PUP-TAGUIG FACULTY LOADING AND SCHEDULING SYSTEM

Software Test Plan

Version 1.0

Prepared by:

UNTITLED

Malaluan, Kyla Rica C.

Martinez, Emmanuel

Naoe, Adrian B.

Rasquero, Via Clariz A.

August 25, 2024

Table of Content

Revision History

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

Approved By

Name	Signature	Department	Date
Malaluan, Kya Rica		BSIT – Quality Assurance | Developer	
Martinez, Emmanuel		BSIT – Project Manager | Developer	
Naoe, Adrian		BSIT – Technical Lead | Developer	
Rasquero, Via Clariz		BSIT – Technical Writer | Developer	

1. Introduction

This is the Software Test Plan document for the PUP Taguig Faculty Loading and Scheduling System (PUPT FLSS). It contains a detailed and executable strategy for conducting testing activities of the system. Its main focus is to test the functionality of the PUPT Faculty Loading and Scheduling System. This includes the course scheduling, faculty workload management, report generation, and system performance under normal and peak conditions.

2. Test Items

This activity focuses on the following modules and functionalities:

1. Scheduling Module
2. Program Management
3. Admin Management
4. Faculty Management
5. Curriculum Management
6. Rooms Management
7. Reports Generation

Item to Test	Test Description	Responsibility
Scheduling Module		
Scheduling Module	Verify that the system allows administrators to schedule classes without conflicts in time, room availability, or faculty assignments. Ensure the scheduling adheres to institutional policies and displays conflicts or overlaps correctly to the user.	To resolve any conflicts or errors in the scheduling process and ensure synchronization across modules.
Program Management		
Program Management	To verify that new programs can be created, updated, or deleted correctly, ensuring data integrity and consistency across all associated records.	To correct any data inconsistency issues and maintain the integrity of program information.
Admin Management		
Admin Management	To check if admin roles are correctly assigned and that permissions are enforced according to predefined policies, including access to sensitive data and management capabilities.	To fix any unauthorized access or permission issues and ensure proper role assignment.
Faculty Management		
Faculty Management	To ensure that faculty details are accurately maintained and that faculty can be correctly assigned to courses and schedules without conflicts.	To address any discrepancies in faculty assignments and maintain accurate faculty records.

Curriculum Management		
Curriculum Management	To confirm that courses and curriculums can be set up, modified, or deleted as per requirements, and that these changes are reflected across all relevant modules.	To resolve any issues with curriculum updates and ensure data consistency.
Rooms Management		
Rooms Management	To validate that rooms are correctly allocated for classes, exams, and events, without conflicts or overbooking, and that room availability is accurately reflected in the system.	To correct any room allocation errors and ensure accurate room availability status.
Reports Generation		
Reports Generation	To verify that reports are generated accurately, reflecting up-to-date data from all modules, and are available in the required formats for stakeholders.	To address any errors in report generation and ensure that data is correctly presented.

3. Software Risk Issues

Several factors are outside the control of the PUP Taguig Faculty Loading and Scheduling System but have direct impacts on its functionality and performance and must be monitored closely:

- The system's performance, including load times and data synchronization, may be affected by the speed and stability of the internet connection, as it is a web-based application.
- The system is optimized for use with modern web browsers like Google Chrome and Microsoft Edge; using unsupported or outdated browsers may cause slower performance or unexpected behavior.

- Devices not meeting the minimum hardware and software requirements (e.g., sufficient RAM, processor speed, or storage) may experience slow performance, crashes, or other operational issues.
- Regular backups of the database are required to prevent data loss in the event of a system failure, hardware malfunction, or unexpected shutdown.
- The system's user authentication relies on secure email and password combinations; compromised credentials may lead to unauthorized access, potentially impacting data privacy and security.
- Unexpected changes in university policies, academic calendars, or regulations could require immediate system updates, posing a risk to the stability and integrity of the system until the changes are implemented.
- Integration issues may arise with external systems, such as the Human Resource Information System (HRIS), if they are not maintained or updated consistently, causing discrepancies in faculty data.
- High usage during peak periods, such as the start of a semester or during faculty scheduling, may lead to system overload, causing delays or temporary inaccessibility.

4. Features to be Tested

The following areas will be the primary focus during the testing of the Faculty Loading and Scheduling System:

- **Submission of Faculty Preferences:** Test the submission process where faculty members select their preferred courses, time slots, and year levels for the semester.
- **Review of Faculty Assignments:** Verify that administrators can review and approve faculty assignments, ensuring no conflicts in scheduling or workload.
- **Generation of Scheduling Reports:** Check the generation of various reports, including faculty loads, room allocation, and overall schedules for different programs.
- **Maintenance of the System Data:** Test the ability to add, update, or delete faculty data, course information, and program details in the system.
- **Room Allocation Functions:** Ensure the system correctly allocates rooms for classes and events without double-booking or conflicts.
- **Alerts and Notifications:** Validate that the system sends alerts and notifications to faculty and administrators for schedule changes, approvals, or pending actions.

5. Features Not to be Tested

The following areas will not be specifically addressed during testing. All testing in these areas will be indirect, resulting from other testing efforts:

- **Security testing against malware and external threats** will not be included, as the focus will be on functional and performance testing.
- **Functionalities already provided by integrated external systems,** such as the Human Resource Information System (HRIS) for faculty data management.
- **User actions that redirect to other pages or involve using different functionalities** while in the middle of an ongoing process, such as submitting faculty preferences or approvals.
- **System performance issues due to bad internet connections** or failures to save changes/upload documents caused by network instability.

6. Test Approach

Given the short development timeframe and the necessity to deliver a functional application that may require frequent changes based on university policies and academic calendar adjustments, the development team has chosen to utilize the Agile Software Development Life Cycle (SDLC) methodology. This approach consists of five key phases in the test cycle: Test Planning, Daily Scrums, Test Agility Review, Release Readiness, and Impact Assessment.

Test Planning: The development team begins by planning and preparing the testing schedule, defining the processes, and determining the expected outcomes. This phase involves identifying the scope of testing, allocating resources, and outlining detailed test cases based on system requirements and user stories. The team focuses on areas such as faculty data management, scheduling conflicts, and report generation.

Daily Scrums: To ensure continuous alignment and communication throughout the development process, the team conducts daily scrums. These brief meetings, typically lasting 15-30 minutes, allow team members to share progress, discuss any issues or obstacles, and adjust plans as needed. Developers and testers can quickly clarify misunderstandings about processes or the codebase, facilitating a more efficient workflow.

Test Agility Review: After each sprint or development cycle, the testing team and developers review the testing process itself. They assess the effectiveness of their testing strategies, identify areas for

improvement, and adapt their approach based on evolving requirements, such as changes in academic programs or faculty assignments. This phase ensures that the testing process remains flexible and responsive to project needs.

Release Readiness: The team ensures that the Faculty Loading and Scheduling System is ready for deployment by conducting final testing cycles, addressing any critical bugs, and verifying functionality against predefined acceptance criteria. This phase includes preparing deployment documentation, user manuals, and training materials to support a smooth rollout for both faculty members and administrators.

Impact Assessment: After deployment, the team evaluates the system's performance and gathers user feedback. They monitor system usage, identify any post-deployment issues, and assess how well the system meets its intended objectives, such as improved faculty scheduling efficiency and reduced conflicts. Based on this feedback, the team makes necessary adjustments and plans for future updates or enhancements.

7. Item Pass/Fail Criteria

All functionalities of the Faculty Loading and Scheduling System should perform as expected. A pass rate of at least 95% across all test cases is required. Any test cases that fail must not critically impact the core functionalities of the system, such as faculty data management, scheduling, and report generation, or significantly hinder the user’s ability to use the application effectively.

8. Suspension Criteria and Resumption Requirements

- A critical defect is identified, and the necessary code fix requires substantial re-testing of the affected functions, particularly those involving faculty scheduling or data management.
- Major modifications are needed due to changes in business or technical specifications, requirements, or scope, resulting from escalated test issues or additional requests made by stakeholders.
- External dependencies, such as integrations with the Human Resource Information System (HRIS) or other third-party services, become unavailable or unresponsive.
- The test environment encounters significant issues, such as server outages, hardware failures, or any other problem that reduces system performance to below 50% of its normal operating capacity.

- Testing will resume once the critical problems are resolved, necessary modifications are implemented, external dependencies are restored, or the test environment is stabilized and fully functional.

9. Test Deliverables

- Test Cases Documentation
- Test Scenarios / Use Cases
- Software Test Report
- Software Test Plan

10. Remaining Test Tasks

- Conduct comprehensive re-testing of the Faculty Loading and Scheduling System to ensure all identified issues have been resolved.
- Report newly discovered bugs and issues to the development team for analysis and feedback.
- Address and fix any issues found during the testing phase, particularly those affecting faculty data management, scheduling, and report generation.

11. Environmental Needs

The following elements are essential to support the testing efforts at all levels of the Faculty Loading and Scheduling System:

- A computer, laptop, or smartphone that meets the minimum hardware and software requirements to run the system effectively.
- A modern web browser (e.g., Google Chrome, Microsoft Edge) to access the system's web-based interface.
- Reliable internet access to access the system online, synchronize data, and facilitate timely notifications.
- Access to the Test Cases document sheet to track test cases and document results.

12. Staffing and Training Needs

The testing will be conducted by both the development team and key stakeholders, including faculty members and administrative staff. If stakeholders or faculty members are unavailable, at least one dedicated tester should be assigned to continue the testing process. Training sessions may be needed for stakeholders and faculty members to familiarize them with the testing procedures and the functionality of the system.

13. Responsibilities

The Lead Tester will oversee the entire testing process, including organizing schedules and ensuring all testing activities are completed as planned. The entire Project Team will participate in reviewing the system, responding to change requests, and resolving defects identified during development and testing. The Stakeholders (such as faculty members and administrators) will test the system for inconsistencies and defects, provide feedback to the developers, and ensure that the system meets their functional requirements and expectations.

14. Schedule

Time has been allocated within the project test plan for the following testing activities. The specific dates and times for each activity are outlined in the project test plan timeline. The necessary personnel for each task, including the test team, development team, management, and stakeholders, are identified in the project timeline and plan. Coordination of all personnel involved will be managed by the project manager in collaboration with the development and test team leaders.

- Review of System Requirements: Conducted by team members to ensure that all requirements related to faculty data management, scheduling, and reporting are well-understood.
- Development of Test Plan: Preparation of the test plan and test cases, with time allocated for each testing activity, including unit testing, integration testing, and user acceptance testing (UAT).

15. Planning Risks and Contingencies

During the testing process, testing may be halted if an unexpected issue arises that requires immediate fixing or if essential staff are unavailable. In the absence of a formal risk management plan, such challenges will be managed through proactive communication and agile decision-making, with the project team collaborating to identify potential risks as they emerge, such as delays in integrating with external systems (e.g., the Human Resource Information System) or issues with data accuracy. The team will assess the impact of these risks on the testing schedule, prioritize fixing critical bugs affecting core functionalities like faculty scheduling and report generation, and implement contingency measures to minimize delays, such as reallocating tasks to available team members or adjusting the timeline for less critical features. In cases where stakeholder or faculty member availability poses a challenge, the team will promptly adapt and communicate revised timelines for reviews and acceptance testing, ensuring the testing process remains flexible while maintaining overall project timelines and deliverables.

16. Approvals

Project Manager | Developer	Emmanuel Martinez
Technical Lead | Developer	Adrian Naoe
Quality Assurance | Developer	Kyla Rica Malaluan
Technical Writer | Developer	Via Clariz Rasquero

17. Glossary

Term/Acronym	Definition
PUPT FLSS	Polytechnic University of the Philippines Taguig Faculty Loading and Scheduling System – A web-based application designed to manage faculty scheduling, workload assignments, and room allocation.
Agile SDLC	Agile Software Development Life Cycle – A methodology that emphasizes iterative development, collaboration, and flexibility, used

	for managing the system's development and testing processes.
HRIS	Human Resource Information System – An external system integrated with the FLSS to manage faculty data, including personal information, employment status, and qualifications.
Faculty Loading	The process of assigning courses, teaching loads, and schedules to faculty members based on their availability, preferences, and institutional requirements.
Scheduling Conflicts	Situations where there are overlapping or conflicting schedules for faculty, rooms, or courses that need to be resolved to maintain the system's integrity.
Test Plan	A document that outlines the strategy, objectives, resources, schedule, and scope of testing activities for the system to ensure that all functionalities are thoroughly evaluated.
User Acceptance Testing (UAT)	A phase in the testing process where end-users, such as faculty members and administrators, test the system to verify that it meets their requirements and expectations.
Daily Scrum	A short daily meeting used in Agile methodologies to discuss progress, identify obstacles, and plan daily activities.
Release Readiness	A stage in the development process that ensures the system is fully tested, all critical issues are resolved, and the application is prepared for deployment.
Impact Assessment	The evaluation of the system's performance post-deployment to determine its effectiveness, identify any issues, and plan future enhancements.

Test Case	A set of conditions or variables used by testers to determine whether the system behaves as expected.
Contingency Measures	Actions planned and executed to mitigate potential risks or delays in the project schedule.