Software Plan

# **1. Project Overview**

Project Title: Student Document Request Tracker

Team Members: Omandam Jasmine, Calledo Martina Monica, Duron Roman Patrick

Date Started: September 30,2025

Expected Completion: December 1 ,2025

Short Description: (What the software is and why it’s being built)

**Student Document Request Tracker** is a software tool designed to streamline and automate the process of requesting, tracking, and managing student documents. It allows students, administrative staff, and educators to submit, review, and monitor the status of document requests, ensuring faster processing and better organization. This software is being built to reduce administrative overhead, minimize errors, and enhance communication between all stakeholders involved in the document handling process. It is especially beneficial in educational institutions where document requests (like transcripts, certificates, and recommendation letters) are frequent and time-sensitive.

# **2. Objectives**

Main Goal: To create an efficient, automated system for tracking and managing student document requests, improving the workflow for both students and administrative staff.

Specific Objectives:

* Provide an easy-to-use platform for students to submit document requests.
* Automate document tracking, approvals, and delivery updates.
* Minimize administrative overhead by centralizing the document request process.
* Improve transparency by providing real-time status updates to all users.

# **3. Scope**

In-Scope Features:

* Document request submission (by students).
* Status tracking (pending, approved, rejected, processed).
* Notifications to students and administrative staff.
* Admin dashboard to monitor and manage requests.
* Search and filter functionality for requests and student data.

Out-of-Scope Features:

* Integration with external document verification systems.
* Support for non-student users (e.g., alumni).
* Mobile app development (for initial phase).
* Advanced data analytics and reporting tools.

# **4. Stakeholders**

**Primary Users:**

* Students (requesting documents).
* Administrative Staff (processing requests and providing updates).

**Other Stakeholders:**

* Faculty (approving academic-related document requests).
* IT Support Team (system maintenance and troubleshooting).

# **5. Requirements (Simplified)**

**Functional Requirements:**

* The system must allow students to create and submit document requests with supporting information.
* The system must allow administrative staff to view, approve, and reject requests.
* Notifications must be sent to relevant parties (students and staff) at each stage of the request.

**Non-Functional Requirements:**

* The system should be able to handle up to 1,000 simultaneous user interactions.  
  The interface should be user-friendly and accessible on various devices (desktop, tablet).
* Data should be encrypted and comply with data protection regulations (e.g., GDPR).

# **6. System Design (High-Level)**

**Architecture / Modules:**

* **User Interface (UI):** For students and staff to interact with the system.
* **Request Management System:** Manages requests and tracks their status.
* **Notification System:** Sends alerts and updates to users.  
  **Admin Dashboard:** For administrative staff to manage, approve, or reject requests.
* **Database:** Stores user data, document requests, and statuses.

**Technologies / Tools:**

* Frontend: Windsurf or Bootstrap
* Backend: Node.js
* Database: MySQL.
* Notification System: Email (SMTP) or SMS via a service.
* Authentication: JWT for secure login.

**Diagram (optional):** [Insert diagram here if needed for system flow]

# **7. Project Timeline**

Major Milestones:  
**Week 1:** Requirements gathering and analysis; Project setup.

**Week 2:** Design UI/UX and define database schema; Start backend development.

**Week 3–4:** Complete backend integration; Implement user interface.

**Week 5:** System testing; Beta release to limited users.

**Week 6:** Final testing, bug fixes, and launch.

or

Gantt Chart / Task List : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **8. Risks & Mitigation**

Possible Risks:

* **Delayed development:** Tight timelines and dependencies may cause delays.
* **User adoption:** Students and staff may be hesitant to adopt a new system.

Mitigation:

* **Delayed development:** Set clear milestones and regular check-ins; allocate buffer time for unexpected issues.
* **User adoption:** Provide training materials, user guides, and support; ensure a user-friendly interface with clear instructions.

# **9. Testing & Quality Plan**

What to Test:

* User interface responsiveness and accessibility.
* Backend functionality, including request tracking and notifications.
* System security, especially user data encryption.
* Performance under load (up to 1,000 concurrent users).

How to Test:

* **Unit Testing:** Test individual components like request submission and status updates.
* **Integration Testing:** Ensure all modules work seamlessly together (UI, backend, and database).
* **User Acceptance Testing (UAT):** Involve real users to ensure the system meets their needs and is easy to use.

# **10. Deliverables**

Expected Outputs:

* A fully functional web application for managing student document requests.
* User documentation and training materials.
* Post-launch support and bug fixes for the first 3 months.