StudyPilot Project Quality Assurance (QA) Plan

Cevdet Onat Cerit – 231101078
Alvin Dora Akıncı – 231101052
Emin Arslan – 231101007
Nehir Tıraş – 231101065
Nehir Aydın – 231101053

StudyPilot Project Quality Assurance (QA) Plan

Document-Specific Task Matrix:

Task	Team Member Responsible
Quality Assurance Strategy	Nehir Aydın, Nehir Tıraş
Quality Factors and Metrics	Cevdet Onat Cerit
Test Plan	Emin Arslan, Alvin Dora Akıncı

1. Quality Assurance Strategy

1.1. Overview

The Quality Assurance (QA) process ensures that StudyPilot meets the expected functionality, usability, and performance standards throughout development. The QA activities include:

- Clarifying and validating requirements during initial stages.
- Performing continuous testing during development cycles.
- Conducting verification after each new feature implementation (e.g., Pomodoro, Spotify, Weather API).
- Ensuring system stability via regression testing after bug fixes or new updates.

1.2. Testing Approaches

The following testing methodologies will be applied:

1. Integration Testing

- Ensures correct interaction between modules such as Task Manager, Timer Module, Analytics Module, and API Services.
- Example: Testing data flow between Weather API service and suggestion generator.

2. Unit Testing

- Individual functions (e.g., prioritization algorithm, timer countdown logic) will be tested for accuracy.
- Example: Testing task sorting function under various deadlines.

3. Usability Testing

- Evaluates whether users can easily navigate dashboards, manage tasks, start Pomodoro sessions, and view analytics.
- Example: Ensuring users can add/edit/delete tasks without confusion.

4. Regression Testing

- Ensures existing features continue functioning correctly after updates.
- Example: After adding a new analytics chart, verifying Pomodoro still works.

5. System Testing

- End-to-end testing of StudyPilot’s workflow — from task entry to timer usage, to Spotify/weather suggestions, to analytics.
- Example: Running a full study session with music and checking if data is reflected in analytics.

1.3. Automation and Manual Testing

System Components to be Tested with Automated Tests:

● **Unit Tests:**
✓ Prioritization algorithm accuracy
✓ Timer countdown and break-cycle correctness
✓ Task CRUD operations
✓ Analytics data calculations

● **Integration Tests:**
✓ Weather API → Suggestion Module
✓ Spotify API → Playlist display
✓ Timer → Analytics Module

● **LocalStorage Operations:**
✓ Data writing and reading consistency
✓ Task persistence between sessions

System Components to be Tested with Manual Tests:

● **Usability Tests:**
✓ Page navigation (Dashboard, Tasks, Pomodoro, Analytics)
✓ Visual clarity and responsiveness (mobile/tablet/desktop)

● **Spotify Integration:**
✓ Authorization popup
✓ Playlist loading and playback availability
✓ Correct playlist categories for focus mode

User Acceptance Tests (UAT):

✓ The system's functionality and its ability to meet user expectations will be manually assessed based on real user experiences.

● **Weather Integration:**
✓ Weather accuracy display
✓ Study suggestion generation

● **Pomodoro User Experience:**
✓ Start, pause, resume, finish session flows
✓ Notification behavior

● **Complex Interaction Scenarios:**
✓ Simultaneously running timer + Spotify + weather update
✓ Switching tasks during active sessions

2. Quality Factors and Metrics

The team has identified four primary quality factors and measurable performance metrics:

Quality Factor	Description	Measurement Metric
Performance	Page loads and task operations should be fast	Dashboard load time < **2 seconds**
Usability	User interface must be intuitive and easy to navigate	User satisfaction score (0–5 scale)
Reliability	Timer and data persistence accuracy	Timer deviation ≤ **1 second**, LocalStorage consistency
Integration Accuracy	Spotify & Weather API must return correct data	Successful API responses (%)

These metrics will be monitored during testing and evaluated at the end of the project.

3. Test Plan

3.1. Test Scenarios

Below are five core StudyPilot test scenarios:

Test Name	Preconditions	Steps	Expected Outcome
Task Creation & Display	Website running	1. Open Tasks page 2. Add a task	New task appears with correct fields; stored in LocalStorage
Pomodoro Timer Start	Website running	1. Click "Start Timer" 2. Let timer run	Timer counts down correctly; session saved in analytics
Spotify Playlist Loading	Spotify API credentials valid	1. Click "Focus Music" 2. Request playlist	Study playlist loads and is playable
Weather-Based Suggestion	Internet active	1. Open Dashboard 2. Weather auto-fetch	Weather displayed; personalized suggestion shown ("Cloudy → Study indoors")
Task Prioritization Algorithm	At least 3 tasks exist	1. Add tasks with different deadlines	Tasks sorted by urgency (nearest deadline first)

3.2. Bug Tracking

● All bugs will be tracked through **GitHub Issues**.
● Each issue will include:
✓ Bug description
✓ Steps to reproduce
✓ Expected vs Actual results
✓ Severity level (Critical, High, Medium, Low)

● Fixes will be addressed by priority.
● After every bug fix, **regression tests** will be performed to ensure other features remain unaffected.