PUP-TAGUIG FACULTY LOADING AND SCHEDULING SYSTEM

Software Requirements Specification

Version 1.0

Prepared by:

UNTITLED

Malaluan, Kyla Rica C.

Martinez, Emmanuel Miles.

Naoe, Adrian B.

Rasquero, Via Clariz.

August 25, 2024

Table of Content

Revision History

Version	Name	Reason for Changes	Date
1.0	UNTITLED	Initial Draft	August 25, 2024

Approved By

Name	Signature	Department	Date
Malaluan, Kyla Rica C.		BSIT – Quality Assurance | Developer	August 25, 2024
Martinez, Emmanuel Miles .		BSIT – Project Manager | Developer	August 25, 2024
Naoe, Adrian B.		BSIT – Technical Lead | Developer	August 25, 2024
Rasquero, Via Clarisse		BSIT – Technical Writer | Developer	August 25, 2024

1. Introduction

1.1 Purpose

This document presents the Software Requirement Specification (SRS) for the Polytechnic University of the Philippines Taguig Faculty Loading and Scheduling System (PUPT-FLSS) with the initial release of version 1.0. The primary objective of this document is to describe the system's design in detail, addressing both functional and non-functional requirements. The PUPT-FLSS aims to streamline the manual processes related to faculty loading and scheduling, replacing them with an efficient, web-based platform. This system integrates key features such as academic year management, faculty scheduling, and report generation, ensuring administrators can effectively manage faculty resources. This SRS is intended for the technical team and stakeholders involved in the system's development and testing

1.2 Document Conventions

Entire document should be justified.

- Convention for Main Title
  - Font face: Times
  - Font style: Bold
  - Font Size: 18
- Convention for Sub Title
  - Font face: Times
  - Font style: Bold
  - Font Size: 14
- Convention for Body
  - Font face: Times
  - Font Size: 12

1.3 Intended Audience and Reading Suggestions

This document is intended for the system developers, testers, stakeholders, and the project coordinator. It outlines the system's overall structure, key features, and functionalities.

Additionally, it details the functional and non-functional requirements that the system must fulfill. To assist in understanding the terminology used, a glossary is provided in the appendices, along with analysis models that visually represent the expected information flows. The document will undergo regular reviews to ensure alignment with project goals and client requirements. Any changes to the system requirements should be documented in the final version, serving as a reference for both development and validation teams.

1.4 Project Scope

The PUP Taguig Faculty Loading and Scheduling System (PUPT-FLSS) seeks to optimize the manual processes involved in managing faculty workloads and scheduling by digitizing and automating these tasks. This system offers an interface where faculty members can manage their schedules and administrators can assign teaching loads with greater efficiency. The PUPT-FLSS streamlines the organization, compilation, and management of faculty schedules while providing features such as curriculum management, room scheduling, and reporting capabilities. Through the automation of scheduling tasks, the system helps reduce administrative workload and minimize errors, ensuring a more accurate and efficient allocation of faculty resources

1.5 References

PUPT-FLSS

PUPT-FLSS-SRS

SRS Content Guide: https://uccs.edu/Documents/tboult/srs.doc

SRS Content Guide: https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.300-2.pdf

1. Overall Description

1. Product Perspective

The PUP Taguig Faculty Loading and Scheduling System (PUPT-FLSS) is a web-based application designed to streamline the faculty loading and scheduling processes. The system integrates with the Human Resource Information System (HRIS) to maintain consistency in faculty data and facilitate the creation of teaching schedules.

The process begins with the Director or HAP (Administrator) setting the active school year, semester, and curriculum within the system. Once this step is completed, faculty members are notified via email, enabling them to log in and select their preferences for courses and year levels. Faculty can choose their desired courses and corresponding year levels for teaching. If all selected courses and year levels are assigned, the system proceeds with the scheduling process. In the case where unassigned courses or year levels remain, administrators are notified to address the gaps. Once all selections are complete, the system automatically assigns schedules and generates reports, simplifying the administrative workload. This structured workflow reduces the need for manual interventions, minimizes scheduling conflicts, and allows faculty to play an active role in managing their assignments. Ultimately, the PUPT-FLSS enhances the overall efficiency of academic administration

2. Product Features

The PUP Taguig Faculty Loading and Scheduling System (PUPT-FLSS) supports both academic and administrative functions within the university. For the academic function, there are several major user roles, including Faculty Members, Program Directors, and Administrators. These users interact with the system to manage faculty schedules and workload assignments. On the administrative side, roles include Super Admin, Curriculum Administrators, and Academic Coordinators. These roles ensure the overall system functionality and manage scheduling oversight.

Faculty Members can log in to the system, select their preferred courses and year levels, and submit their teaching preferences. Program Directors and Administrators are responsible for reviewing and confirming these preferences, ensuring that all courses and

year levels are appropriately assigned to faculty. If there are unassigned courses, the system notifies the administrators, who can take corrective action by assigning them accordingly.

The Super Admin has full access to the system, allowing them to oversee all operations, make necessary changes, and ensure the smooth operation of the scheduling and faculty assignment process. Other administrative users, such as Curriculum Managers, help set the active academic year, semester, and curriculum, ensuring that faculty members have the correct courses and sections to choose from.

One of the core features of the PUPT-FLSS is its ability to automate faculty workload assignments and reduce manual intervention in the scheduling process. This digital system eliminates the risks associated with manual scheduling, such as miscommunication, data entry errors, and scheduling conflicts. By providing a centralized platform for managing course assignments, faculty members can submit their preferences earlier in the process, while administrators can efficiently review, adjust, and finalize the schedules.

The system's automation also significantly reduces the workload for reviewers and coordinators. With the capability to view real-time data and adjust schedules dynamically, administrators and faculty members can collaborate more efficiently, ensuring that course allocations are completed well before the start of each academic term. This shift from manual to digital scheduling streamlines the entire process, improving accuracy and reducing time spent on administrative tasks

3. User Classes and Characteristics

2.3.1 Faculty Employee

Faculty members are key users of the PUP Taguig Faculty Loading and Scheduling System (PUPT-FLSS). They can log in to the system to select their preferred courses and year levels, set their availability, and submit their teaching preferences for the academic term. Faculty members can also view their assigned schedules, make necessary adjustments before finalization, and download relevant reports such as teaching load summaries. Additionally, they can track their course assignments and ensure that their preferences align with the schedules set by the administration.

2.3.2 Admin

Admins play a critical role in managing the system. Their responsibilities include overseeing the scheduling process, reviewing faculty-submitted preferences, and resolving any conflicts or gaps in the course assignments. Admins are responsible for setting the active school year, semester, and curriculum, ensuring that all necessary courses and sections are available for faculty to choose from. They can also generate reports on faculty assignments and loads, making sure that schedules adhere to institutional policies and guidelines.

2.3.3 Super Admin

The Super Admin holds the highest level of control within the PUPT-FLSS. They have full access to all system functionalities and are responsible for managing the system's structure and settings. This includes creating and updating user roles, permissions, and accounts, as well as configuring academic programs, sections, and curriculums. The Super Admin can oversee the entire scheduling process and generate comprehensive reports on system usage, faculty loads, and scheduling performance. They also handle the system's maintenance, ensuring that updates and adjustments are made as necessary to support the overall functionality of the system

2.4 Operating Environment

1. The PUP Taguig Faculty Loading and Scheduling System (PUPT-FLSS) operates in a web-based environment.
2. The PUPT-FLSS can be accessed via any modern web browser with a stable internet connection.
3. The system is developed using the Laravel framework, a robust PHP-based framework, for backend functionalities.
4. The front-end interface of the system is built using Angular, ensuring responsiveness and user-friendly interaction.
5. The system uses MySQL as its database management system, which stores all faculty-related data, schedules, and curriculum information.
6. The local development environment is supported by XAMPP, which includes Apache, MySQL, and PHP

2.5 Design and Implementation Constraints

1. The PUP Taguig Faculty Loading and Scheduling System (PUPT-FLSS) will be designed in compliance with the institution's capstone project guidelines, ensuring adherence to established best practices and standards.
2. The PUPT-FLSS will be developed using the Laravel framework for the backend and Angular for the frontend, ensuring a responsive and dynamic user interface.
3. The database engine for PUPT-FLSS will be powered by MySQL, providing a reliable and scalable solution for managing and retrieving data related to faculty loading and scheduling processes.
4. The system will be maintained during the implementation period, and appropriate training will be provided to ensure users can efficiently interact with the system upon completion.

2.6 User Documentation

1. Manuals and How-to video tutorials will be provided once the project is finished.

2.7 Assumptions and Dependencies

1. The PUP Taguig Faculty Loading and Scheduling System (PUPT-FLSS) will facilitate the scheduling process for faculty and administrators, ensuring efficient management of faculty workloads and generating automated reports for faculty assignments.
2. The PUPT-FLSS will require a stable internet connection to function correctly. Any disruptions in internet connectivity, such as a weak or lost connection, may affect the system’s performance.
3. The system relies on the seamless integration with the Human Resource Information System (HRIS) to ensure that faculty data is up-to-date for scheduling purposes. Any delays or errors in data synchronization between PUPT-FLSS and HRIS could impact the scheduling process.
4. The proper functioning of the system will depend on regular maintenance and updates, ensuring that any issues or bugs are promptly addressed to minimize disruptions in faculty scheduling operations.

3. System Features

This section outlines the functional requirements of the application and details the features it offers. Each feature is described thoroughly to support future system enhancements and testing. The system's features are prioritized and organized based on their importance, as outlined below.

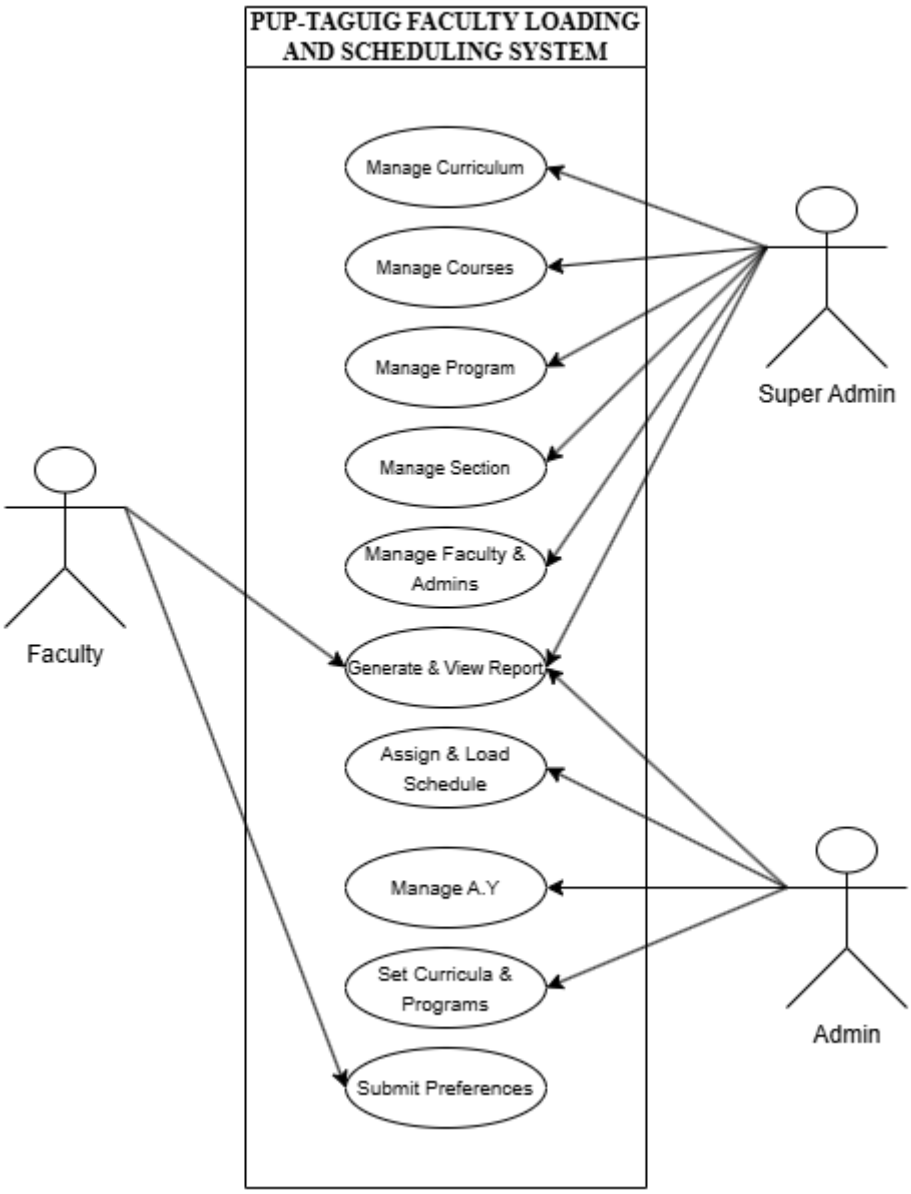

Figure 2. System Use Case Diagram

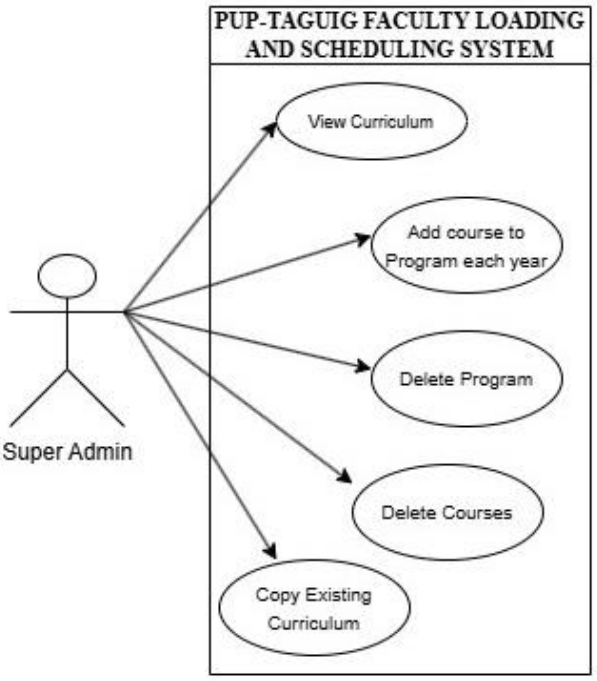

Figure 3. Super Admin Curriculum & Course Assignment Management

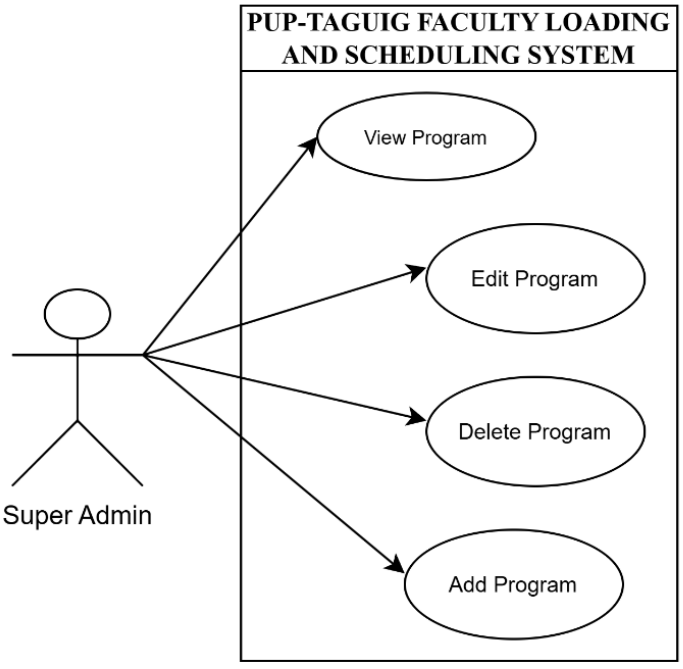

Figure 4. Super Admin Program Management

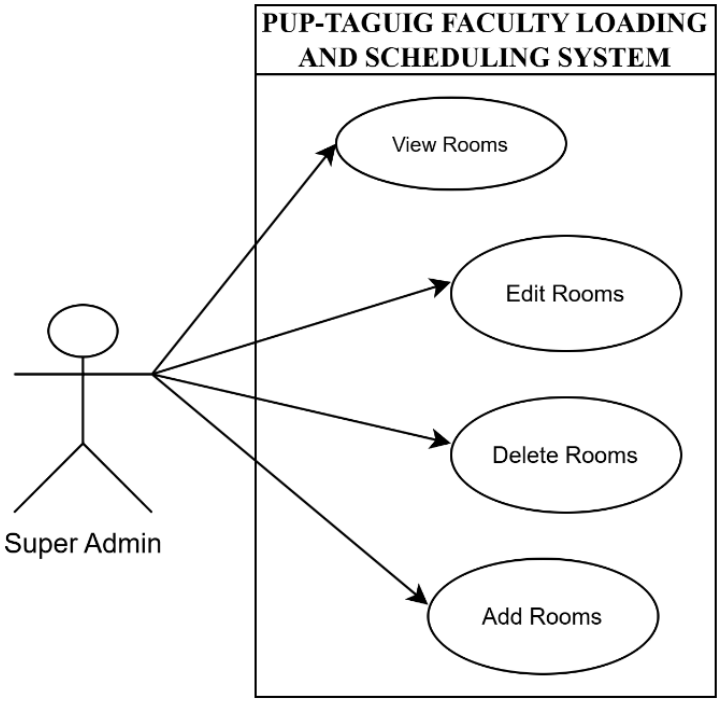

Figure 5. Super Admin Section Management

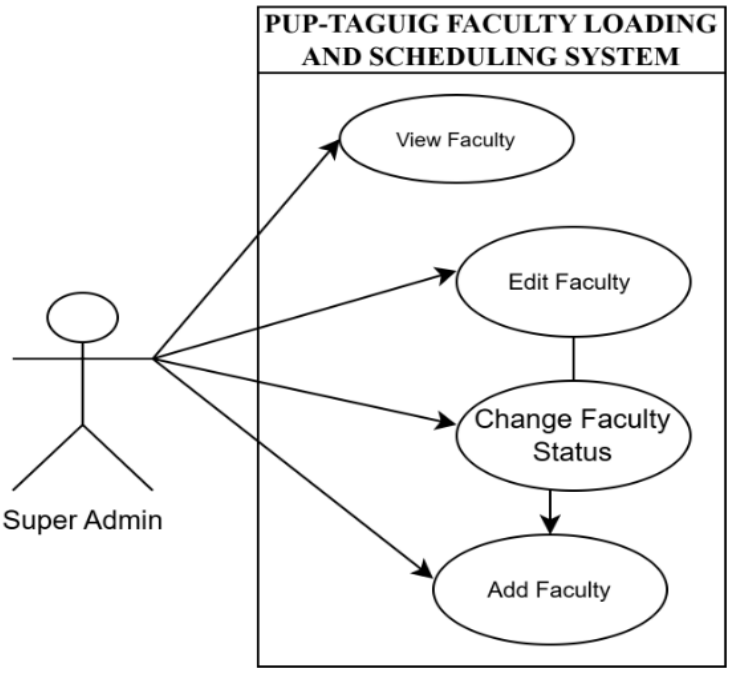

Figure 6. Super Admin Faculty Credentials Management

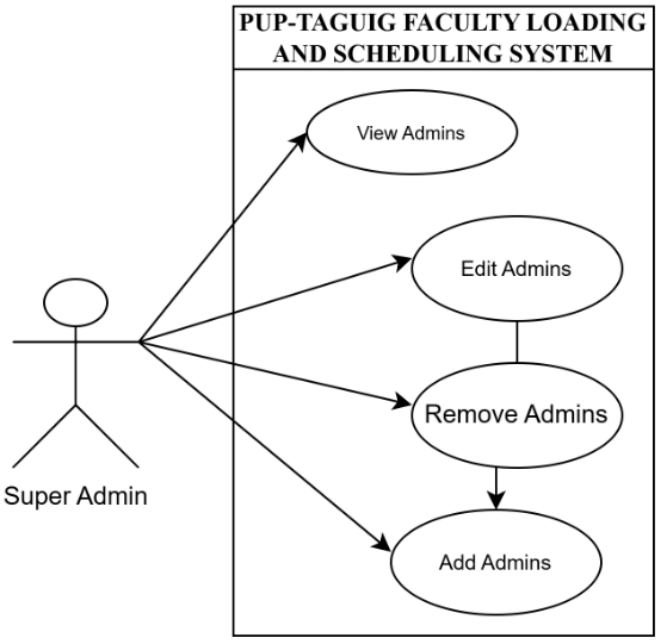

Figure 7. Super Admin, Admin and Super Admin Credentials Management

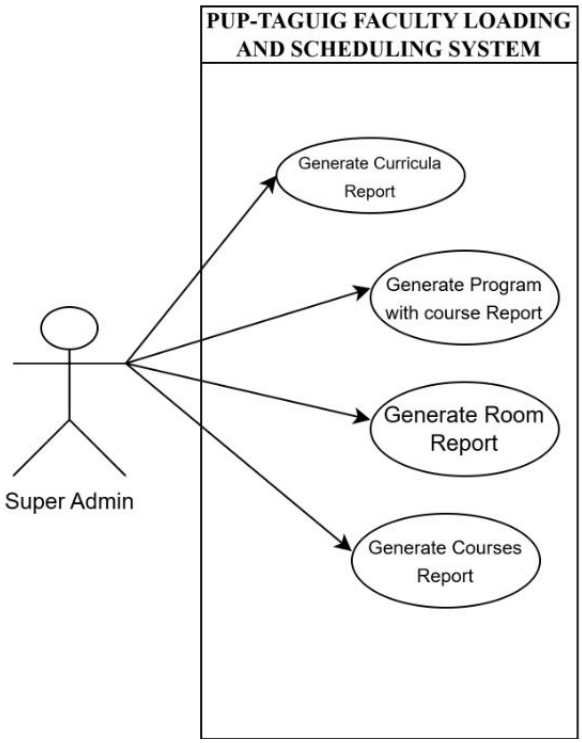

Figure 8. Super Admin Report Management

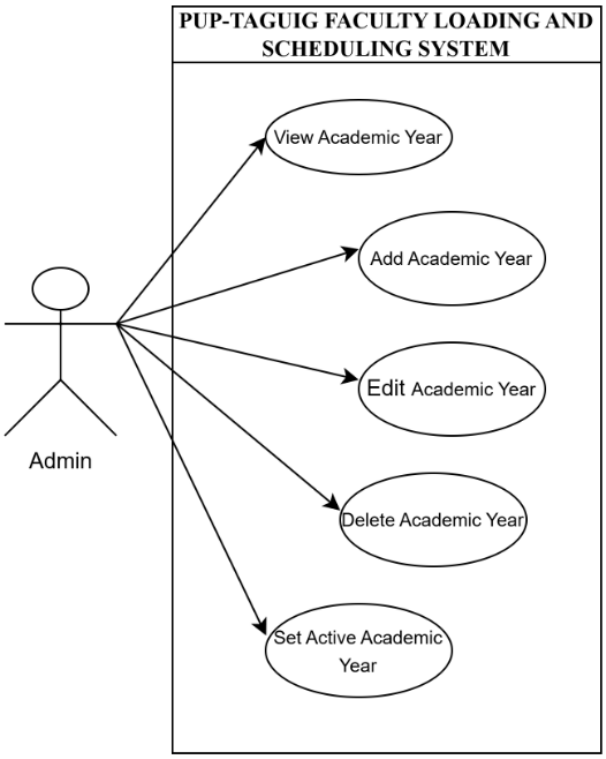

Figure 9. Admin Academic Year Management

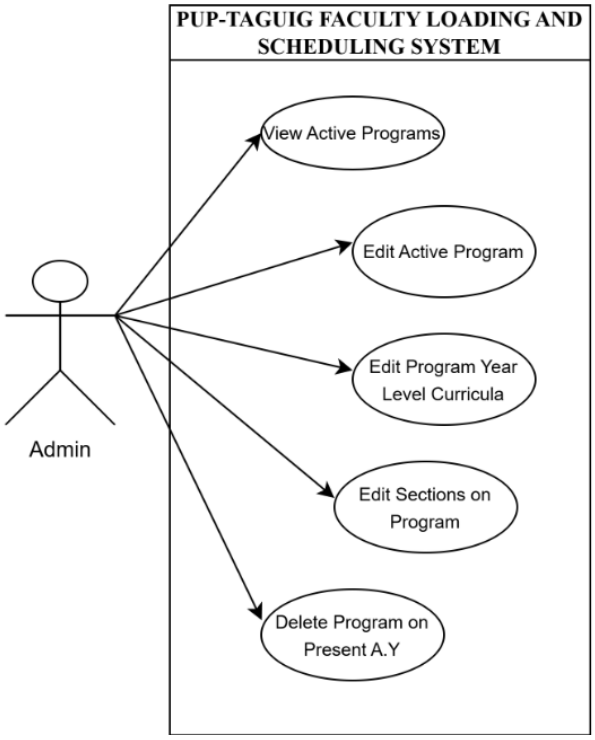

Figure 10. Admin Active Program Management

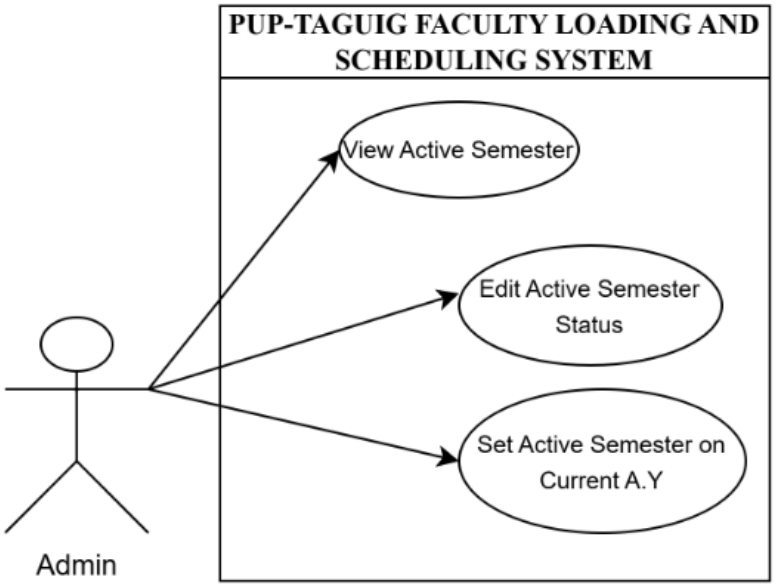

Figure 11. Admin Active Semester Management

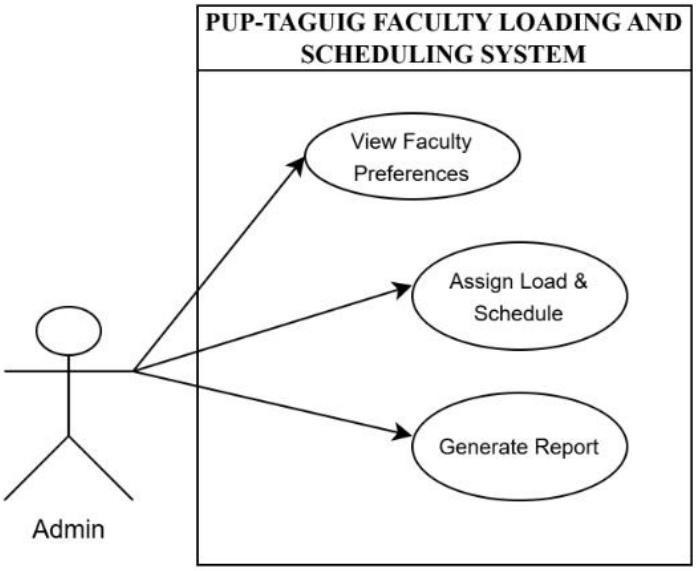

Figure 11. Admin Faculty Loading & Scheduling Management

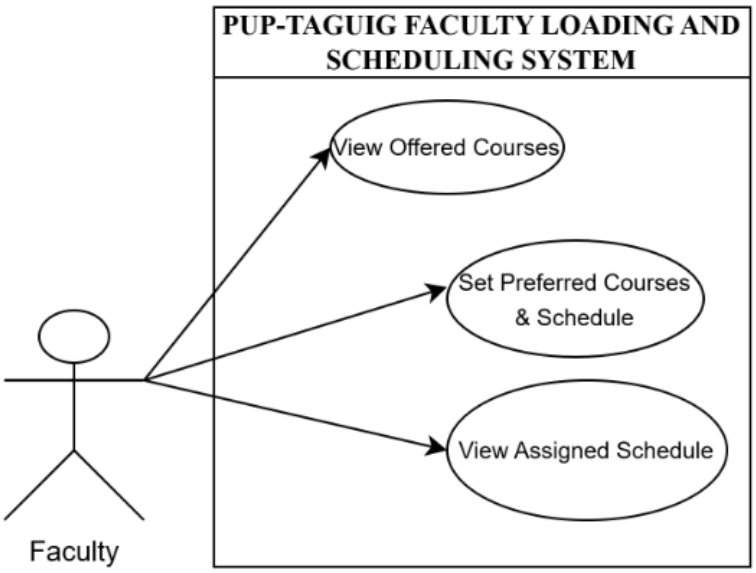

Figure 12. Faculty Preferences Submission

3.1 Faculty Dashboard

3.1.1 Description and Priority

The Faculty Dashboard serves as the main interface for faculty members to access essential functions related to teaching schedules, preferences, and reports

3.1.2 Stimulus/Response Sequences

Stimulus	**Response**
The Faculty Member will login through PUPT-FLSS	The Faculty shall be redirected to Home/Dashboard with all the cards regarding loading & scheduling.

3.1.3 Functional Requirements

1. The system shall allow the Faculty Members to view the interface of PUPT-FLSS.
2. The dashboard cards must reflect the correct information and reports by admins.

3.2 Submissions of Preferences Module through PUPT-FLSS

3.2.1 Description and Priority

This is where faculty view offered courses and submit they desired schedule through the active semester in specific academic year.

3.2.2 Stimulus/Response Sequences

Stimulus	**Response**
The Faculty Member will view the available courses and submit their preferred schedule through the active semester for a specific academic year.	The system will display the list of offered courses for the current semester and academic year. It will allow faculty members to select their preferred teaching schedule, submit the choices, and provide confirmation. The system will automatically validate the selection against existing schedules for conflicts and notify the faculty member of the successful submission.

3.2.3 Functional Requirements

1. The system shall allow the Faculty Members to view their assigned course load and teaching schedules in real time.
2. The system shall notify Faculty Members when their teaching load or assigned schedule has been updated or modified.
3. The system shall allow administrators to review and approve faculty requests for schedule adjustments or changes.
4. The system shall allow Faculty Members to request specific classrooms or resources for their scheduled classes, with administrator approval.

3.3 Admin Dashboard

3.3.1 Description and Priority

The Admin Dashboard serves as the main interface for admin to access essential functions related to teaching schedules, preferences, and reports

3.3.2 Stimulus/Response Sequences

Stimulus	**Response**
The Faculty Admin will login through PUPT-FLSS	The Admin shall be redirected to Home/Dashboard with all the cards regarding loading, scheduling and reports.

3.3.3 Functional Requirements

1. The system shall allow the Admin of PUPT-FLSS to view the interface of PUPT-FLSS.
2. The dashboard cards must reflect the correct information and reports by super admins.

3.4 Admin Active Academic Management

3.4.1 Description and Priority

The Admin Active Academic Management module allows administrators to manage key academic settings such as school year, curriculum, and course offerings. This serves as the control center for the academic calendar and ensures that the appropriate academic configuration is active for faculty loading and scheduling. This function is essential for keeping the system up to date with the current academic cycle.

3.4.2 Stimulus/Response Sequences

Stimulus	**Response**
The admin sets the active academic school year and semester through the FLSS.	The system shall activate the selected academic year and semester, notifying faculty and students of the updated academic schedule.

The admin sets the active curriculum for a specific academic program and year level.	The system shall save the selected curriculum and apply it to the chosen program and year level, making it available for faculty to schedule courses.

3.4.3 Functional Requirements

1. The system shall allow the admin to set and manage the active academic school year and semester.
2. The system shall allow the admin to select the active curriculum for each program and year level.
3. The system shall notify faculty members via email once the academic configuration is updated.
4. The system shall display a confirmation message to the admin once academic settings have been successfully applied.
5. The system shall prevent faculty loading and scheduling operations if the academic year is not set as active.
6. The system shall generate error messages if any discrepancies are detected in the academic setup, such as missing courses or invalid curriculum entries.

3.5 Admin Faculty Loading and Scheduling Management

3.5.1 Description and Priority

The Admin Faculty Loading and Scheduling Management module is the primary interface for administrators to manage faculty course assignments and scheduling. This module allows administrators to assign courses to faculty members, resolve scheduling conflicts, and ensure that teaching loads adhere to institutional guidelines. It is crucial for the efficient management of academic resources and faculty workloads.

3.5.2 Stimulus/Response Sequences

Stimulus	**Response**
The admin assigns courses to a faculty member through the scheduling module.	The system shall assign the course to the selected faculty member and check for any scheduling conflicts or overloads.
The admin generates a report on faculty teaching loads for the current semester.	The system shall generate a comprehensive report detailing the teaching loads, and course assignments.

3.5.3 Functional Requirements

1. The system shall allow the admin to assign courses to faculty members based on their qualifications and availability.
2. The system shall automatically check for scheduling conflicts and notify the admin if any are detected.
3. The system shall allow the admin to adjust teaching loads and schedules, ensuring compliance with institutional policies.
4. The system shall generate detailed reports on faculty teaching loads, course assignments, and schedule adjustments.
5. The system shall notify faculty members via email when their schedule is updated or a new course is assigned.

3.6 Super Admin Dashboard

3.6.1 Description and Priority

The Super Admin Dashboard serves as the main interface for admin to access essential functions related to teaching schedules, preferences, and reports

3.6.2 Stimulus/Response Sequences

Stimulus	**Response**
The Super Admin will login through PUPT-FLSS	The Super Admin shall be redirected to Home/Dashboard with all the cards regarding loading, scheduling and reports.

3.6.3 Functional Requirements

3. The system shall allow the Super Admin of PUPT-FLSS to view the interface of PUPT-FLSS.
4. The dashboard cards must reflect the correct information and reports by other super admins.

3.7 Faculty, Admins & Super Admin Management Module

3.7.1 Description and Priority

The Faculty Management module is the interface that allows super admins and admins to manage administrative user accounts within the system. This module provides essential features such as creating, editing, and deleting admin accounts, along with the ability to manage their roles and statuses. The module ensures that only authorized personnel have access to the administrative functions of the Faculty Loading and Scheduling System.

3.7.2 Stimulus/Response Sequences

Stimulus	**Response**
The super admin clicks the "Add Admin" button to create a new admin account	The system shall display a form for entering the admin code, name, password, role (admin or super admin), and status (active or inactive). Once saved, the new admin will be added to the list.

The admin clicks the "Edit" button next to an admin account.	The system shall display the editable details for the selected admin account. After modifications are saved, the system shall update the account.
The admin clicks the "Delete" button next to an admin account.	The system shall display a confirmation prompt. If confirmed, the system shall permanently remove the admin account from the list.

3.7.3 Functional Requirements

1. The system shall allow super admins to add new admin accounts by entering their details and assigning roles and statuses.
2. The system shall allow admins and super admins to edit existing admin accounts, including changing roles and status.
3. The system shall display a list of all admins with their respective admin codes, names, roles, and statuses.
4. The system shall allow super admins to delete admin accounts, displaying a confirmation prompt before the action.
5. The system shall provide the option to export the list of admins to a PDF for reporting or documentation purposes.
6. The system shall notify admins of any changes to their account via email (e.g., role changes, deactivation).
7. The system shall provide a search and pagination feature to browse and manage a large number of admin accounts efficiently.

3.8 Curriculum & Course Assignment Module

3.8.1 Description and Priority

The Curriculum & Course Assignment module allows administrators to manage academic curricula and assign courses to specific programs and year levels. This module is critical for

ensuring that the correct courses are assigned to students based on the academic year and program requirements. Administrators can create, edit, and delete curricula, as well as assign courses to different semesters and academic programs.

3.8.2 Stimulus/Response Sequences

Stimulus	**Response**
The admin clicks "Add Curriculum" to create a new curriculum for a specific year.	The system shall display a form to input the curriculum year, which will be saved and added to the list of available curricula.
The admin clicks "Add Course" to assign a new course to a specific semester.	The system shall display a form to input the course code, title, and associated details like lecture hours and lab hours. The course will then be added to the selected semester.
The admin clicks "Edit" next to a course in the curriculum.	The system shall display the course details in an editable form, allowing the admin to modify any necessary information. Once saved, the changes will be reflected in the curriculum overview.
The admin clicks "Delete" next to a course or curriculum.	The system shall display a confirmation prompt. Once confirmed, the course or curriculum will be removed from the system.

3.8.3 Functional Requirements

1. The system shall allow the admin to create new curricula by inputting the curriculum year and saving it to the system.
2. The system shall allow the admin to assign courses to a specific program, year level, and semester.
3. The system shall display all courses assigned to a curriculum, including details such as lecture hours, lab hours, units, and tuition hours.
4. The system shall allow the admin to edit existing curricula and course details, updating information as needed.

5. The system shall provide the option to delete courses or curricula with a confirmation prompt to prevent accidental deletions.
6. The system shall allow the admin to export the curriculum and course assignments to a PDF for documentation purposes.
7. The system shall display error messages if invalid data is entered while assigning or editing a course.
8. The system shall support pagination and search functionality to easily browse and manage a large number of courses and curricula.

3.9 Super Admin Program Module

3.9.1 Description and Priority

The Programs Module allows administrators to manage the academic programs offered by the institution. This module enables admins to create, edit, and delete programs, as well as provide key information such as program descriptions, status, and duration. The module ensures that all academic programs are properly categorized and maintained for faculty loading, scheduling, and curriculum assignment purposes.

3.9.2 Stimulus/Response Sequences

Stimulus	**Response**
The Super admin clicks "Add Program" to create a new academic program.	The system shall display a form for inputting the program code, title, description, years of study, and status. Upon saving, the program will be added to the list
The Super admin clicks "Edit" next to an existing program.	The system shall display the program details in an editable form. After saving the changes, the program will be updated in the list.

The Super admin clicks "Delete" next to a program.	The system shall display a confirmation prompt. Upon confirmation, the program will be removed from the list.
The Super admin clicks "Export to PDF" to generate a report of all programs.	The system shall generate a downloadable PDF containing a list of all programs and their details.

3.9.3 Functional Requirements

1. The system shall allow the admin to add new academic programs by entering the program code, title, description, years of study, and status.
2. The system shall allow the admin to edit the details of existing programs.
3. The system shall display a list of all programs with their respective program codes, titles, descriptions, statuses, and duration (years).
4. The system shall allow the admin to delete programs, with a confirmation prompt to avoid accidental deletions.
5. The system shall provide a search function to allow admins to easily locate specific programs in the list.
6. The system shall allow the admin to export the list of programs to a PDF for documentation or reporting purposes.
7. The system shall display error messages if invalid data is entered during program creation or editing.
8. The system shall support pagination to browse through a large number of academic programs efficiently.

3.10 Super Admin Room Management

3.10.1 Description and Priority

The Rooms Module enables administrators to manage classroom and laboratory facilities within the institution. This module allows admins to add, edit, and delete room information, including the room code, location, floor level, room type, capacity, and availability status. The

module is essential for ensuring that all rooms are properly categorized and available for course scheduling and faculty assignments.

3.10.2 Stimulus/Response Sequences

Stimulus	**Response**
The Super Admin will login through PUPT-FLSS	The system shall display a form where the admin can input the room code, location, floor level, room type, capacity, and status. Upon saving, the new room will be added to the list.
The admin clicks "Edit" next to a room entry	The system shall display the room details in an editable form. After making changes and saving, the room details will be updated in the list.
The admin clicks "Delete" next to a room entry.	The system shall display a confirmation prompt. Upon confirmation, the room will be removed from the list.
The admin clicks "Export to PDF" to generate a report of all rooms.	The system shall generate a downloadable PDF containing the list of rooms and their details for reporting purposes.

3.10.3 Functional Requirements

1. The system shall allow the admin to add new rooms by inputting the room code, location, floor level, room type, capacity, and status.
2. The system shall allow the admin to edit the details of existing rooms.
3. The system shall display a list of all rooms, including their room codes, locations, floor levels, room types (e.g., lecture or lab), capacities, and statuses.
4. The system shall allow the admin to delete rooms, with a confirmation prompt to prevent accidental deletions.

5. The system shall allow the admin to export the list of rooms to a PDF file for documentation and reporting purposes.
6. The system shall provide a search feature to allow the admin to quickly find specific rooms based on the room code or location.
7. The system shall support pagination to efficiently browse and manage a large number of room entries.
8. The system shall display error messages if invalid data is entered while adding or editing a room.

4. External Interface Requirements

4.1 User Interfaces

- The PUPT Faculty Loading and Scheduling System (PUPT-FLSS) will be used by Faculty Members, Super Admins, Department Heads (e.g., Dean, Chair, or HAP, Directors), Program Coordinators, and Admin Staff. Users must be able to log in and log-out out of the system.

- Users must be able to log in and log out of the system using their unique credentials.

4.2 Hardware Interfaces

Requirement	**Recommended**
Processor	Minimum of 2.5 GHz quad-core processor or higher, or a minimum of dual-core processor with hyper-threading support
Memory	Minimum of 8GB of RAM or higher to support modern web development frameworks such as Angular and Laravel

Hard Disk	Minimum of 500MB of free storage for project files, with at least 20GB available for system operations and caching
Screen Resolution	Minimum of 1080p resolution for optimal user interface and design workflow

4.3 Software Interfaces

Requirement	**Recommended**
Client on Internet	Any modern web browser (e.g., Google Chrome, Firefox, Microsoft Edge)
Client on Intranet	Any modern web browser (e.g., Google Chrome, Firefox, Microsoft Edge)
Web Server	Apache or Nginx with PHP support, XAMPP for local development
Database Server	MySQL or MariaDB Database Server
Development End	Angular for frontend development and Laravel Framework for backend development

4.4 Communications Interfaces

- The device used to access the web application must be connected to the internet.

5. Other Nonfunctional Requirements

This section describes the non-functional requirements for the PUPT Faculty Loading and Scheduling System (PUPT-FLSS). Performance, safety, security, and usability requirements are defined to align with stakeholders' needs to enhance the system's qualities. Business rules are also indicated to ensure smooth and efficient operations within the system.

5.1 Performance Requirements

- The system shall respond to user actions within 3 seconds on average, under normal operational loads.
- The system must be able to handle up to 200 concurrent users without degradation in performance.
- The backend (Laravel) must efficiently handle database operations, ensuring that queries are processed and returned within 2 seconds under normal conditions.
- The system shall optimize API calls between the Angular frontend and the Laravel backend to ensure minimal latency.

5.2 Safety Requirements

- The system shall be backed up daily to prevent data loss in case of hardware failure or system crash.
- Regular checks must be in place to ensure system integrity and recovery measures must be defined to ensure minimal downtime.

5.3 Security Requirements

- The PASUC NBC 461 Evaluation Management Information System must maintain confidentiality to its data and will only be given to appropriate authority.
- The system must be secure with email and password that is limited only to its users.

5.4 Usability Requirements

- The PASUC NBC 461 Evaluation Management Information System shall be user-friendly and easy to navigate without

5.5 Business Rules

- Faculty members shall only be allowed to submit their schedule preferences during the designated scheduling period.
- Admins must review and approve all schedule submissions before finalization.
- The system shall enforce a maximum teaching load per faculty member according to institutional policies.
- Users shall receive email notifications for important updates, such as schedule approvals, modifications, or system maintenance notices.
-

6. Other Requirements

- No other requirements have been identified.

7. Glossary

Term/Acronym	Definition
Faculty Loading	The process of assigning teaching responsibilities and workloads to faculty members based on their qualifications, availability, and institutional requirements.
Laravel Framework	A popular PHP framework used for building web applications, known for its elegant syntax and adherence to the Model-View-Controller (MVC) architectural pattern.
Angular	A front-end web application framework developed by Google, used for building dynamic and responsive user interfaces for web applications.
MySQL	An open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for managing and retrieving data

XAMPP	A free and open-source cross-platform web server solution stack package that includes Apache, MySQL, PHP, and Perl, used for developing and testing web applications locally.
Model-View-Controller (MVC)	A software architectural pattern that separates an application into three interconnected components: the model (data), the view (user interface), and the controller (business logic), promoting organized code and separation of concerns