

Philist Test Plan

Document Change History

Version	Date	Contributor	Description
v1.0	01/15/2024	Dominique Gonzaga	Initial draft of the test plan created, including intro, objectives, and approach
v1.1	01/22/2024	Dominique Gonzaga	Added test scope (in-scope/out-of-scope), project roles, and estimated schedule
v1.2	01/29/2024	Dominique Gonzaga	Included glossary of terms and acronyms; updated roles and assumptions
v1.3	02/05/2024	Dominique Gonzaga	Integrated revised QA timeline and detailed task breakdown based on updated plan
v1.4	02/12/2024	Dominique Gonzaga	Refined test deliverables, tools section, and clarified testing phases

Contents

1. Introduction	3
1.1 Purpose	3
1.2 Background	3
1.3 Scope	3
In Scope:	3
Out of Scope:	3
1.4 Project Identification	3
1.5 Quality Objectives	4
Primary Objective	4
Secondary Objective	4
1.6 Assumptions for Test Execution	4
1.7 Constraints	4
2. Test Approach	4
2.1 Testing Stages	4
2.2 Testing Types	4
2.3 Tasks, Estimations, and Schedule	5
2.4 Bug Life Cycle	5
2.5 Priority Levels	5
3. Test Deliverables	5
4. Environmental Needs	6
4.1 Hardware	6
4.2 Software	6
4.3 Tools	6
5. Roles & Responsibilities	6
Test Engineer	6
Developers	6
Project Manager	7
6. Dependencies & Risks	7
7. Schedule & Milestones	7
8. Approvals	8
9. Terms & Acronyms	8

1. Introduction

1.1 Purpose

To define the testing approach, deliverables, and schedule for Philist, ensuring the application functions as intended and meets the needs of students, teachers, and administrators.

1.2 Background

Philist is an integrated platform designed to streamline academic workflows in educational institutions. It supports students, teachers, and administrators in managing academic tasks such as scheduling, grading, communication, and performance tracking.

1.3 Scope

In Scope:

- Core functionalities: user authentication, class management, enrollment, administrative workflows
- Validation of frontend and backend components
- Cross-browser and cross-device compatibility testing
- Performance testing under typical load conditions

Out of Scope:

- Integration with external APIs beyond core scope
- Advanced security/penetration testing
- Load testing under extreme conditions
- Testing on tablets or unsupported devices
- Browsers other than Chrome and Safari

1.4 Project Identification

Role	Name(s)
Project Name	Philist
Project Managers	Sam Nam, Julie Requiso
Test Engineer	Dominique Gonzaga
Developers	Ronnel Suan, Olliver Jules Morales, Mikko Dacasin, Gael Murillo, Laurence Co
UI/UX Designer	Arcel Vocales

1.5 Quality Objectives

Primary Objective

Ensure Philist meets all functional requirements and delivers a seamless user experience across supported platforms.

Secondary Objective

Address security concerns and assess performance under expected loads to ensure data integrity, responsiveness, and usability.

1.6 Assumptions for Test Execution

- Test environments will mirror production
- Test data will reflect real-world usage
- QA team will collaborate actively with developers and PMs

1.7 Constraints

- Fixed project timelines
 - Limited QA resources
 - Dependency on infrastructure or third-party systems
-

2. Test Approach

2.1 Testing Stages

- Functional Testing
- GUI Testing
- Usability Testing
- Regression Testing
- Smoke Testing
- Sanity Testing

2.2 Testing Types

- Functional Testing
- Usability Testing

2.3 Tasks, Estimations, and Schedule

Task	Member	Estimated Hours	Timeline
Test Planning & Strategy	Test Engineer	40 hours	Jan 15 – Feb 2
Test Case Design & Documentation	Test Engineer	50 hours	Feb 5 – Feb 16
Test Execution – Cycle 1	Test Engineer	60 hours	Feb 19 – Mar 15
Bug Logging & Collaboration	Test Engineer	20 hours	Mar 18 – Mar 29
Test Execution – Cycle 2	Test Engineer	50 hours	Apr 1 – Apr 26
Final Retesting & QA Sign-off	Test Engineer	30 hours	Apr 29 – May 10
Test Report & QA Closure	Test Engineer	20 hours	May 13 – May 31
Buffer / Presentation Week	Test Engineer	10 hours	Jun 3 – Jun 7
Total		280 hours	

2.4 Bug Life Cycle

New → Assigned → In Progress → Retest → Closed

2.5 Priority Levels

Priority	Description
Highest	Blocks testing or product usage completely
High	Serious issue, impacts functionality
Medium	Noticeable, but non-blocking
Low	Minor issue or cosmetic bug
Lowest	Trivial, little or no impact

3. Test Deliverables

- Test Plan
 - Test Cases
 - Test Reports
 - Defect Logs
-

4. Environmental Needs

4.1 Hardware

- Desktops/laptops for admin testing
- Mobile devices for student/teacher testing

4.2 Software

Device	OS	Browser
Desktop	Windows	Chrome
	macOS	Safari
Mobile	Android	Chrome
	iOS	Safari

4.3 Tools

Process	Tool
Test Case Creation	Google Sheets
Test Tracking	Google Sheets
Test Execution	Manual
Test Management	Google Sheets
Defect Management	Jira
API Testing	Postman

5. Roles & Responsibilities

Test Engineer

- Write test plans, design test cases, execute tests, and log bugs

Developers

- Implement features and resolve reported defects

Project Manager

- Manage timelines, ensure coordination, and approve deliverables
-

6. Dependencies & Risks

- Delayed feature delivery
 - Limited test coverage due to time constraints
 - Environment or infrastructure issues
-

7. Schedule & Milestones

Phase	Dates	Notes
Test Planning	Jan 22 – Feb 9, 2024	Review requirements, define scope, create test strategy and test plan
Test Case Design & Prep	Feb 12 – Feb 23, 2024	Write and organize test cases, prep test data, verify environments
Test Execution – Cycle 1	Feb 26 – Mar 22, 2024	Functional testing, UI testing, logging high-priority bugs
Defect Resolution – Cycle 1	Mar 25 – Apr 5, 2024	Work with devs, retest fixed issues
Test Execution – Cycle 2	Apr 8 – May 3, 2024	Regression testing, exploratory testing, edge cases
Final Fixes & Retesting	May 6 – May 17, 2024	Final pass, verifying fixes, polish before sign-off
Test Closure & Reporting	May 20 – May 31, 2024	Write final QA report, summarize defect metrics, lessons learned
Buffer / Presentation Week	Jun 3 – Jun 7, 2024	Optional: wrap-up meeting, handover, GitHub polish, demo prep

8. Approvals

This test plan requires approval from the Project Manager.

9. Terms & Acronyms

- **GUI** – Graphical User Interface
- **OS** – Operating System
- **PM** – Project Manager
- **TE** – Test Engineer
- **UI/UX** – User Interface / User Experience