Baños (UPLB). The SHAC is responsible for verifying the completeness of units taken based on the majors specified in the degree program of a student record, its non-credit courses, electives, specialization courses, and General Education (GE) subjects. The committee also checks the correctness of grades, grade point calculations, and the summation of these grade points for the computation of the General Weighted Average (GWA). Along with more complex processes involved in verifying a student's graduation status, these are checked manually for each candidate student for graduation.

Through SHACker, the committee can easily detect inconsistencies in the student records which speeds up the process and ensures data integrity. Since computations, curriculum, and enrollment status checking from the student record are automated, the SHAC may focus more on cross-validating other supporting documents such as the Underload Permit, Replacement of GE courses form (i.e., CAS OCS Form 014) and GE Plan of Study (i.e., CAS OCS Form 019), among many others.

1.2 Document Conventions

The title page of this document is written in Raleway font, size 32, and 14 for the subtitles. Headers are written in bold Raleway, with section headers at size 18 and subheaders at size 14. Appendix E is written in Raleway, font size 10. The rest of the text in this document is in Raleway, size 12.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification is intended for the users (SHAC) and developers (students of CMSC128 - D4L) of the SHACker system. Reading the document is suggested in order to gain an overview of its scope and limitations found in section 1.4. The different functionalities of SHACker are introduced in section 2.2 and are thoroughly described in section 3. Readers with extensive knowledge of the system may skip through the document, as desired.

1.4 Project Scope

Intended for streamlining the student record verification processes done by UPLB CAS SHAC, the SHACker is deployed locally. The system only accepts student records on CSV and XLSX formats with a valid student number, name, and CAS degree program (See Appendix D). The application is ideally suited for desktop viewing, but it may still be accessed on mobile devices within the same local network.

The considerations for record inconsistency are limited to the information obtained from the student record. Such considerations are elaborated in section 3.14.3. Any additional checks that require information from supporting documents (i.e., shifting, transfer, cross-register, exchange, course substitution, extra courses, etc.) are out of scope.

1.5 References

The Software Requirements Specification of SHACker is based on a template by Karl E. Wiegers. This document is copyrighted, but permission is given to use, modify, and distribute the material, as stated in the title page of this document. The SRS of STADTRA Version 1.0 of CMSC 128 AB-5L, dated 2015 March 23, was retrieved from the CMSC 128 course module and was used as a reference for this document.

2. Overall Description

2.1 Product Perspective

The software, SHACker, is a tool that aims to aid the CAS Scholarships, Honors, and Awards Committee (SHAC) in evaluating and validating the grade records of potential graduating students. It is a new, self-contained product that produces a summary of candidate graduating students verified by SHAC personnel. This system was selected and assigned to CMSC 128 D-4L by Prof. Reginald Recario and was ideated from his experience in the committee mentioned above.

2.2 Product Features

The system will:

- Allow administrator authentication,
- Allow an authenticated administrator to create, delete, or promote regular users,
- Allow regular user authentication,
- Allow an authenticated user to change their name and password,
- Allow an authenticated user to add student records manually,

- Allow an authenticated user to add student by uploading .csv or .xlsx files in solo or bulk,
- Allow an authenticated user to detect formatting and computational inconsistencies in record records,
- Allow an authenticated user to sign and unsign student records
- Allow an authenticated user to edit and annotate student records
- Allow an authenticated administrator to delete student records,
- Allow an authenticated user to view a list of all unsigned, partially signed, and fully signed student records,
- Allow an authenticated user to view a summary of all fully verified student records.
- Allow an authorized user to download the student record summary in .pdf,
- Log any activity made with regards to the student records,
- Allow an authenticated administrator to delete logs

2.3 User Classes and Characteristics

Two user classifications will be implemented: (1) an administrator and (2) a regular user. A regular user can add, manage, check, and sign student records. An administrator has the same permissions but has higher account privileges. Such privileges include adding, deleting, and promoting regular user accounts. An administrator is also granted authorization to manage sensitive data, which includes the ability to delete student records and logs.

2.4 Operating Environment

The system back-end is stand-alone and can run in any environment with NodeJS v.16.14.2 and MongoDB v.5.0.9 installed. The front-end is also stand-alone and will require a client environment to have a browser capable of handling JavaScript.

2.5 Design and Implementation Constraints

The design and implementation of the system will be in strict accordance with the allotted duration of the second semester in the UPLB academic calendar of 2021-2022. Moreover, communication and interaction among developers are constrained by what is permitted under the guidelines imposed for a remote setup. In line with this, the implementation considers the internet connectivity of the programmers, in addition to the hardware constraints of their machines, since development is to be done remotely and separately. Furthermore, testing for local network deployment is limited to those with access to multiple devices. The university's data privacy policy which restricts access to other student documents will also limit the project's design. Upon handing the software to the client, all responsibilities regarding system deployment, revisions, and maintenance are transferred to the client.

2.6 User Documentation

There is currently no user manual available for the system. However, the project source code is fully documented.

2.7 Assumptions and Dependencies

Given the limited data provided to the application, the following assumptions are imposed by SHACker on its data:

- Student records have no history of shifting degree programs.
- Student records have no history of transferring from a different university or UP campus.
- Student records have no history of enrolling courses from a different campus (i.e., cross-registration) or another university (e.g. via exchange program).
- Student records have no history of change in terminal course plan (e.g. changing from Special Problem (SP) to Thesis option midway through implementation).
- Subjects that have been deemed completed by the CAS Office of the College Secretary (OCS) and the Student Academic Information System cannot be retaken in the student record.
- Student record is compliant with a valid CAS offered degree program.
- Checking for completeness of subjects based on a given curriculum is only implemented for student records following the 2018 curriculum.
- Student records with other curriculum versions are still considered in the system but will only have their format and computations checked.
- Thesis unit partitions considered are 2-2-2, 3-3, and 6 units.
- SP unit partitions considered are 1-2 and 3 units.
- Unit partitions for SP and Thesis other than the aforementioned are out of scope.
- Excess placeholder units for Thesis and SP in the student record are not accepted by the system.

The software has been developed with Node Package Manager (NPM) v.8 and requires NodeJS v.16.14.2 and MongoDB v.5.0.9. installed. SHACker is dependent on several npm packages which are listed below alongside their versions.

Module	*Description*
Bcrypt v5.0.1	for password hashing
Cookie-parser v1.4.6	to parse HTTP request cookies
CORS v2.8.5	to handle resource sharing among different origins
Dotenv v16.0.1	to load environment variables from a .env file
Express v4.18.1	to manage the web framework for NodeJS
Generate-password v1.7.0	to generate initial passwords for regular users
Jsonwebtoken v8.5.1	to securely exchange encrypted payload

Mongoose v5.13.14	to access the MongoDB database
XLSX v.0.18.5	for spreadsheet parsing
Jspdf v2.5.2	to download the summary in PDF format
Jspdf-autotable v3.5.25	to format the summary in tabular form
@emotion/react 11.9.0	for simple styling in React
@emotion/styled 11.8.1	for styled API of @emotion/react
@mui/icons-material 5.8.2	to access the icons in Material UI
@mui/material 5.8.2	to access the React components in Material UI
@testing-library/jest-dom 5.16.4	to test the state of the DOM
@testing-library/react 12.1.5	to access React DOM testing utilities
@testing-library/user-event 13.5.0	to interact with the user on a browser
React-dom 18.1.0	entry point to the DOM and server renderers for React
React-router-dom 6.3.0	for using React Router in web applications
React-scripts 5.0.0	includes the scripts and configuration used by Create React App
React 18.1.0	JavaScript library for building user interfaces
Universal-cookie 4.0.4	to use universal cookies in React
Web-vitals 2.1.4	to measure performance metrics in JavaScript

3. System Features

3.1 Administrator Creation

3.1.1 Description and Priority

This is a high-priority feature which provides a mechanism to create user privilege hierarchy. This allows users to create an account which can be used to add, manage, and promote other users for the system.

3.1.2 Stimulus/Response Sequence

- The assigned administrator clicks the “Sign Up” hyperlink and is redirected to the “Create Admin Account” page.
- The administrator enters their first and last name, email, password, and a reconfirmation of their password in the text fields provided.
- The administrator clicks the “Sign Up” button.
- Upon successful sign up, a message appears below the text fields stating that the user was saved and the user is redirected to the “Login” page.
- Upon failure to sign up, a series of warning messages appear below the text field of interest.

- The administrator may choose to click the cancel button. This redirects them to the "Landing" page.

3.1.3 Functional Requirements

- AC1 - Form validation that checks the following must be properly implemented:
    - All input fields must be filled.
    - Email must follow formatting conventions.
    - Password must follow formatting conventions
- AC2 - A message will appear below to prompt the user for a successful signup.
- AC3 - "Signup" page must not be accessible when an administrator account is already existing.

3.2 Add Users (Admin Privilege)

3.2.1 Description and Priority

This is a high-priority feature. It allows the administrator to create a regular user for the application.

3.2.2 Stimulus/Response Sequence

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The administrator clicks the "Manage Users" button.
- The administrator is navigated to the "Manage Users" page.
- In the "Manage Users" page, the administrator clicks the "Add New User" button.
- The administrator enters the first name, last name, and email of the assigned regular user in the text fields provided.
- The administrator clicks the "Save" button.
- Upon successful addition of a regular user, a dialog box showing their randomly generated password is prompted.
- When the administrator clicks the "Okay" button of the dialog box, they are redirected to the "Manage Users" page.
- Upon failure to add a regular user, a warning message will show that the regular user was not saved.
- The administrator may choose to click the cancel button. This redirects them to the "Manage Users" page.

3.2.3 Functional Requirements

- AU1 - A verification function checks for the user's role (i.e., administrator or regular user) to access the "Manage Users" page.
- AU2 - The system will ensure that the application will only add new users with complete information.
    - All input fields must be filled.

    - The email address must be valid, with an email prefix and email domain.
- AU3 - A verification function ensures that no users with the same email address will be added.
- AU4 - Random password generation must be implemented properly.
- AU5 - The newly generated password is displayed in a prompt that will appear only once which cannot be accessed again upon closing.
- AU6 - A message will appear below to prompt about a successful addition of a new regular user.

3.3 Delete Users (Admin Privilege)

3.3.1 Description and Priority

This is a high-priority feature. It allows the administrator to delete an existing regular user and revoke their access to the system.

3.3.2 Stimulus/Response Sequences

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Manage Users” button.
- The user is navigated to the “Manage Users” page.
- In the “Manage Users” page, the administrator clicks the “trash can” icon on the row of the existing regular user to be deleted.
- Upon deletion, the “Manage Users” page will automatically be refreshed.

3.3.3 Functional Requirements

- DU1 - A verification function checks for the user's role (i.e., administrator or regular user) to access the “Manage Users” page.
- DU2 - The regular user removed from the system is found on the same row on which the “trash can” icon was clicked.
- DU3 - The page will reload to display the updated list of all existing users in the system upon a successful delete user operation.

3.4 Promote User (Admin Privilege)

3.4.1 Description and Priority

This is a medium-priority feature. It allows the administrator to promote an existing regular user to administrator status. This function would also subsequently demote the previous administrator to a regular user.

3.4.2 Stimulus/Response Sequences

- The user clicks the "Sidebar" icon in the top-left corner of the screen.
- The user clicks the “Manage Users” button.
- The user is navigated to the “Manage Users” page.

- In the “Manage Users” page, the administrator clicks the “shield” icon on the row of the existing regular user to be promoted.
- Upon promotion, the previous administrator is redirected to the “Login” page and the new administrator receives admin privileges.

3.4.3 Functional Requirements

- PU1 - A verification function checks for the user's role (i.e., administrator or regular user) to access the “Manage Users” page.
- PU2 - The user will be navigated to the “Login” page upon a successful promotion of a regular user to an administrator.
- PU3 - The function that grants admin privileges to the regular user promoted and revokes admin privileges to the previous administrator should be properly implemented.

3.5 User Authentication

3.5.1 Description and Priority

This is a high-priority feature which provides a mechanism to prevent unauthorized users from accessing pages with potentially sensitive data. If a user is not logged in, the user will automatically be redirected back to the “Login” page.

3.5.2 Stimulus/Response Sequences

- The user enters their credentials (i.e., email and password).
- The user clicks the "Sign In" button.
- The program returns a prompt if the user has succeeded or failed to log in.
- Upon successful login, the user is taken to the “Home” page.
- Upon failure to login, the user is restricted to the “Login” page
- To log out, the user clicks the sidebar button shown at the top-right corner of the page and clicks the "Sign Out" button.
- Alternatively, the user can click the "Sign Out" button on the top-right corner of the page to log out.

3.5.3 Functional Requirements

- UA1 - Form validation that ensures all input fields are not empty must be implemented in the “Login” page.
- UA2 - A search function checks if the email address inputted exists in the database and when existing, verifies whether the entered credentials matches with the encrypted password in the database.
- UA3 - A dialog box appears to prompt the user of their success or failure to login.
- UA4 - Upon signing out, all cookies and tokens stored will be removed.

3.6 Edit User Profile

3.6.1 Description and Priority

This is a low-priority feature which provides a mechanism for authenticated users to edit their account profile. Here, users can modify their first and last names. The email address is also displayed but is not editable.

3.6.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” in the top-left corner of the screen.
- The user clicks the “user” icon or the email address hyperlink.
- The user is navigated to the “Edit Profile” page.
- In the “Edit Profile” page, the user clicks the “pen” icon.
- The user modifies their first name and/or last name.
- The user may then click the “Save” button to save their changes.
- The user may also choose to click the “Cancel” button instead to drop the changes made and be redirected to the “Home” page.

3.6.3 Functional Requirements

- EUP1 - Form validation against empty name fields must be fully functional.
- EUP2 - The page will reload to display the updated user profile upon a successful edit user profile operation.

3.7 Change Account Password

3.7.1 Description and Priority

This is a high priority feature. It provides authenticated users an avenue to modify the randomly generated password assigned to their respective accounts initially.

3.7.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” in the top-left corner of the screen.
- The user clicks the “user” icon or the email address hyperlink.
- The user is navigated to the “Edit Profile” page.
- The user clicks the “Change Password” button
- The user is navigated to the “Change Password” page.
- The user inputs their old password, new password and retyped new password.
- The user clicks the “Save” button to save changes
- The program returns a prompt if the user has succeeded or failed to change the password.
- The page reloads.

3.7.3 Functional Requirements

- CAP1 - Form validation that checks the following must be properly implemented:
  - All input fields must be filled.
  - The new and retyped password must match.
  - New password must not be the same as the old password.
  - New password must follow formatting conventions.
- CAP2 - A verification function checks whether the credentials entered match with the encrypted password in the database.
- CAP3 - A dialog box appears to prompt the user of their success or failure to change their password.

3.8 View List of Student Records

3.8.1 Description and Priority

This is a high-priority feature. This page displays all the student records uploaded in the database in table format. When a row in the table is clicked, the user is navigated to the “Edit Student Record” page of the student indicated in the row. This page also features a search function to filter out students based on their student number as well as a sorting function, which the user can use to sort the columns by Student Number, First Name, Last Name, and Course in ascending or descending order.

3.8.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” icon shown at the top of the screen.
- The user clicks the “Student Record” hyperlink.
- The user is taken to the “Student Records” page.
- The user types on the “Searchbar” to search for a specific student based on their student number.
- The user clicks the table headers to sort the student records based on these fields.
- The user clicks on the left and right arrows on the bottom right of the page to view more student records.

3.8.3 Functional Requirements

- VSR1 - The function to retrieve all student records must be functional.
- VSR2 - The sorting function implemented must work properly.
- VSR3 - The search function based on student number must function properly.
- VSR4 - The pagination of the table must function properly.

3.9 Add Student Records

3.9.1 Description and Priority

This is a high-priority feature. It provides authenticated users an interface to add student records to the database, which they can update and validate later on. This can be done either manually, by filling out a student record form, or automatically, by uploading .csv or .xlsx files. File uploads can be in single or in bulk.

3.9.2 Stimulus/Response Sequences

- The user clicks the "Sidebar" button shown at the top of the screen.
- The user clicks the "Student Record" hyperlink.
- The user is taken to the "Student Records" page.
- To add a new record manually:
  - The user clicks the "Add Record" button.
  - The user is taken to the "Add New Record" page, where the user can fill out the information required for the student record.
  - The user clicks the "Save" button.
  - The system validates the value and format of the information entered by the user.
  - If the validation is successful, the record is saved and the user will be redirected to the "Student Records" page.
  - If the validation fails, error messages will be displayed below the concerned input fields.
- To upload records in single or in bulk:
  - The user clicks the "Upload" button.
  - The user selects one or more .csv or .xlsx files that contain the student records.
  - The system shows the status of each file that was selected if the upload succeeded or failed.

3.9.3 Functional Requirements

- ASR1 - File upload in singles or in bulk must be fully operational.
- ASR2 - A warning message will be displayed when the file format is invalid. Only files with a .csv or .xlsx extension should be accepted.
- ASR3 - Form validation must be implemented for manual record addition.
- ASR4 - Parser for .csv and .xlsx files must be fully functional
- ASR5 - Parsed student record object normalization and formatting must work properly
- ASR6 - Success messages regarding the status of the file uploads will be displayed per filename in a dialog box.
- ASR7 - Failure messages regarding the status of the file uploads will be displayed per filename in a dialog box alongside the cause for rejection.

3.10 View Student Record

3.10.1 Description and Priority

This is a high-priority feature. It allows authenticated users to view the entirety of the student record in an organized form layout for easier cross-checking.

3.10.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” button shown at the top of the screen.
- The user clicks the “Student Record” hyperlink.
- The user is taken to the “Student Records” page.
- The user types on the “Searchbar” to search for a specific student based on their student number.
- The user clicks the row where the desired student is indicated.
- The user is taken to the “Edit Record” page where they can view the record in a filled out form where the user can edit the record of the student.
- The user clicks the “Scroll to Bottom” button to slide down to the bottom of the page.
- The user clicks the “Scroll to Top” button to slide back to the top of the page.
- The user is shown the “Signed By” field and the “Remarks” of previous users.

3.10.3 Functional Requirements

- VSR1 - Function for retrieval of student records from the database must be fully operational.
- VSR2 - Rendering of values must accurately reflect the student record.

3.11 Edit Student Record

3.11.1 Description and Priority

This is a high-priority feature. It allows authenticated users to revise the student record in the event of having inconsistencies and typographical errors.

3.11.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” button shown at the top of the screen.
- The user clicks the “Student Record” hyperlink.
- The user is taken to the “Student Records” page.
- The user clicks on a student record.
- The user modifies a field on the student record.
- The user clicks the “Scroll to Bottom” button to slide down to the bottom of the page.
- The user enters the changes committed on the "remarks" field

- The user clicks the "Update" button
- The screen reloads and shows the updated records.

3.11.3 Functional Requirements

- ESR1 - Function that checks if the form contents were modified must be fully operational.
- ESR2 - Security safeguard that requires the user to provide a corresponding remark to the modifications committed must be implemented.
- ESR3 - A message will appear below to prompt the success or failure of the update of student records.
- ESR 4 - Function to update the student records in the database must be implemented.
- ESR5 - The page will reload to display the updated student record upon a successful update user record operation.

3.12 Delete Student Record (Admin Privilege)

3.12.1 Description and Priority

This is a high-priority feature which allows an authenticated administrator to delete a student record once it is no longer needed in the system..

3.12.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” button shown at the top of the screen.
- The user clicks the “Student Record” hyperlink.
- The user is taken to the “Student Records” page.
- The user clicks on a student record.
- The user clicks the “Scroll to Bottom” button to slide down to the bottom of the page.
- The user clicks on the "Delete button"
- The user is redirected to the updated “Student Records” page.

3.12.3 Functional Requirements

- DSR1 - A verification function checks for the user's role (i.e., administrator or regular user) to access the “Delete” button.
- DSR2 - A message will appear below to prompt the success or failure of deletion of student records.
- DSR3 - Function to delete student records in the database must be implemented.

3.13 Delete All Student Records (Admin Privilege)

3.13.1 Description and Priority

This is a high-priority feature. It allows an authenticated administrator to clear all student records stored in the system.

3.13.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” button shown at the top of the screen.
- The user clicks the “Student Record” hyperlink.
- The user is taken to the “Student Records” page.
- The user clicks the “Delete All” button to clear all stored records.
- The user inputs their password in the prompt to confirm this action.
- The page reloads and the user is shown the updated “Student Records” page.

3.13.3 Functional Requirements

- DASR1 - A verification function checks for the user's role (i.e., administrator or regular user) to access the “Delete All” button.
- DASR2 - Security safeguard that requires the user's password for highly-sensitive operations must be in place.
- DASR3 - A verification function checks whether the credentials entered match with the encrypted password in the database.
- DASR4 - Function to clear all student records stored in the database must be implemented.
- DASR5 - A dialog box appears to prompt the user of the success or failure of the clearing operation.

3.14 Inconsistency Detection for Student Records

3.14.1 Description and Priority

This is a high-priority feature. It allows the user to view all the inconsistency warnings of a student record. This includes checks on the format of the record as well as validation of the computations in it. Furthermore, the record can be compared with its respective curriculum that provides messages on any discrepancies found.

3.14.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” icon shown at the top of the screen.
- The user clicks the “Student Records” hyperlink.
- The user is taken to the “Student Records” page.
- The user clicks on a student record.
- The user clicks the "Check Inputs" button
- The user is shown all the detected computational warnings
- The user is shown all the detected curriculum inconsistency warnings.

- The warnings are displayed in a table, below the record, that indicates the message, course, semester, and year the warnings are applicable.

3.14.3 Functional Requirements

- IDSR1 - A message that prompts a student's enrollment status when AWOL or LOA will be displayed.
- IDSR2 - A message that prompts underloaded semesters (units < 12) will be displayed.
- IDSR3 - Warning(s) about subjects with invalid non-numerical and out of range numerical grades will be displayed.
- IDSR4 - Warning(s) about subjects pending completion (INC, DFG, 4) will be displayed.
- IDSR5 - Warning(s) about incorrect grade point computation will be displayed.
- IDSR6 - Warning(s) about invalid course units will be displayed.
- IDSR7 - Warning(s) about incorrect tally for the cumulative sum of grade points will be displayed.
- IDSR8 - Warning about incorrect total number of units taken.
- IDSR9 - Warning about incorrect final GWA.
- IDSR10 - Display message(s) will appear and inform the user of lacking HK units.
- IDSR11 - Display message(s) will appear and inform the user of lacking NSTP units.
- IDSR12 - Display message(s) will appear and inform the user of lacking core GE units.
- IDSR13 - Display message(s) will appear and inform the user of lacking GE Elective units.
- IDSR14 - Display message(s) will appear and inform the user of the completion of Terminal course requirements (SP or Thesis).
- IDSR15 - Display message if student record did not complete all courses specified in the curriculum.
- IDSR16 - Display message if student record did not meet the number of specialization and major units specified in the curriculum.
- IDSR17 - Display message if student record did not meet the total number of units required in the curriculum.
- IDSR18 - Display message if student record did not exclude the 2nd semester of AY. 2019-2020 from GWA computation.

3.15 Sign Student Record

3.15.1 Description and Priority

This is a medium-priority feature. This provides the user an option to "sign" the record to denote that it has been checked by a SHAC personnel. The summarized view of inconsistency warnings and other relevant details are provided here.

3.15.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” icon shown at the top of the screen.
- The user clicks the “Student Record” hyperlink.
- The user is taken to the “Student Records” page.
- The user clicks on a student record.
- The user clicks the "Check Inputs" Button and cross-checks the warnings shown.
- To sign as the first user
    - The user clicks the "Sign" Button
- To sign as the second user
    - The user clicks the checkbox besides the "I agree" message
    - The user clicks the "Sign" Button

3.15.3 Functional Requirements

- SSR1 - The checker which ensures that a student record has been checked for any inconsistency before allowing a sign operation must be working
- SSR2 - Function to update the student records in the database must be implemented.
- SSR3 - A verification function checks for the user's role (i.e., administrator or regular user) to access the “Unsign” button. A regular user can only remove their signature, while an administrator can clear all signatures.

3.16 Summary

3.16.1 Description and Priority

This is a high-priority feature. This feature allows authorized users to view all the verified student records. A verified student record is a record that has been signed by exactly two (2) authenticated users in the system. The student records displayed in the page can be sorted by student number, last name, degree program, and general weighted average (GWA).

3.16.2 Stimulus/Response Sequences

- The user clicks the “Sidebar” button shown at the top of the screen.
- The user clicks the "Summary" hyperlink.
- The user is taken to the "Summary" page that contains a table of verified student records..
- The user can click on a column header (student number, full name, degree program, and GWA) to sort the records in ascending or descending order.
- The user can click on the previous/next button at the bottom of the table to view the records not displayed in the current page of the table.

- The user can click on a student record in the table, and the user will be redirected to the "View Individual Student Record" page.

3.16.3 Functional Requirements

- S1: Record Verification - The "Summary" page only shows the student records that have been signed by two users.

3.17 Logs

3.17.1 Description and Priority

This is a low-priority feature which provides a page for viewing the activities of all users across the software. The information in the logs include the following: email, record accessed, action done, and the time of access.

3.17.2 Stimulus/Response Sequences

- The user clicks the "Sidebar" button shown at the top of the screen.
- The user clicks the "Logs" hyperlink.
- The user is navigated to the "Logs" page.
- If the user has administrator privileges, a "Delete All Logs" button is available.
- The administrator user clicks the "Delete All Logs" button to delete all logs.

3.17.3 Functional Requirements

- L1 - The logs will be shown in reverse chronological order based on the time of access.
- L2 - Logging the user activity takes place when any action in the student records is done. Additionally, the delete logs action is also logged.
- L3 - The function that checks for the user's privilege level must be working for the Delete Logs button.

3.18 About

3.18.1 Description and Priority

This is a low-priority feature designed to show the profiles of all the individuals involved in the development of the project.

3.18.2 Stimulus/Response Sequences

- The user clicks the "Sidebar" icon shown at the top of the screen.
- The user clicks the "About us" hyperlink.
- The user is navigated to the "About" page.
- The user is presented with an overview of the software specifications and a list of the people involved in the creation of the software.

3.18.3 Functional Requirements

- A1 - Retrieval function for the images of the developers must be implemented.

4. External Interface Requirements

4.1 User Interfaces

The user interface of the system was built using React and MUI, which are JavaScript libraries for creating interactive and intuitive user interfaces for the web. It follows the Material Design guidelines created by Google through the use of the components provided by the Material UI library of MUI.

For best compatibility, the users of the system can use any modern web browser from 2022, such as Microsoft Edge, Google Chrome, Mozilla Firefox, and Apple Safari to access the user interface of the system. It is also recommended that the web application is viewed on a monitor with a resolution of at least 1920x1080 pixels (Full HD).

For a visual representation of the user interfaces, see Appendix E.

Component	**Description**
Landing Page	Interface upon software startup
Login Page	Contains a form where users use to login
Create Admin User Page	Contains a form where the admin user can create an account
Toolbar	Contains the Sidebar, Back button, Quick Access buttons and the Sign Out button
Sidebar	Navigation component that contains the hyperlinks to the main pages of the web app.
Home Page	Displays recent changes, total records, and total verified records
View List of Student Records Page	Displays the list of student records in table format.
Edit Student Records Page	Displays the details of a student record where the user can update, check, and sign.

Add New Record Page	Contains a form where the user can manually input and save a new student record.
Upload Student Records Page	Contains a button that, when pressed, opens a file selector dialog where the user can upload a single or multiple student records.
Manage Users Page	Displays all the users in the system, which the administrator can manage.
Add User Page	Provides an interface to the admin where regular user accounts can be created
Summary Page	Displays all the verified student records. Verified student records are student records signed by two users
Logs Page	Displays the actions of all users done in student records and logs
Edit Profile Page	Displays user's information and where they can edit their information
Change Password Page	Provides an interface where the user can change their current password
About Page	Displays the personal information of the developers
Error Message Display Standards	Displays the error title and the error description.

4.2 Hardware Interfaces

The back-end and database components will be deployed in the same server computer. Client computers can access the system by using a standard web browser and connect via intranet. The communication between server and client will be using the JSON format.
This application has been tested in the devices with the following device specifications:

Type of Device	*Specifications*
Laptop	Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz (16.0 GB RAM) 64-bit operating system, x64-based processor

Phone	Octa-core (4x2.0 GHz Cortex-A76 & 4x2.0 GHz Cortex-A55) (128GB) 8GB RAM
Tablet	System-on-chip Apple A7 with 64-bit architecture. Coprocessor Apple M7. System-on-chip CPU 1.3 GHz dual-core Apple Cyclone. (1GB)

4.3 Software Interfaces

The database management system to be used is MongoDB, an open-source, NoSQL database. Mongoose, an Object Data Modeling (ODM) module, will be used to interact with the database.

4.4 Communications Interfaces

The software will use HTTP to send requests and receive responses with the backend server.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

In order to account for the occasional power interruptions within the UPLB campus, the system must not be dependent on Internet connectivity. However, the validations done by the SHAC committee must be done on-premise among multiple authorized personnel. In line with this, the system must be accessible across multiple machines over an intranet. With this, the client hosts must be connected to the same local network where the server computer is connected to. This way, the clients can access the web application using the server's private Internet Protocol (IP) address.

Since the application is to be available to multiple users at the same time, based on the specifications agreed upon during the client meeting, concurrent operations on the same set of data are considered out of scope.

5.2 Safety Requirements

There are currently no safety requirements.

5.3 Security Requirements

The security of the system is implemented through client authentication. Users' passwords are hashed and salted using bcrypt before storing them in the database to prevent them from being in plaintext format.

Sensitive information, particularly the secret key for token authentication, is stored in a *.env* file as an environment variable. This enables the team to share code and collaborate over the Internet without exposing sensitive data. However, to make it simple and convenient for the laboratory instructor to test the project, the team uploaded the original *.env* file.

The server performs these two major checks sequentially:

1. The server checks for the IP address of the requesting clients. If a specific IP address has been detected as an outsider, the server blocks the request.

2. If a request was not blocked from the first check, the server then checks the token if it is a valid token. If it is, the transaction goes through. Otherwise, it is rejected. This step is not performed for accessing the signup and login page since this is the user interface for creating the first account and for logging in.

The server must only be accessible by authorized personnel. The device must be secured to prevent any form of physical tampering which may lead to any unwanted access to potentially sensitive data.

5.4 Software Quality Attributes

This software exhibits intuitiveness through its interface design to ensure that the user will be able to navigate through the pages with minimal supervision and instruction. Robustness is also one of the system's attributes with its rigorous input sanitation and error handling. The system is also reliable as it consistently performs according to its specifications where inputs are mapped to its corresponding output. It will also feature a reliable authentication system as it contains sensitive information, such as the transcript of records of students.

Appendix A: Glossary

Table 1. Acronyms and their definitions.

Acronym	Definition
AWOL	Absent Without Leave
CAS	College of Arts and Sciences
GE	General Elective
GWA	General Weighted Average
IP	Internet Protocol
LOA	Leave of Absence
NPM	Node Package Manager
OCS	Office of the College Secretary
SHAC	Scholarships, Honors, and Awards Committee
UPLB	University of the Philippines - Los Baños

Table 2. Terms and their definitions.

Term	Definition
SHAC Personnel	A member of the SHAC Committee of the College of Arts and Science in the University of the Philippines - Los Baños

Appendix B: Analysis Model

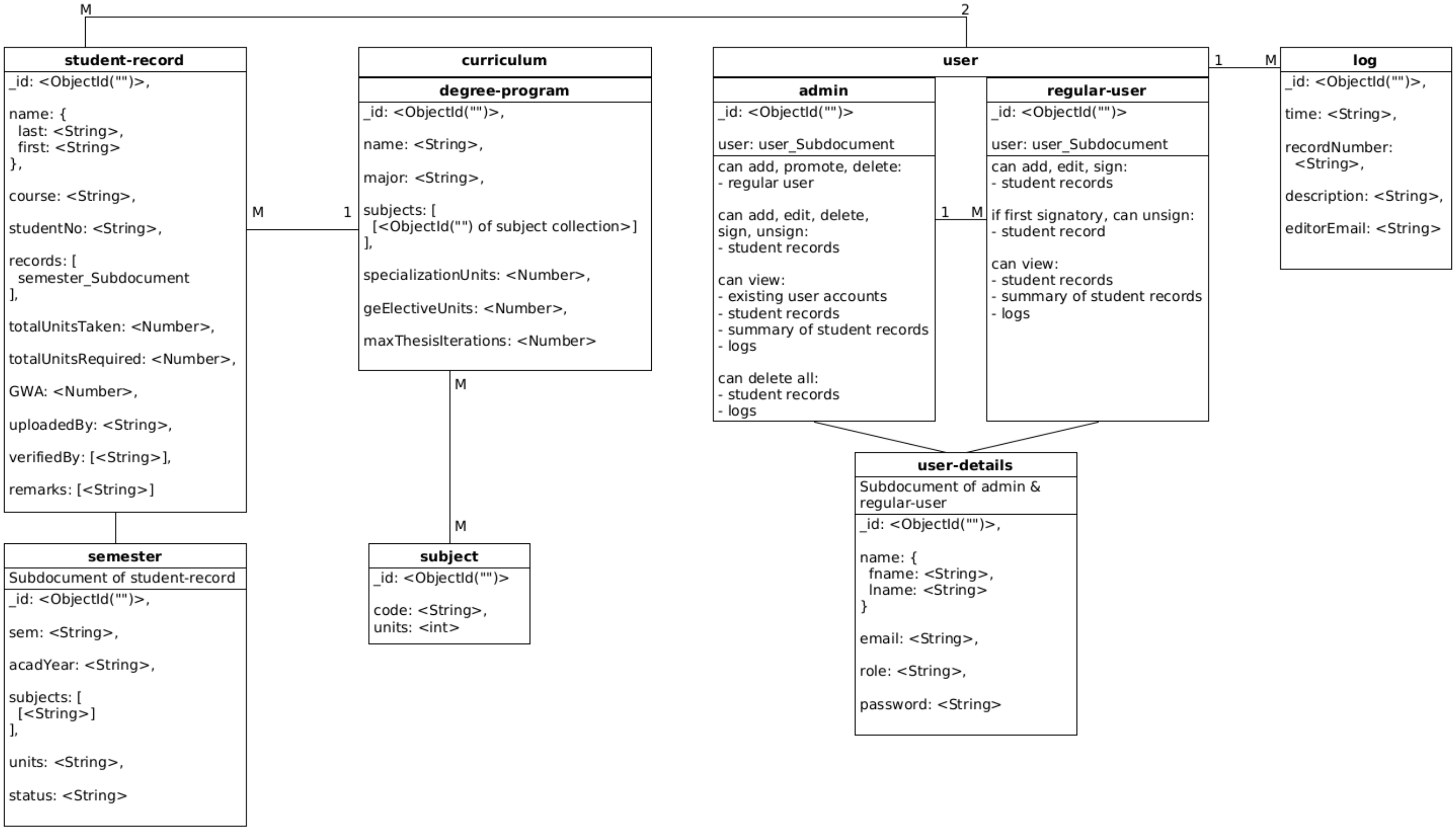


Figure 1. Analysis Model of SHACker.

There are five collections in the database: student-record, curriculum, subject, user, and log. A student record has a last and first name of the student, a degree program (i.e., course in Figure 1), a unique student number, an array of semestral records, total units taken, total units required, and GWA. It also includes the email address of the user that uploaded the student record (i.e., uploadedBy) , the email addresses of users who signed the student record (i.e., verifiedBy), and annotations of the users who plan to edit the student record (i.e., remarks). Under each semester record is its semester type (i.e., sem; 1, 2, M), academic year, an array of subjects, units taken for the semester, and enrollment status (i.e., Enrolled, LOA, AWOL). The subjects are of type String which contains the subject code, units, grade, gradepoint, and summation relative to the student record.

A degree program has a shortened degree name, a major (limited to SP, Thesis, and MST majors), an array of subjects, specialization units, GE elective units, and maximum thesis iterations. A subject has a subject code and units. Curricula stored in the database are compiled in Appendix F.

The array of subjects is a list of subjects explicitly stated in the curriculum that need to be taken by the student. GE elective units is the number of units that a student needs to take for GE electives (See Appendix F, Table 6.) To compute for the specialization units, the number of units from the subject array and the GE electives is subtracted from the total number of units offered in the curriculum like so:

$$specializationUnits = totalUnits - (unitsOfListedSubjects + GEElectiveUnits)$$

A user has a last and first name, email, role, and password. A user can be an administrator or a regular user. A regular user can add, edit, and sign student records; unsign a student record if they are the first signatory; and view all student records, a summary of verified student records, and all logs. An administrator has the same actions of a regular user but with higher privileges. Aside from the earlier stated actions, the administrator can add, promote, and delete regular users; can unsign and delete any student records; can view all existing user accounts; and can delete all student records and logs.

A log has a date and time created, a record number which refers to the student number of the student record, a description which refers to the action done on the student record, and an editor email which refers to the email of the user that acted on the student record.

Appendix C: Use Case Diagram

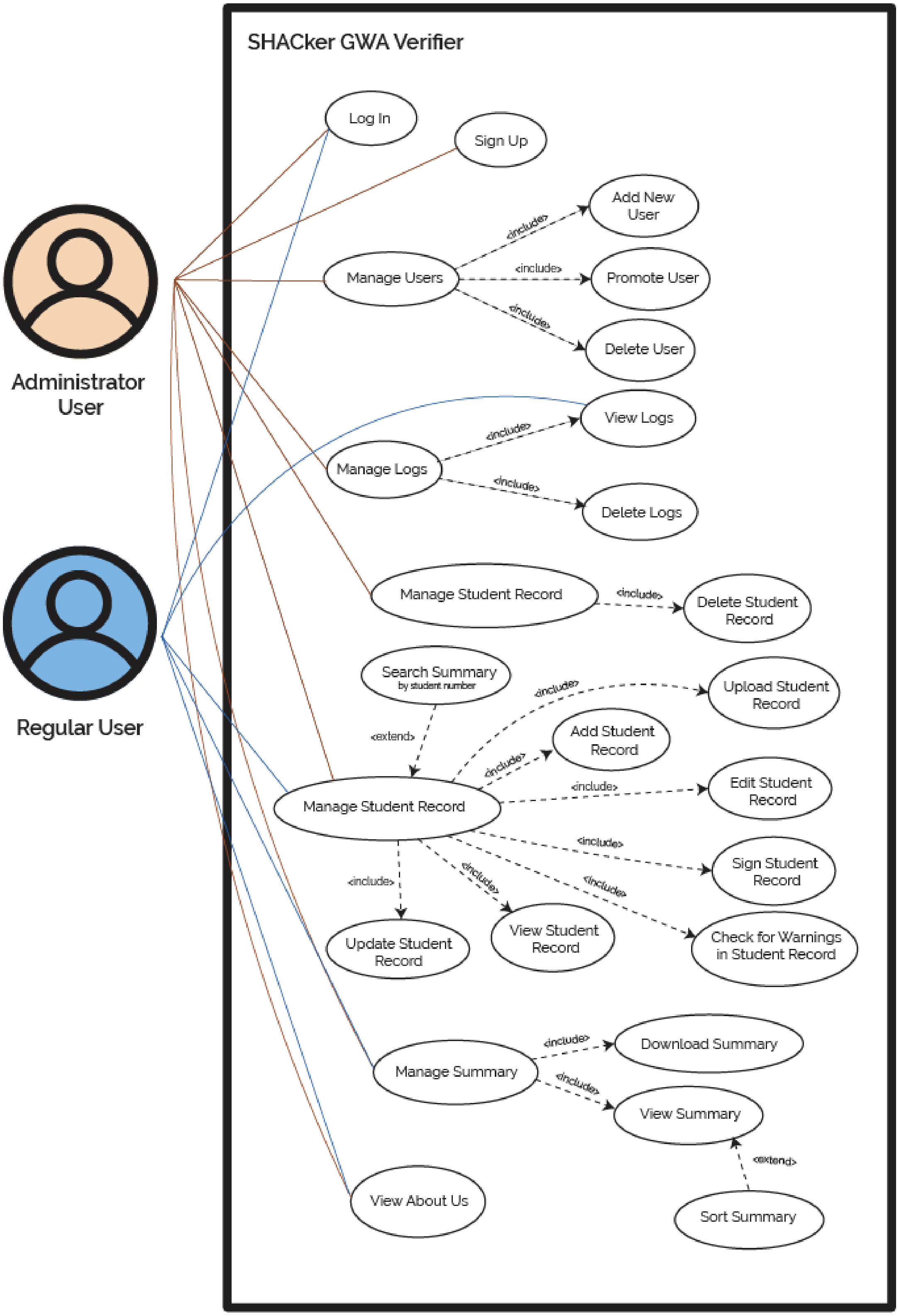


Figure 2. Use Case Diagram of SHACker

Appendix D: Sample Student Record

Below is a sample format of the student record which prescribes the fields found in a valid input file in .csv or .xlsx.

Table 3. Student Record Format.

lname	**fname**					
course						
studNo						
Header						
course code	**grade**	**unit**	**grade point**	**running sum**	**total sem units**	**sem and year**
. . .						
	total units taken			**total running sum**		
GWA	**GWA**					
total units required						

The following is a partially filled student record with values assigned to the fields:

Table 4. Partially Filled Student Record.

Mahusto	Jenelle					
BSCS						
2019-12345						
CRSE NO.	Grade	Units	Enrolled	Term		
CMSC 12	1	3	3	3		
CMSC 56	1.25	3	3.75	6.75	3	I/19/20
	6			6.75		
GWA	1.125					
130						

Appendix E: Screenshots of User Interfaces

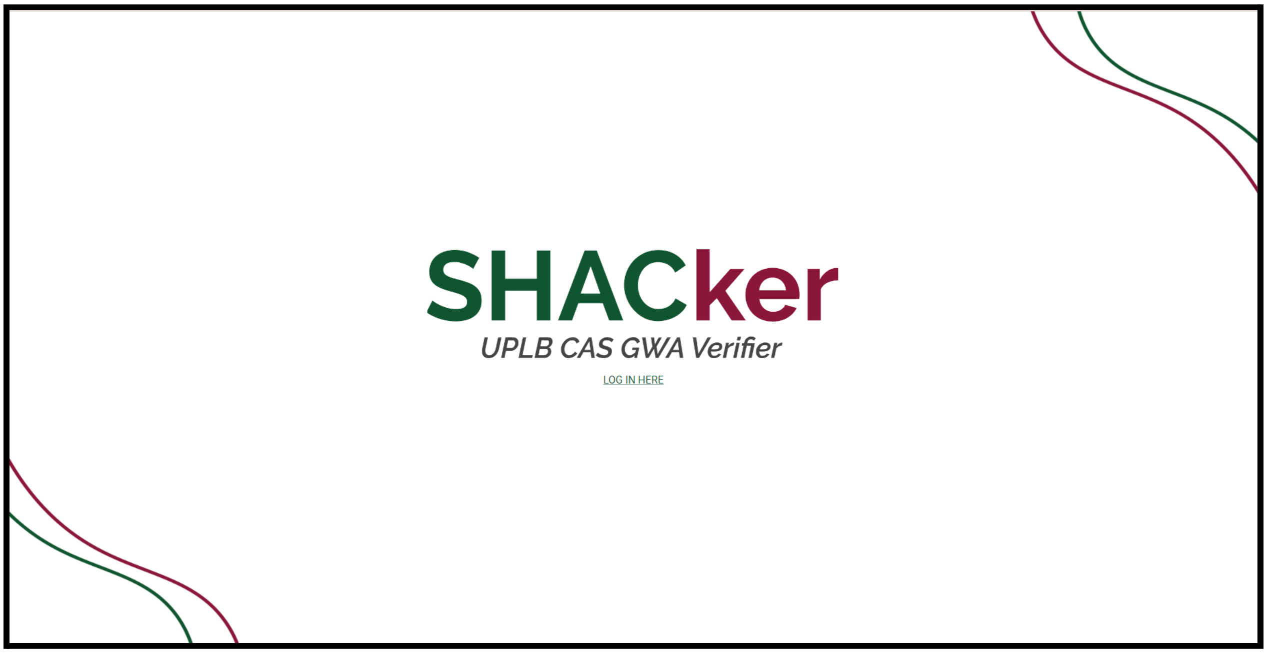


Figure 3. Landing Page (With Admin)

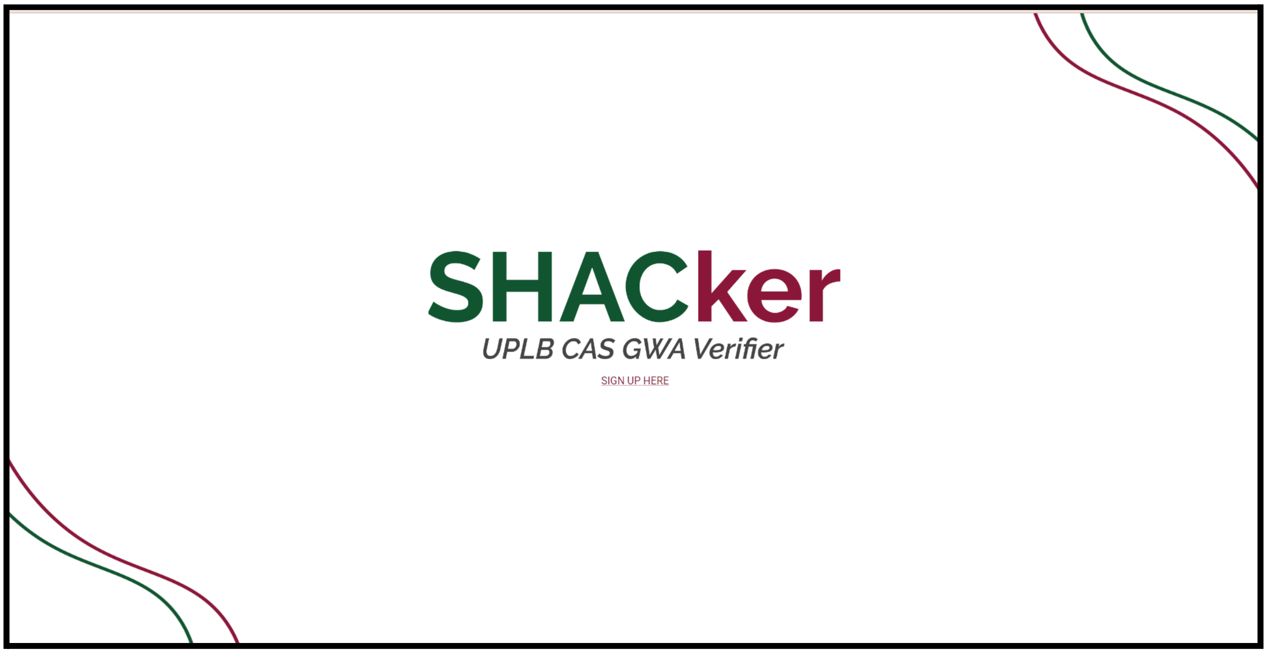


Figure 4. Landing Page (No Admin)

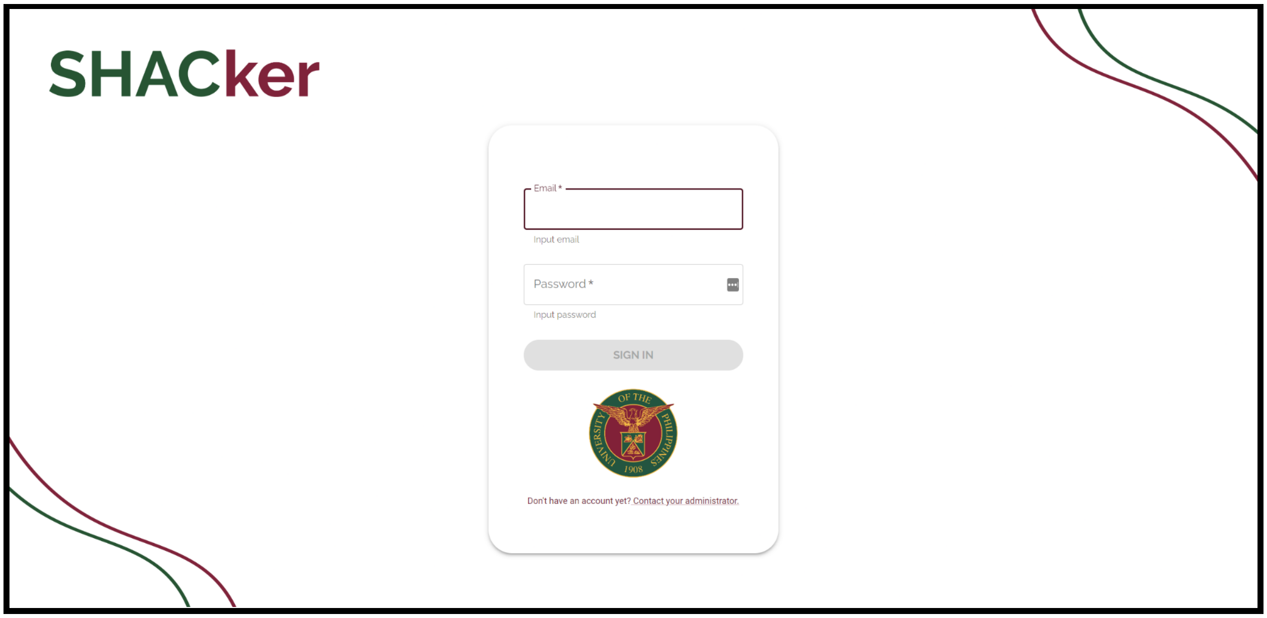


Figure 5. Login Page

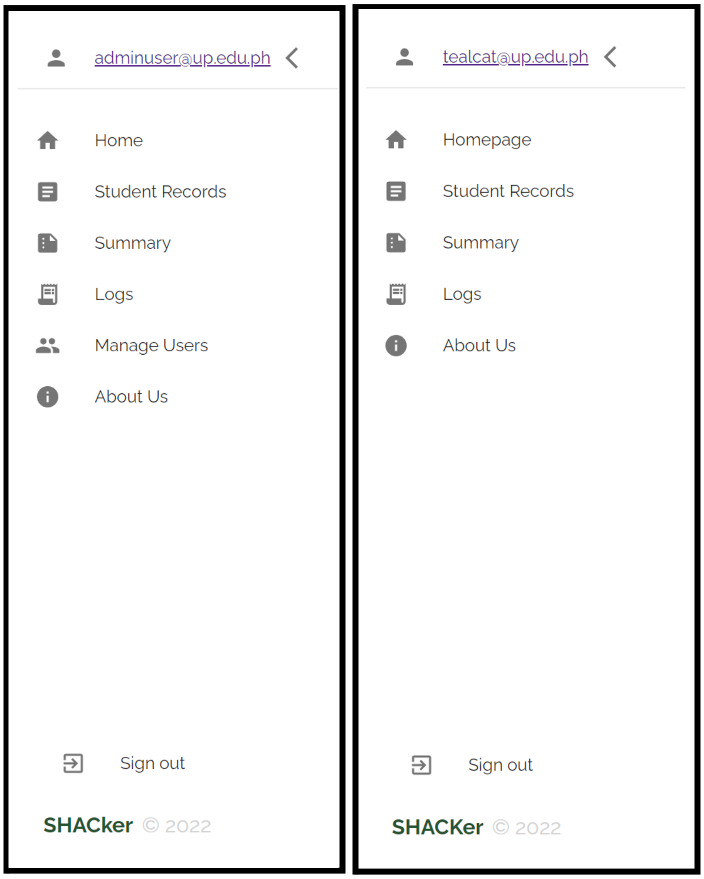


Figure 6. Admin Sidebar (Left) and Regular Sidebar (Right)

Appendix E: Screenshot of User Interfaces

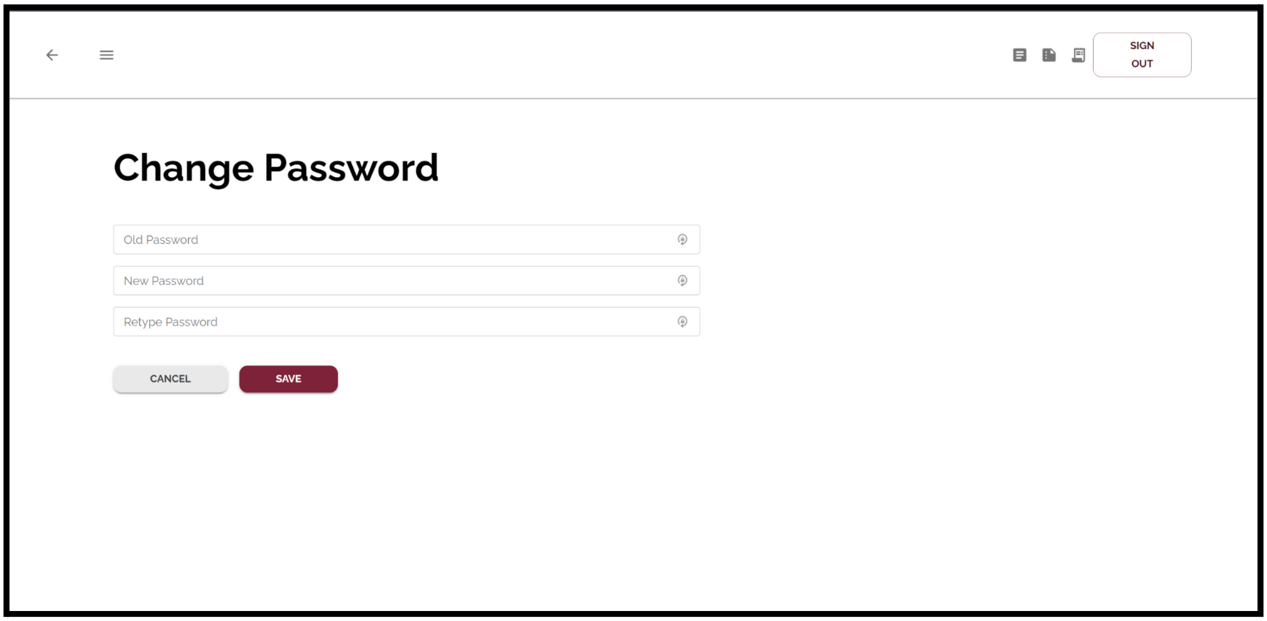
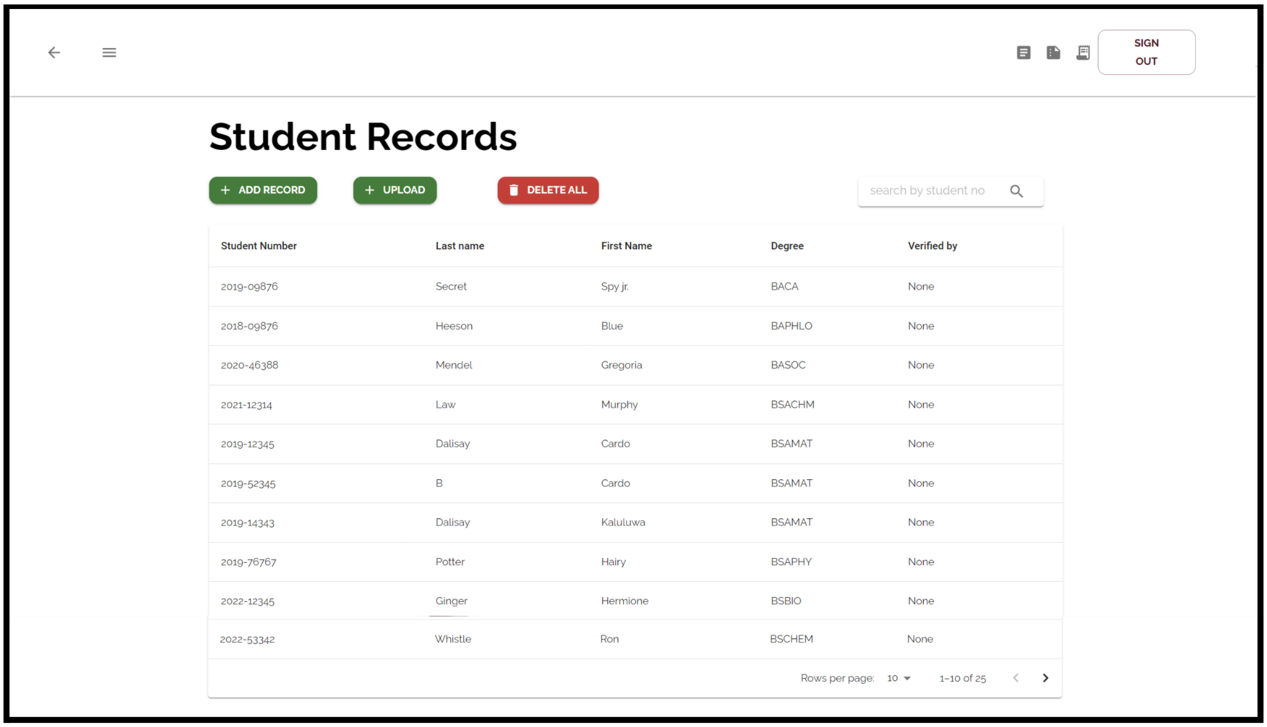


Figure 7. Change Password Page



Student Number	Last name	First Name	Degree	Verified by
2019-09876	Secret	Spy jr.	BACA	None
2018-09876	Heeson	Blue	BAPHLO	None
2020-46388	Mendel	Gregoria	BASOC	None
2021-12314	Law	Murphy	BSACHM	None
2019-12345	Dalisay	Cardo	BSAMAT	None
2019-52345	B	Cardo	BSAMAT	None
2019-14343	Dalisay	Kaluluwa	BSAMAT	None
2019-76767	Potter	Hairy	BSAPHY	None
2022-12345	Ginger	Hermione	BSBIO	None
2022-53342	Whistle	Ron	BSCHEM	None

Figure 8. View Student Records Page

Appendix E: Screenshot of User Interfaces

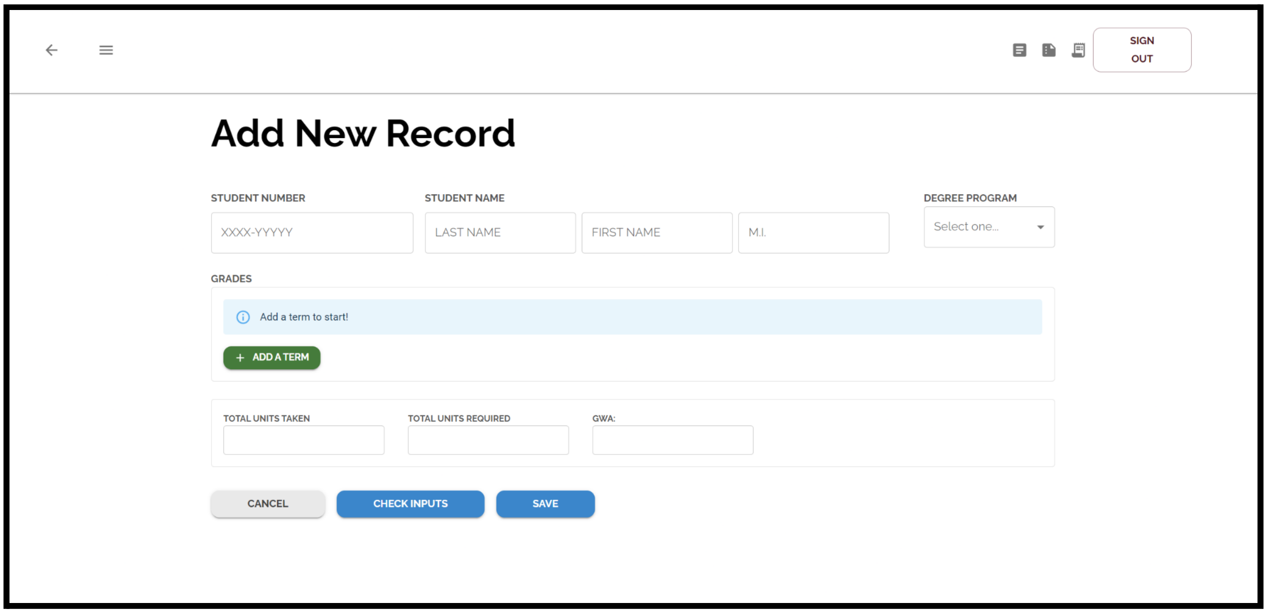
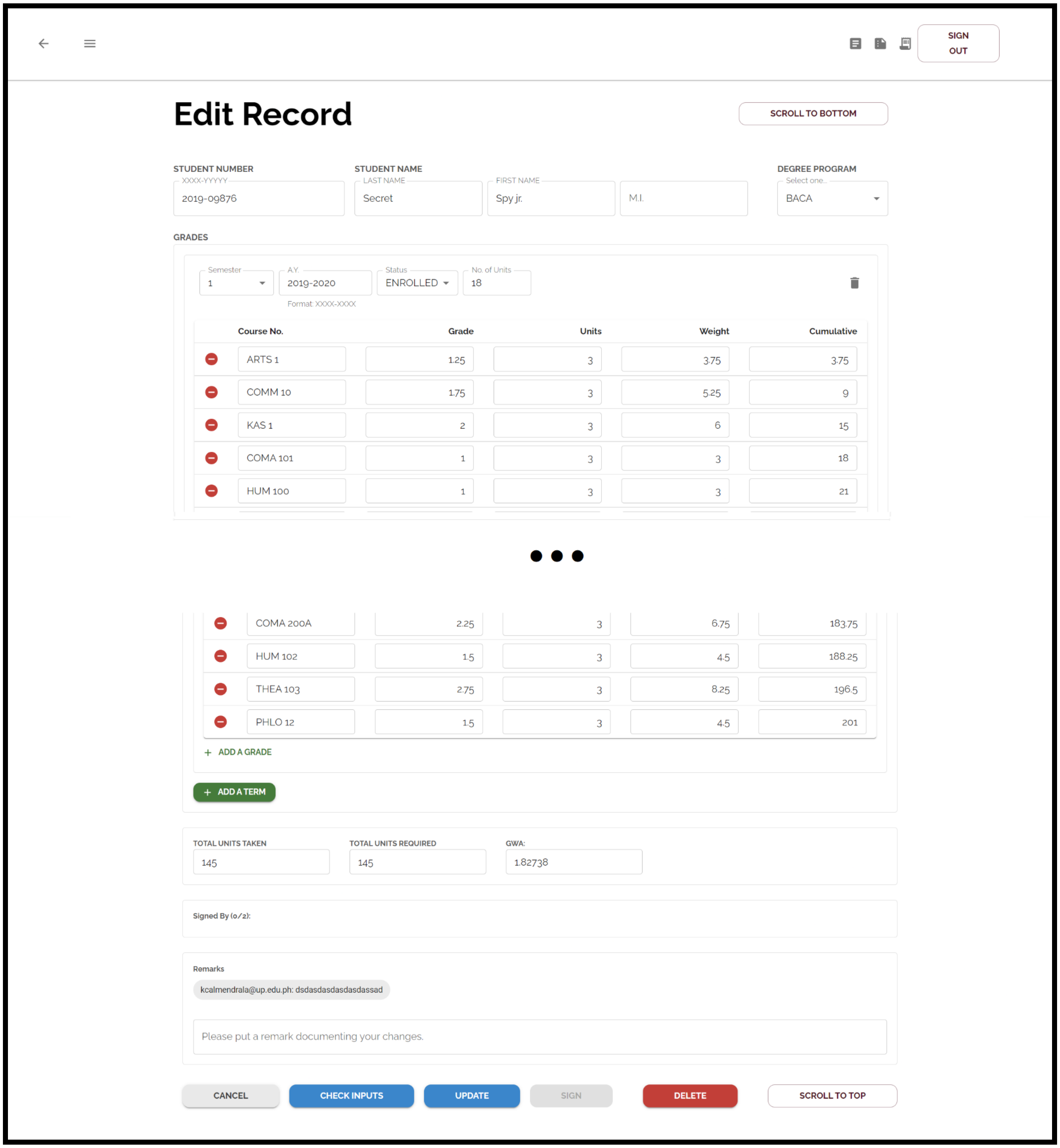


Figure 9. Add New Record Page



Edit Record

STUDENT NUMBER: XXXX-YYYYY 2019-09876

STUDENT NAME: LAST NAME Secret; FIRST NAME Spy jr.; M.I.

DEGREE PROGRAM: Select one... BACA

GRADES

Semester: 1 | A.Y.: 2019-2020 (Format: XXXX-XXXX) | Status: ENROLLED | No. of Units: 18

Course No.	Grade	Units	Weight	Cumulative
ARTS 1	1.25	3	3.75	3.75
COMM 10	1.75	3	5.25	9
KAS 1	2	3	6	15
COMA 101	1	3	3	18
HUM 100	1	3	3	21

● ● ●

COMA 200A	2.25	3	6.75	183.75
HUM 102	1.5	3	4.5	188.25
THEA 103	2.75	3	8.25	196.5
PHLO 12	1.5	3	4.5	201

TOTAL UNITS TAKEN: 145

TOTAL UNITS REQUIRED: 145

GWA: 1.82738

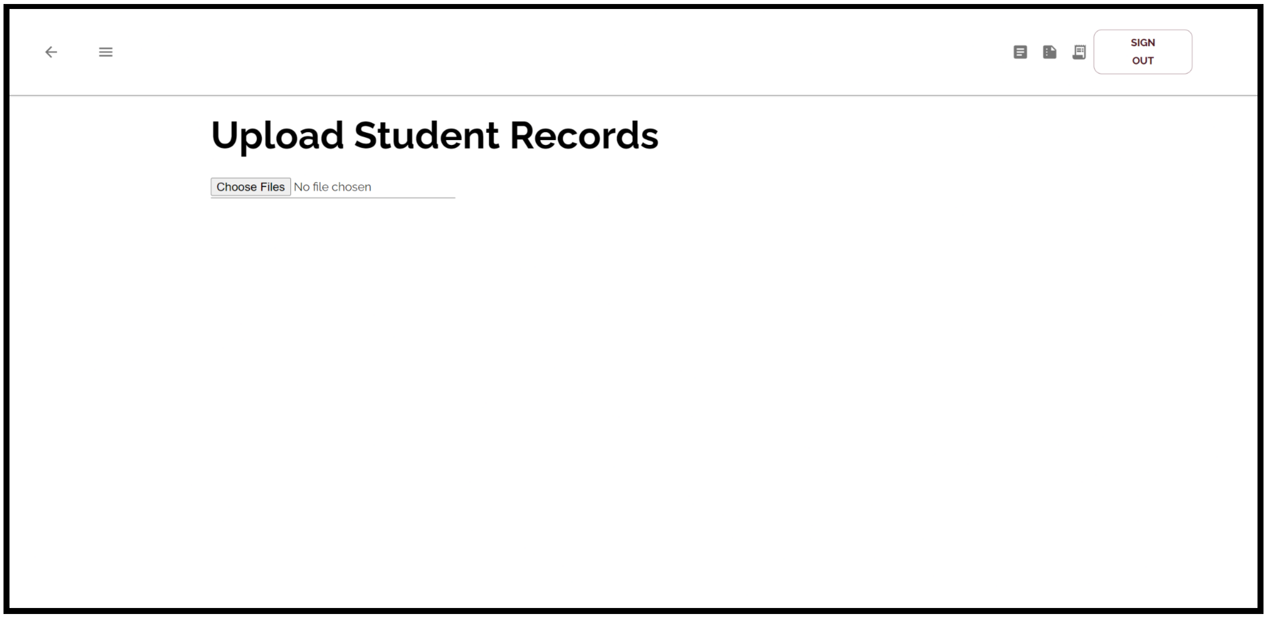
Signed By (0/2):

Remarks

kcalmendrala@up.edu.ph: dsdasdasdasdasdassad

Please put a remark documenting your changes.

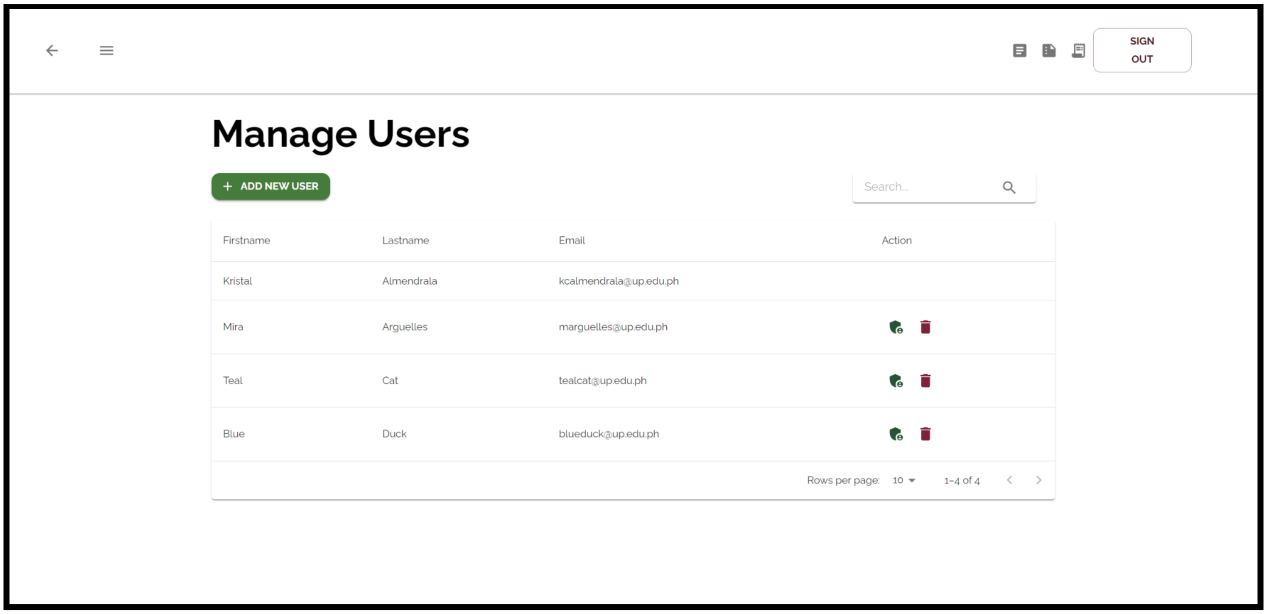
Figure 10. Edit Student Records Page



Upload Student Records

No file chosen

Figure 11. Upload Student Records Page

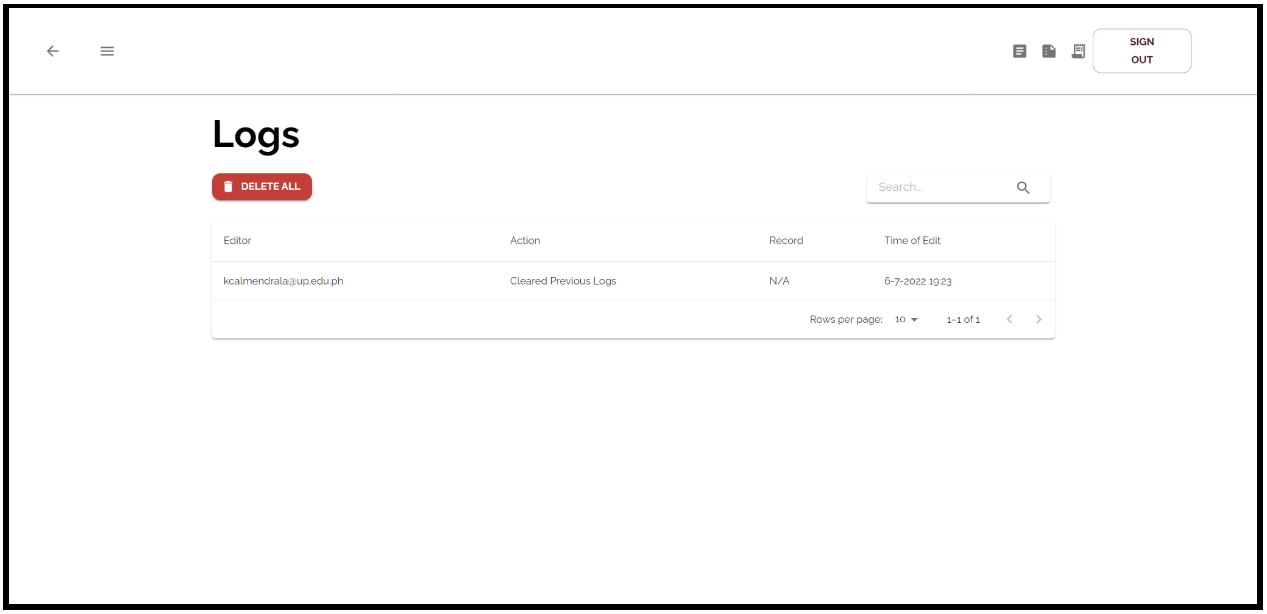


Manage Users

Firstname	Lastname	Email	Action
Kristal	Almendrala	kcalmendrala@up.edu.ph	
Mira	Arguelles	marguelles@up.edu.ph	🛡 🗑
Teal	Cat	tealcat@up.edu.ph	🛡 🗑
Blue	Duck	blueduck@up.edu.ph	🛡 🗑

Rows per page: 10 ▾ 1–4 of 4 ‹ ›

Figure 12. Manage Users Records Page

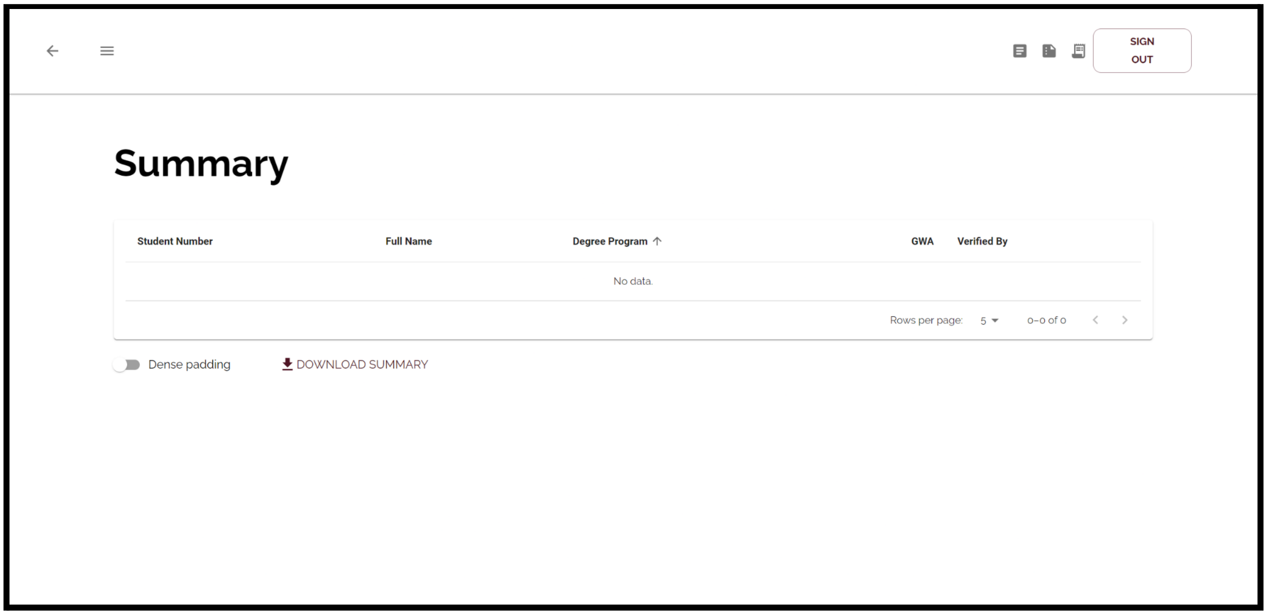


Logs

Editor	Action	Record	Time of Edit
kcalmendrala@up.edu.ph	Cleared Previous Logs	N/A	6-7-2022 19:23

Rows per page: 10 1–1 of 1

Figure 13. Logs Page

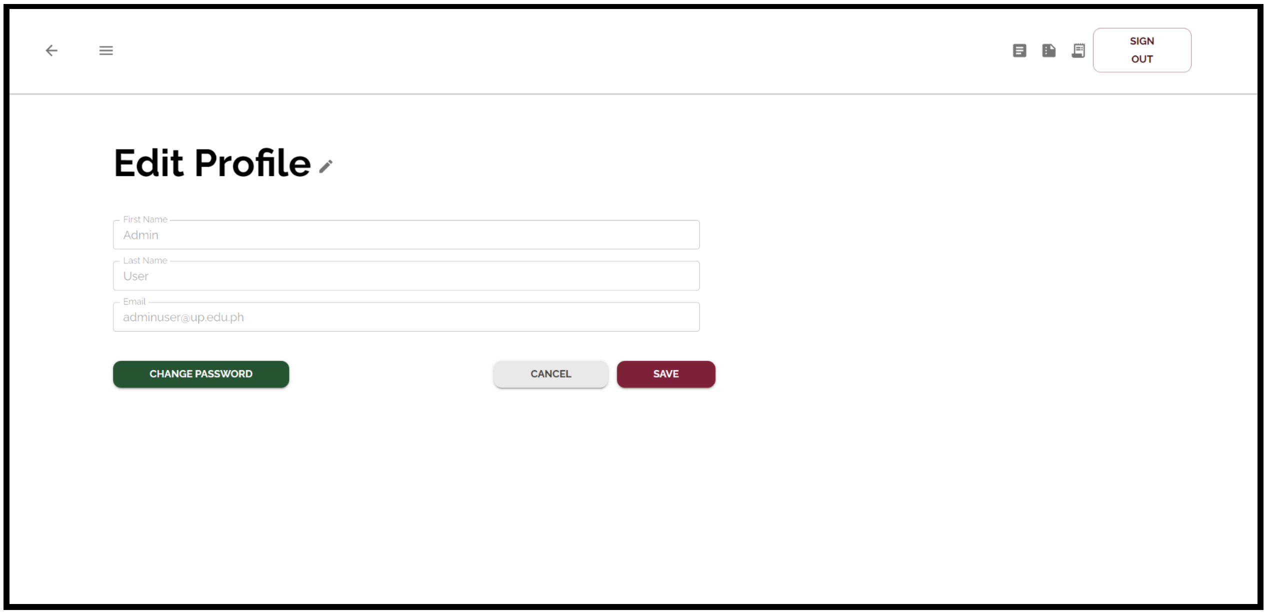


Summary

Student Number	Full Name	Degree Program ↑	GWA	Verified By
		No data.		

Rows per page: 5 0–0 of 0

Figure 14. Summary Page



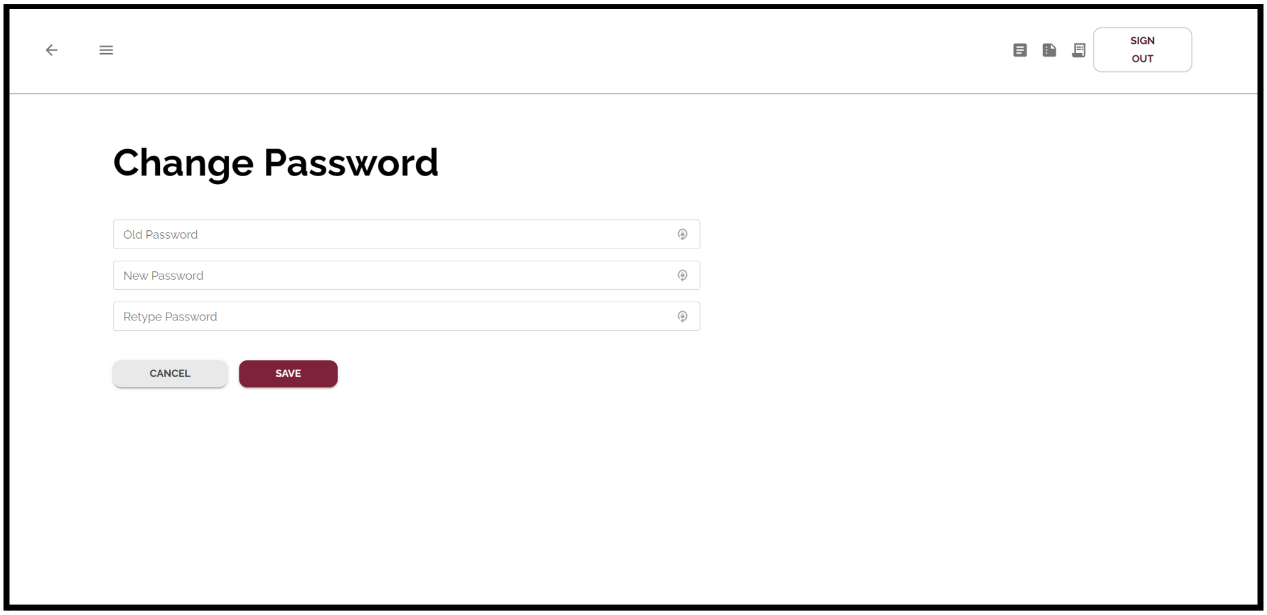
Edit Profile

First Name: Admin

Last Name: User

Email: adminuser@up.edu.ph

Figure 15. Edit Profile Page



Change Password

Old Password

New Password

Retype Password

Figure 16. Change Password Page

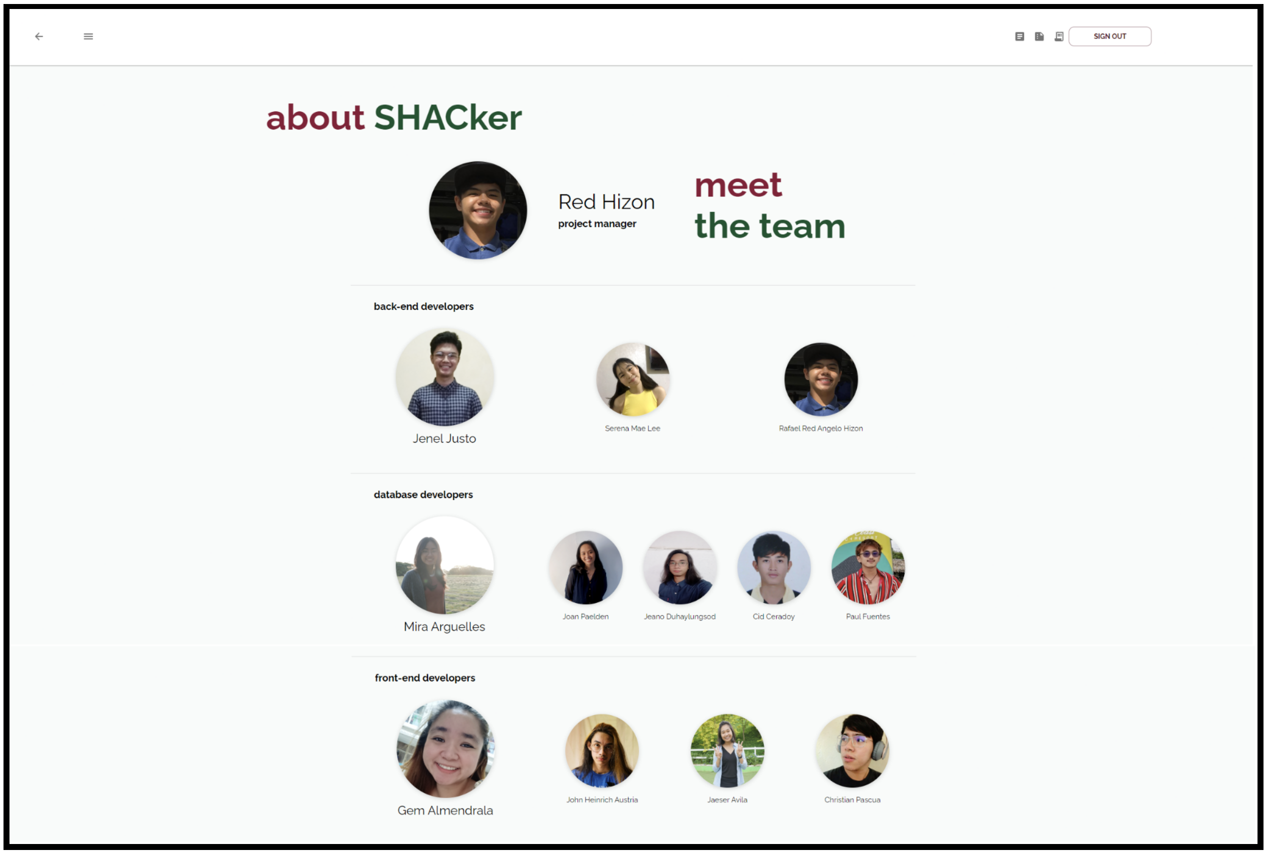


Figure 17. About Us Page

Appendix F: Available Curriculum Data

Table 5. Required GE Courses

Course Code	Credited Units
ARTS 1	3
COMM 10	3
ETHICS 1	3
KAS 1 / HIST 1	3
STS 1	3
PI 10	3

Table 6. Elective GE Courses

Course Code	Credited Units
HUM 3	3
KAS 4	3
MATH 10	3
PHILARTS 1	3
PHLO 1	3
PS 21	3
SAS 1	3
SCIENCE 10	3
SCIENCE 11	3
SOSC 3	3
WIKA 1	3

Bachelor of Arts in Communication Arts

Degree program code	BACA
Specialization units	39
GE elective units	9

Table 7. Required subjects explicitly listed in 2018 BACA curriculum

Course Code	Credited Units
ARTS 1	3
COMM 10	3
KAS 1/HIST 1	3
COMA 101	3
HUM 100	3
SPCM 102	3
HK 11	0
ENG 101	3
ENG 104	3
HUM 101	3
THEA 101	3
THEA 102	3
HK 12/HK 13	0
NSTP 1	0
COMA 190	3
COMA 105	3
COMA 200	6
COMA 200A	3
ETHICS 1	3
STS 1	3
COMA 102	3
ENG 100	3
SPCM 101	3
COMA 103	3
COMA 192	3
SPCM 104	3
NSTP 2	0
COMA 150	3
COMA 193	3
COMA 199	0
PI 10	3
HUM 104	3
HUM 170	3
HUM 102	3
THEA 103	3

Bachelor of Arts in Philosophy

Degree program code	BAPHLO
Major/Option	SP
Specialization units	33
GE elective units	9

Table 8. Required subjects explicitly listed in 2018 BAPHLO (SP) curriculum

Course Code	Credited Units
PHLO 11	3
PHLO 12	3
ARTS 1	3
COMM 10	3
ETHICS 1	3
KAS 1/HIST 1	3
NSTP 1	0
HK 11	0
PHLO 150	3
PHLO 171	3
SPEC	3
ECON 11/POSC 10	3
STS 1	3
PI 10	3
NSTP 2	0
HK 12/HK13	0
PHLO 110	3
PHLO 173	3
PHLO 111	3
PHLO 120	3
PHLO 174	3
PHLO 181	3
PHLO 195	3
PHLO 197	3
PHLO 113	3
PHLO 182	3
PHLO 176	3
PHLO 178	3
PHLO 190	6
PHLO 160	3
PHLO 184	3
PHLO 185	3
PHLO 112	3

Bachelor of Arts in Philosophy

Degree program code	BAPHLO
Major/Option	Thesis
Specialization units	33
GE elective units	9

Table 9. Required subjects explicitly listed in 2018 BAPHLO (Thesis) curriculum

Course Code	Credited Units
PHLO 11	3
PHLO 12	3
ARTS 1	3
COMM 10	3
ETHICS 1	3
KAS 1/HIST 1	3
NSTP 1	0
HK 11	0
PHLO 150	3
PHLO 171	3
SPEC	3
ECON 11/POSC 10	3
STS 1	3
PI 10	3
NSTP 2	0
HK 12/HK13	0
PHLO 110	3
PHLO 173	3
PHLO 111	3
PHLO 120	3
PHLO 174	3
PHLO 181	3
PHLO 195	3
PHLO 197	3
PHLO 113	3
PHLO 182	3
PHLO 176	3
PHLO 178	3
PHLO 200	6
PHLO 160	3
PHLO 184	3
PHLO 185	3
PHLO 112	3

Bachelor of Science in Agricultural Chemistry

Degree program code	BSACHM
Major/Option	Thesis
Specialization units	15
GE elective units	9

Table 10. Required subjects explicitly listed in 2018 BSACHM curriculum

Course Code	Credited Units	Course Code	Credited Units	Course Code	Credited Units
CHEM 18	3	PHYS 72	4	CHEM 43.1	2
CHEM 18.1	2	PHYS 72.1	2	CHEM 192	3
MCB 11	3	AGRI 21	3	AGRI 42	3
MATH 27	3	AGRI 32	3	CHEM 112.1	2
ETHICS 1	3	ECON 11	3	CHEM 115	3
KAS 1/HIST 1	3	NSTP 2	0	CHEM 137.1	2
HK 11	0	CHEM 44	3	CHEM 161.1	2
CHEM 19	3	CHEM 44.1	2	COMM 10	3
MATH 28	3	STAT 162	3	CHEM 198	3
BIO 30	3	AGRI 22	3	CHEM 133	4
AGRI 31	3	AAE 111/AAE 120	3	CHEM 120	3
ARTS 1	3	AGRI 61	3	CHEM 199	1
HK 12/HK 13	0	CHEM 111	3	AGRI 171	3
CHEM 32	3	CHEM 131	4	ACHM 200	6
CHEM 32.1	2	CHEM 140	4	STS 1	3
PHYS 71	4	CHEM 161A	3	CHEM 180	3
PHYS 71.1	1	AGRI 51	3	CHEM 185	3
BOT 20	3	AGRI 41	3	AGRI 199	1
CMSC 12/AMAT 152	3	CHEM 111.1	2	CHEM 137	3
PI 10	3	CHEM 112	3	CHEM 161B	3
NSTP 1	0	CHEM 43	3		

Bachelor of Science in Applied Mathematics

Degree program code	BSAMAT
Major/Option	SP
Specialization units	27
GE elective units	9

Table 11. Required subjects explicitly listed in 2018 BSAMAT (SP) curriculum

Course Code	Credited Units
AMAT 19	3
MATH 36	5
BIO 11.1	2
KAS 1/HIST 1	3
ETHICS 1	3
HK 11	0
MATH 37	3
STAT 101	3
CHEM 18	3
CHEM 18.1	2
ARTS 1	3
HK 12/HK 13	0
AMAT 110	3
MATH 38	5
MATH 101	3
PHYS 51	4
PHYS 51.1	1
NSTP 1	0
AMAT 112	3
AMAT 152	3
AMAT 170	3
STS 1	3
NSTP 2	0
MATH 151	3
MATH 174	3
MATH 181	3
COMM 10	3
MATH 155	3
MATH 175	3
MATH 195	3
COMA 150	3
PI 10	3
AMAT 190	3
AMAT 198	3
AMAT 199	1
AMAT 105	3

Bachelor of Science in Applied Mathematics

Degree program code	BSAMAT
Major/Option	Thesis
Specialization units	24
GE elective units	9

Table 12. Required subjects explicitly listed in 2018 BSAMAT (Thesis) curriculum

Course Code	Credited Units
AMAT 19	3
MATH 36	5
BIO 11.1	2
KAS 1/HIST 1	3
ETHICS 1	3
HK 11	0
MATH 37	3
STAT 101	3
CHEM 18	3
CHEM 18.1	2
ARTS 1	3
HK 12/HK 13	0
AMAT 110	3
MATH 38	5
MATH 101	3
PHYS 51	4
PHYS 51.1	1
NSTP 1	0
AMAT 112	3
AMAT 152	3
AMAT 170	3
STS 1	3
NSTP 2	0
MATH 151	3
MATH 174	3
MATH 181	3
COMM 10	3
MATH 155	3
MATH 175	3
MATH 195	3
COMA 150	3
PI 10	3
AMAT 200	6
AMAT 198	3
AMAT 199	1
AMAT 105	3

Bachelor of Science in Applied Physics

Degree program code	BSAPHY
Major/Option	Thesis
Specialization units	21
GE elective units	9

Table 13. Required subjects explicitly listed in 2018 BSAPHY curriculum

Course Code	Credited Units
PHYS 101	4
CHEM 18	3
CHEM 18.1	2
MATH 27	3
ARTS 1	3
KAS 1/HIST 1	3
HK 11	0
PHYS 102	4
PHYS 111	4
MATH 28	3
COMA 150	3
HK 12/HK 13	0
APHY 10.1	1
APHY 101	3
PHYS 103	4
PHYS 112	4
ETHICS 1	3
NSTP 1	0
PHYS 142	3
APHY 191	3
PHYS 104	4
PHYS 113	4
PHYS 121	3
PHYS 131	3
NSTP 2	0
APHY 102	3
PHYS 115	4
PHYS 122	3
PHYS 132	3
PHYS 192.1	2
PI 10	3
PHYS 141	3
PHYS 151	3
PHYS 165	3
PHYS 195	3
COMM 10	3
APHY 198	3
APHY 200	6
APHY 199	1
STS 1	3

Bachelor of Science in Biology

Degree program code	BSBIO
Major/Option	Thesis
Specialization units	33
GE elective units	9

Table 14. Required subjects explicitly listed in 2018 BSBIO curriculum

Course Code	Credited Units
CHEM 18	3
MATH 25	3
MCB 11	3
KAS 1/HIST 1	3
ARTS 1	3
STS 1	3
HK 11	0
BIO 11.1	2
CHEM 18.1	2
BIO 30	3
PI 10	3
HK 12/HK 13	0
CHEM 40	4
CHEM 40.1	1
BIO 14	5
BOT 14	3
ZOO 14	3
CHEM 160	3
BIO 150	4
PHYS 51	4
PHYS 51.1	1
BIO 101	3
CHEM 160.1	2
BIO 140	3
ABME 10	3
NSTP 2	0
STAT 164	3
BIO 120	3
BIO 142	3
COMM 10	3
BIO 195	3
BIO 127	3
ETHICS 1	3
BIO 198	3
COMA 150	3
BIO 199	1
BIO 200	6
NSTP 1	0

Bachelor of Science in Chemistry

Degree program code	BSCHEM
Major/Option	Thesis
Specialization units	9
GE elective units	9

Table 15. Required subjects explicitly listed in 2018 BSCHEM curriculum

Course Code	Credited Units
CHEM 18	3
CHEM 18.1	2
MCB 11	3
MATH 27	3
ETHICS 1	3
KAS 1/HIST 1	3
HK 11	0
CHEM 19	3
CHEM 32	3
CHEM 32.1	2
PHYS 71	4
PHYS 71.1	1
ARTS 1	3
HK 12/HK 13	0
NSTP 1	0
MATH 28	3
CHEM 43	3
CHEM 43.1	2
PHYS 72	4
PHYS 72.1	1
NSTP 2	0
CMSC 12/ AMAT 152	3
CHEM 44	3
CHEM 44.1	2
CHEM 111	3
STAT 162	3
PI 10	3
CHEM 161A	3
CHEM 192	3
CHEM 111.1	2
CHEM 112	3
CHEM 137	3
CHEM 140	4
CHEM 161B	3
COMM 10	3
CHEM 112.1	2
CHEM 115	3
CHEM 137.1	2
CHEM 161.1	2
CHEM 180	3
CHEM 198	3
CHEM 120	3
CHEM 131	4
CHEM 120.1	2
STS 1	3
CHEM 171	3
CHEM 199	1
CHEM 200	6

Bachelor of Science in Computer Science

Degree program code	BSCS
Major/Option	Thesis
Specialization units	15
GE elective units	9

Table 16. Required subjects explicitly listed in 2018 BSCS (Thesis) curriculum

Course Code	Credited Units
CMSC 12	3
CMSC 56	3
ETHICS 1	3
MATH 27	3
STS 1	3
HK 11	0
ARTS 1	3
CMSC 21	3
CMSC 57	3
MATH 28	3
KAS 1/HIST 1	3
HK 12/HK 13	0
CMSC 22	3
CMSC 123	3
CMSC 130	3
CMSC 150	3
PI 10	3
NSTP 1	0
CMSC 23	3
ENG 10	3
CMSC 100	3
CMSC 127	3
CMSC 131	3
STAT 101	3
NSTP 2	0
COMM 10	3
CMSC 124	3
CMSC 125	3
CMSC 132	3
CMSC 141	3
CMSC 170	3
CMSC 128	3
CMSC 137	3
CMSC 142	3
CMSC 173	3
CMSC 180	3
CMSC 198	3
CMSC 199	1
CMSC 200	6

Bachelor of Science in Computer Science

Degree program code	BSCS
Major/Option	SP
Specialization units	18
GE elective units	9

Table 17. Required subjects explicitly listed in 2018 BSCS (SP) curriculum

Course Code	Credited Units
CMSC 12	3
CMSC 56	3
ETHICS 1	3
MATH 27	3
STS 1	3
HK 11	0
ARTS 1	3
CMSC 21	3
CMSC 57	3
MATH 28	3
KAS 1/HIST 1	3
HK 12/HK 13	0
CMSC 22	3
CMSC 123	3
CMSC 130	3
CMSC 150	3
PI 10	3
NSTP 1	0
CMSC 23	3
ENG 10	3
CMSC 100	3
CMSC 127	3
CMSC 131	3
STAT 101	3
NSTP 2	0
COMM 10	3
CMSC 124	3
CMSC 125	3
CMSC 132	3
CMSC 141	3
CMSC 170	3
CMSC 128	3
CMSC 137	3
CMSC 142	3
CMSC 173	3
CMSC 180	3
CMSC 198	3
CMSC 199	1
CMSC 190	3

Bachelor of Science in Mathematics

Degree program code	BSMATH
Major/Option	Thesis
Specialization units	15
GE elective units	9

Table 18. Required subjects explicitly listed in 2018 BSMATH (Thesis) curriculum

Course Code	Credited Units
MATH 20	3
MATH 36	5
BIO 11.1	2
KAS 1/HIST 1	3
ETHICS 1	3
HK 11	0
MATH 37	3
STAT 101	3
CHEM 18	3
CHEM 18.1	2
ARTS 1	3
HK 12/HK 13	0
MATH 38	5
MATH 101	3
PHYS 51	4
PHYS 51.1	1
NSTP 1	0
MATH 103	3
MATH 138	3
MATH 141	3
AMAT 152	3
STS 1	3
NSTP 2	0
MATH 111	3
MATH 155	3
MATH 195	3
COMA 150	3
COMM 10	3
MATH 120	3
MATH 133	3
MATH 151	3
PI 10	3
MATH 198	3
MATH 200	6
MATH 135	3
MATH 165	3
MATH 181	3
MATH 192	3
MATH 199	1

Bachelor of Science in Mathematics

Degree program code	BSMATH
Major/Option	SP
Specialization units	18
GE elective units	9

Table 19. Required subjects explicitly listed in 2018 BSMATH (SP) curriculum

Course Code	Credited Units
MATH 20	3
MATH 36	5
BIO 11.1	2
KAS 1/HIST 1	3
ETHICS 1	3
HK 11	0
MATH 37	3
STAT 101	3
CHEM 18	3
CHEM 18.1	2
ARTS 1	3
HK 12/HK 13	0
MATH 38	5
MATH 101	3
PHYS 51	4
PHYS 51.1	1
NSTP 1	0
MATH 103	3
MATH 138	3
MATH 141	3
AMAT 152	3
STS 1	3
NSTP 2	0
MATH 111	3
MATH 155	3
MATH 195	3
COMA 150	3
COMM 10	3
MATH 120	3
MATH 133	3
MATH 151	3
PI 10	3
MATH 198	3
MATH 190	3
MATH 135	3
MATH 165	3
MATH 181	3
MATH 192	3
MATH 199	1

Bachelor of Science in Mathematics & Science Teaching

Degree program code	BSMST
Major/Option	Biology
Specialization units	0
GE elective units	9

Table 20. Required subjects explicitly listed in 2018 BSMST (Biology) curriculum

Course Code	Credited Units	Course Code	Credited Units	Course Code	Credited Units
MST 101A	1	STAT 166	3	MST 200A	3
BIO 11.1	2	SPCM 156	3	MATH 18	3
CHEM 18	3	ETHICS 1	3	COMM 10	3
CHEM 18.1	2	NSTP 1	0	MST 190	3
PHYS 50	3	MST 101D	1	MST 191	3
MATH 25	3	DEVC 40/ MST 40	3	MST 200B	3
STS 1	3	EDUC 122	3	ARTS 1	3
PI 10	3	BOT 14	3	HFDS 12	3
HK 11	0	CHEM 160	3	PHYS 72	4
MST 101B	1	PHYS 71	4	PHYS 72.1	1
BIO 14	5	PHYS 71.1	1		
CHEM 40	4	COMA 150	3		
CHEM 40.1	1	NSTP 2	0		
AMAT 19	3	MST 123	5		
MATH 27	3	EDUC 144	3		
KAS 1/HIST 1	3	BIO 150	4		
HK 12/HK 13	0	MCB 11	3		
MST 101C	1	ZOO 14	3		
EDUC 102	3	STAT 162	3		
EDUC 111	3	MST 195	3		
BIO 30	3	MST 199	1		

Bachelor of Science in Mathematics & Science Teaching

Degree program code	BSMST
Major/Option	Chemistry
Specialization units	0
GE elective units	9

Table 21. Required subjects explicitly listed in 2018 BSMST (Chemistry) curriculum

Course Code	Credited Units	Course Code	Credited Units	Course Code	Credited Units
MST 101A	1	BIO 30	3	EDUC 144	3
BIO 11.1	2	STAT 162	3	CHEM 102	3
CHEM 18	3	SPCM 156	3	MATH 18	3
CHEM 18.1	2	ETHICS 1	3	COMM 10	3
PHYS 50	3	NSTP 1	0	MST 190	3
MATH 25	3	MST 101D	1	MST 200B	3
STS 1	3	DEVC 40/ MST 40	3	CHEM 180	3
PI 10	3	EDUC 122	3	ARTS 1	3
HK 11	0	STAT 166	3	MST 191	3
MST 101B	1	CHEM 160	3	MST 199	1
BIO 14	5	PHYS 71	4		
CHEM 40	4	PHYS 71.1	1		
CHEM 40.1	1	COMA 150	3		
AMAT 19	3	NSTP 2	0		
MATH 27	3	MST 123	5		
KAS 1/HIST 1	3	EDUC 111	3		
HK 12/HK 13	0	CHEM 160.1	2		
MST 101C	1	PHYS 72	4		
EDUC 102	3	PHYS 72.1	1		
CHEM 32	3	MST 195	3		
CHEM 32.1	2	MST 200A	3		

Bachelor of Science in Mathematics & Science Teaching

Degree program code	BSMST
Major/Option	Mathematics
Specialization units	0
GE elective units	9

Table 22. Required subjects explicitly listed in 2018 BSMST (Mathematics) curriculum

Course Code	Credited Units
MST 101A	1
BIO 11.1	2
CHEM 18	3
CHEM 18.1	2
PHYS 50	3
MATH 25	3
STS 1	3
PI 10	3
HK 11	0
MST 101B	1
BIO 14	5
CHEM 40	4
CHEM 40.1	1
AMAT 19	3
MATH 27	3
KAS 1/HIST 1	3
HK 12/HK 13	0
MST 101C	1
EDUC 102	3
EDUC 111	3
MATH 20	3
MATH 28	3
BIO 30	3
ETHICS 1	3
NSTP 1	0
MST 101D	1
EDUC 122	3
DEVC 40/ MST 40	3
MATH 18	3
MATH 101	3
STAT 166	3
COMA 150	3
NSTP 2	0
MST 123	5
EDUC 144	3
PHYS 71	4
PHYS 71.1	1
STAT 162	3
SPCM 156	3
MST 195	3
MST 200A	3
MATH 103	3
PHYS 72	4
PHYS 72.1	1
COMM 10	3
MST 190	3
MST 200B	3
CHEM 160	3
ARTS 1	3
MST 191	3
MST 199	1
AMAT 105	3

Bachelor of Science in Mathematics & Science Teaching

Degree program code	BSMST
Major/Option	Physics
Specialization units	0
GE elective units	9

Table 23. Required subjects explicitly listed in 2018 BSMST (Physics) curriculum

Course Code	Credited Units
MST 101A	1
BIO 11.1	2
CHEM 18	3
CHEM 18.1	2
PHYS 50	3
MATH 25	3
STS 1	3
PI 10	3
HK 11	0
MST 101B	1
BIO 14	5
CHEM 40	4
CHEM 40.1	1
AMAT 19	3
MATH 27	3
KAS 1/HIST 1	3
HK 12/HK 13	0
MST 101C	1
EDUC 102	3
EDUC 111	3
PHYS 71	4
PHYS 71.1	1
MATH 18	3
BIO 30	3
ETHICS 1	3
NSTP 1	0
MST 101D	1
EDUC 122	3
DEVC 40/ MST 40	3
PHYS 72	4
PHYS 72.1	1
CHEM 160	3
STAT 162	3
COMA 150	3
NSTP 2	0
MST 123	5
EDUC 144	3
APHY 101	3
STAT 166	3
SPCM 156	3
MST 195	3
MST 200A	3
APHY 102	3
PHYS 192.1	2
CMSC 12	3
COMM 10	3
MST 190	3
MST 199	1
MST 200B	3
PHYS 193.1	2
ARTS 1	3
MST 191	3

Bachelor of Science in Statistics

Degree program code	BSSTAT
Major/Option	SP
Specialization units	9
GE elective units	9

Table 24. Required subjects explicitly listed in 2018 BSSTAT curriculum

Course Code	Credited Units
ETHICS 1	3
KAS 1/HIST 1	3
MATH 27	3
BIO 30	3
STAT 101	3
HK 11	0
ARTS 1	3
MATH 28	3
CMSC 12	3
STAT 162	3
HK 12/HK 13	0
STAT 135	3
STAT 182	3
STS 1	3
CMSC 21	3
ECON 11	3
STAT 144	3
STAT 168	3
MATH 182	3
CMSC 22	3
ABME 10	3
STAT 145	3
STAT 163	3
NSTP 1	0
COMM 10	3
STAT 146	3
STAT 173	3
STAT 175	3
STAT 181	3
NSTP 2	0
CMSC 127	3
STAT 147	3
STAT 151	3
STAT 156	3
ENG 10	3
STAT 148	3
STAT 165	3
STAT 190	3
STAT 191	3
STAT 174	3
PI 10	3
STAT 157	3
STAT 167	3
STAT 183	3
STAT 199	1
STAT 192.1	1
STAT 198	3