**Software Requirements Specification (SRS)**



**Project Name: Lab Equipment Management System (LEMS)**  
**Date: [18/03/2025]**

**Group Number: [Group 2]**  
**Prepared by: [Black Elite]**  
**Submitted to: GARCIA, ANN JANETH G.**

### **Declaration and Acknowledgment Statement**

We, **Alex Garcia, Chad Domingo, Nathaniel Amper, Vito Gandeza, and Xhaun Lozada**, hereby declare that the work submitted in this document is part of the academic requirements for **SOFTWARE DESIGN LECTURE** under the supervision of **ENGR. ANN JANETH GARCIA** at **DE LA SALLE UNIVERSITY - MANILA**.

We acknowledge that this submission, along with any accompanying materials, may be used by the professor for academic, research, instructional, or administrative purposes. This includes, but is not limited to, reference for future coursework, publication in academic materials, or incorporation into institutional records.

Furthermore, we understand that once submitted, this work becomes part of the university’s academic records and may be retained, reviewed, or shared at the discretion of the professor or university administration in compliance with institutional policies.

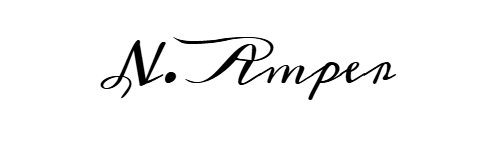
By signing below, we affirm our understanding and agreement to these terms.

Student's Name: Alex Gabriel D. Garcia  
Student ID: 12311022

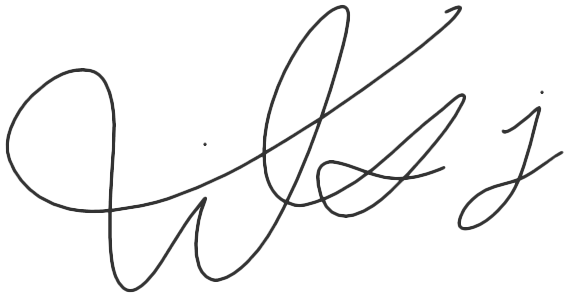
Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student's Name: Chad Wilson A. Domingo  
Student ID: 12325511

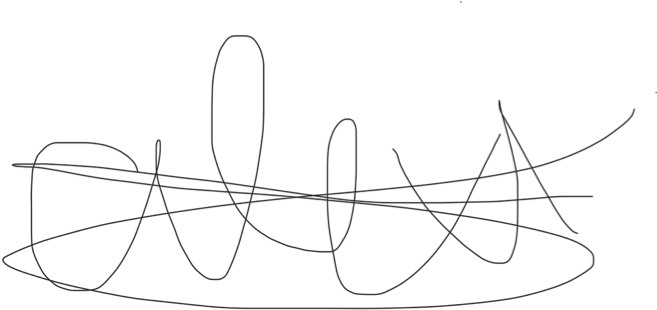
Signature:

Student's Name: Nathaniel N. Amper  
Student ID: 12310085

Signature:

Student's Name: Vito J. Gandeza  
Student ID: 12310972

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student's Name: Xhaun Alexi S. Lozada  
Student ID: 12311480

Signature:



## **1. Introduction**

### 1.1 Purpose

The purpose of this document is to outline the functional and non-functional requirements for the Lab Equipment Management System (LEMS). This system aims to streamline the management of lab equipment in academic and research settings by providing an integrated platform for tracking, scheduling, and maintaining lab equipment. The intended users of this system include lab managers, researchers, students, and administrative staff. The benefits of LEMS include improved efficiency in equipment usage, reduced administrative overhead, and enhanced equipment maintenance practices.

### 1.2 Document Conventions

In this document, the following conventions are used:

* FR-XX refers to Functional Requirements.
* NF-XX refers to Non-Functional Requirements.

### 1.3 Intended Audience and Reading Suggestions

This document is intended for software developers, system architects, quality assurance teams, lab managers, and stakeholders involved in the development and implementation of the Lab Equipment Management System. It is recommended that readers familiarize themselves with the objectives and key functions of the system before delving into the detailed requirements.

### 1.4 Project Scope

LEMS is a web-based platform designed to manage lab equipment effectively. It includes functionalities such as intelligent inventory management, dynamic reservation systems, predictive maintenance tracking, advanced reporting, and granular access control. The system will integrate with existing university login systems and will be accessible via both mobile devices and computers. Constraints include ensuring compliance with data protection regulations such as GDPR and ISO standards, and the system must be scalable to accommodate varying lab sizes and equipment inventories.

### 1.5 References

* IEEE 830-1998 Standard for SRS
* Database Schema Design Guide
* API Documentation



## **2. Overall Description**

### 2.1 Product Perspective

LEMS is designed to integrate seamlessly with existing university infrastructure, including lab facilities and administrative systems. It will function as a standalone web application but will also interface with other systems such as university login systems and third-party vendor services for maintenance and repair.

### 2.2 Product Functions

The Lab Equipment Management System (LEMS) will offer the following key functionalities:

* **Intelligent Inventory Management:** This includes QR code/barcode generation for equipment tagging, location tagging for equipment placement, and a condition scoring system.
* **Dynamic Reservation System:** Features conflict-free scheduling, waitlist automation with priority tiers, and mobile notifications for reservation confirmations and pre-use checklists.
* **Predictive Maintenance Tracker:** Integration with manufacturer APIs for calibration guidelines, third-party vendor coordination for repairs, and maintenance cost forecasting.
* **Advanced Reporting Module:** Customizable dashboards for utilization rates, idle time analytics, user behavior trends, compliance-ready audit trails, and predictive analytics for equipment lifespan.
* **Granular Access Control:** Time-bound access permissions, two-factor authentication for high-value equipment, and audit trails for user actions.

### 2.3 User Characteristics

The primary users of the Lab Equipment Management System (LEMS) will be lab managers, researchers, students, and administrative staff. Users are expected to have basic computer literacy and familiarity with web-based applications. Lab managers and administrative staff will have higher proficiency in system administration and data management, while students and researchers will primarily interact with the reservation and usage functionalities.

### 2.4 Constraints

* The system must be compatible with major web browsers such as Google Chrome, Firefox, and Edge.
* It must support a maximum of 1,000 concurrent users.
* Data storage and processing must comply with GDPR and ISO standards for data protection and security.
* The system must be designed to be scalable to accommodate varying lab sizes and equipment inventories.

### 2.5 Assumptions and Dependencies

* The system assumes a stable internet connection for real-time updates and notifications.
* It relies on a relational database management system (RDBMS) for storing user data, equipment information, and reservation records.
* The system assumes the availability of third-party APIs for integration with manufacturer guidelines and vendor services.



## **3. Specific Requirements**

### 3.1 External Interfaces

This section describes the inputs and outputs of the ELM Prestige system, detailing their purpose, format, and interactions with other components.

#### 3.1.1 User Interfaces

* Name: Customer ELM Interface
  + This allows users to browse products, place orders, and manage their accounts
  + ELM Prestige shows a clean, minimalist web design with sections for product categories, cart, checkout, and order tracking.
  + The data format involves using HTML, CSS, JavaScript for front-end structure and interactions
  + Clickable buttons for navigation through cursor, forming of inputs for purchasing, and authentication fields for login.
* Name: ELM Admin Dashboard
  + This enables business administrators to monitor inventory, manage orders, and update product listings.
  + A structured dashboard with tabs for inventory, orders, analytics, and inquiries from customers
  + The system displays in tables and charts through JavaScript and backend data retrieval.
  + This form inputs for data entry, dropdowns for selection, and buttons for management actions done.

#### 3.1.2 Hardware Interfaces

* Name: ELM Web Hosting Server
  + This stores website files, database, and handles user requests.
  + This is hosted on cloud-based or dedicated servers.
  + This could be stored through gigabytes, along with the bandwidth in megabits per second.
* Name: Receipt Printer
  + This prints transaction receipts for physical store orders
  + This uses USB or network connection using standard print drivers.

#### 3.1.3 Software Interfaces

* Name: PostgreSQL Database
  + This manages and stores customer data, orders, and inventory.
  + SQL tables with structured product, user, and transaction records.
  + SQL queries for retrieval, insertion, and updates.
* Name: Payment Gateways (GCash, Maya, Stripe)
  + The system processes online payments securely.
  + The data format involves API requests and responses in JSON format.
  + This uses API requests using tracking numbers to fetch delivery details

.

#### 3.1.4 Communications Interfaces

* Name:HTTPS Protocol
  + ELM securely transmits user and transaction data between the client and server.
  + This involves encrypted HTTP requests (GET, POST, PUT, DELETE)
* Name: SMTP Email Notifications
  + This sends order confirmation, receipts, and promotional emails.
  + The email text formatted is in HTML or plain text.
  + This uses SMTP protocol to relay emails through a configured mail server.

#### 3.1.5 End Messages

* Order Confirmation could show “Your order has been successfully placed. A confirmation email has been sent.”
* The payment success indicator could show “Payment received. Your order is now being processed.”
* The out of stock alert would be “This product is currently unavailable.”
* Shipping Update could show “Your order has been shipped. Tracking number: [xxxxxx].”



### 3.2 External Interface Requirements

#### 3.2.1 User Interfaces

* A web-based UI with a responsive design to ensure accessibility on both desktop and mobile devices.
* Dashboards for lab managers to oversee equipment status, reservations, and maintenance schedules.

#### 3.2.2 Hardware Interfaces

* The system should be compatible with standard laptops, tablets, and desktops.
* Support for barcode scanners and QR code readers for equipment check-in/check-out processes.

#### 3.2.3 Software Interfaces

* REST API for communication with third-party learning platforms and vendor services.
* Integration with existing university login systems for seamless user authentication.

#### 3.2.4 Communication Interfaces

* Secure HTTP (HTTPS) for web-based interactions to ensure data security and privacy.



### 3.3 System Features



### 3.4 Non-Functional Requirements

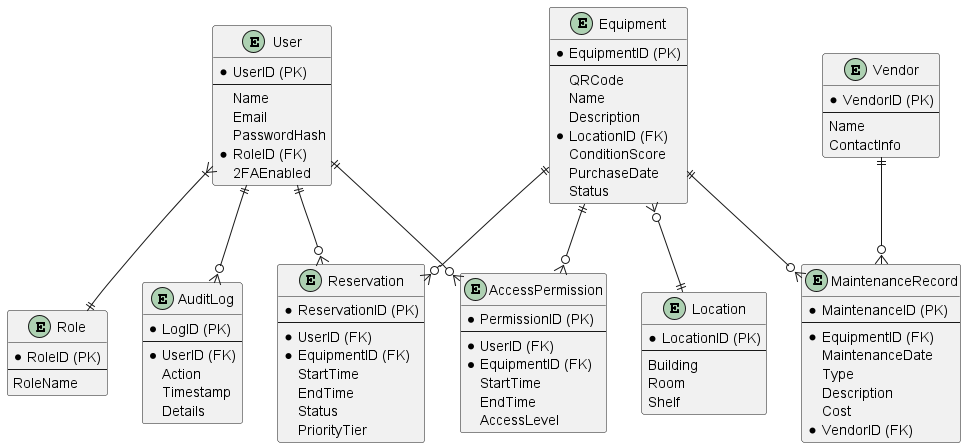


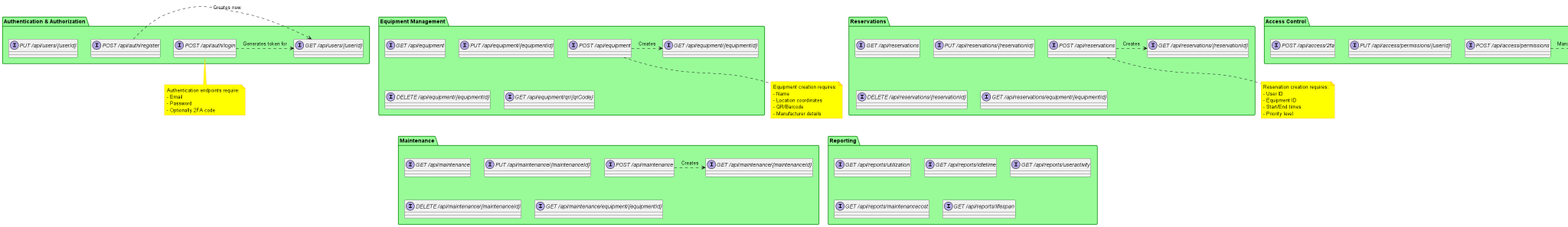
## **4. Appendices**

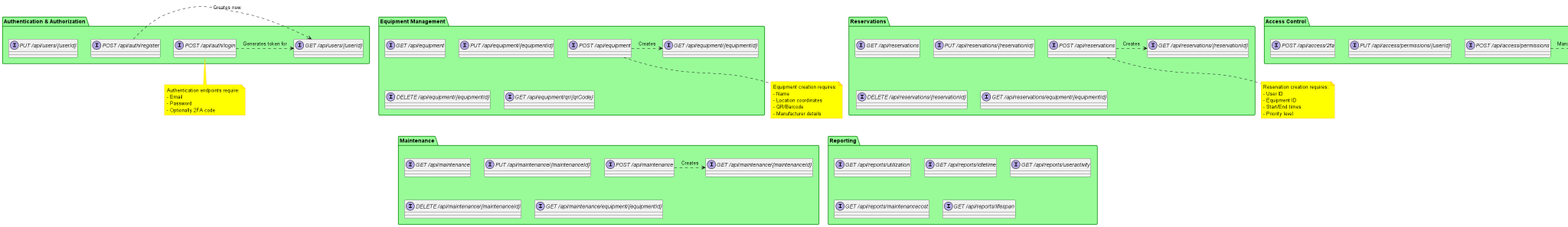
Include additional details, formulas, tables, pictures or references. Maximum of two pictures/tables only per page.

*Example:*

* "Appendix A: Database Schema"



* "Appendix B: REST API Endpoints"





## **5. Index**

* Functional Requirements (FR)
* Non-Functional Requirements (NF)
* User Authentication and Authorization
* Intelligent Inventory Management
* Dynamic Reservation System
* Predictive Maintenance Tracker
* Advanced Reporting Module
* Granular Access Control
* Performance Requirements
* Security Requirements
* Availability and Reliability
* Maintainability and Supportability
* Usability Requirements