ElectroStock

Software Requirements Specification

0.0

23/03/23

Tomas Mariano Güell
Lead Software Developer

Prepared for:
Teacher: Ing. Fratin Juan.
Extracted from: WSU-TC CptS 322—Software Engineering Principles I
Integrants:
Canepa Ignacio
Cimatti Bautista
Genari Tomas
Nacho
Mariano Dubois
Giayetto Francosco
Morales Franco Demaria

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

Table of Contents

1. Introduction

The Electronics Stock Management System is a software solution designed for a high school's Electronics specialty program to efficiently manage the inventory of electronic components, tools, and equipment. The system provides teachers, students, and staff with a user-friendly interface for tracking inventory levels, submitting loan requests, and monitoring loan history. For this purpose the team made different meetings with teachers and area chiefs, collecting information and talking to people specialized in the development area, aiming to comprehend exactly what and which are the requirements that our client demands.

1.1 Purpose

The purpose of the software project is to develop a comprehensive stock management system tailored for the Electronics specialty , which includes a master/student loan system. This system aims to streamline and automate the management of electronic components, tools, and equipment in the school's inventory, as well as track the loan history of items borrowed by students from *teachers. In the brief future the team will probably find and propose new ways to improve the efficiency and the friendliness of the project.*

1.2 Scope

The system will enable the school to track the inventory of electronic components, as well as provide a user-friendly interface for teachers, students, and staff to manage stock levels, monitor usage, and request restocking when necessary.

The key features of the system include:

Inventory management: The system will enable the school to track the inventory of electronic components, tools, and equipment and monitor stock levels.

Stock request and restocking: The system will provide a platform for teachers to request restocking of items when stock levels are running low.

Budget Calculation: It is a crucial aspect of the project. It involves the determination of a budget based on the stock losses that occurred during the year for various reasons. This budget will be calculated using a specialized function, which will also enable the area managers to estimate the renewal cost of stock. This estimation will be based on the current peso/dollar currency exchange rate. The budget calculation and renewal estimation process are important for ensuring that the project stays within financial constraints and that stock availability is maintained at all times.

Master/student loan system: The system will allow teachers to loan equipment and components to students for use in classroom projects or at home. It will also track the loan history of items borrowed by students from teachers and provide alerts for overdue or lost items.

The project team will be responsible for the design, development, testing, and deployment of the software. The project will follow a structured development methodology and will include regular meetings with the school administration to ensure the system meets their needs and requirements..

1.3 References

Excel document with user stories.

IceScrum Project

Github repo with project progress and documentation.

1.4 Overview

The next part of this SRS will give a General description of the product and his respective requirements, also it will give some use cases that can function as examples of his utility.

2. General Description

This project aims to improve the efficiency with which students and teachers manage both the stock and its tracking. Therefore, as a team, we have projected a series of functionalities based on the preparation of multiple user stories and tasks that later served as a guide to estimate the complexity that the team would encounter when starting.

To do this, these stories were previously classified according to their functionality as follows:

Database

Loan System
Accounts System
Stock Control
Budget
E-commerce
Frontend Design

In addition to the classification of user stories, we also considered non-functional requirements, such as performance, reliability, and security. We identified the need for the system to be fast, scalable, and able to handle large amounts of data. We also considered the importance of data integrity, backup and recovery mechanisms, and access control measures.

To ensure that the system meets these requirements, we plan to use modern technologies and best practices in software development. We will also conduct rigorous testing and quality assurance processes throughout the development lifecycle to identify and address any issues that may arise.

Overall, our goal is to create a system that is user-friendly, efficient, and reliable, and that meets the needs of both students and teachers in managing electronic components in a high school setting.

2.4 General Constraints

Integration with existing systems: The developers need to ensure that the new system integrates seamlessly with the school's existing systems to prevent any disruptions or data inconsistencies.

Hardware compatibility: The system needs to be compatible with the hardware used by the school to manage stock and loan equipment. We need to ensure that the system can run smoothly on different devices and operating systems.

Security and privacy: The system must adhere to the school's data security and privacy policies, which may include access control, data encryption, and user authentication protocols.

Time constraints: We must deliver the system within a specified timeline, which was already delivered by the school academics.

User interface design: The system must be designed with a user-friendly interface that is easy to navigate, intuitive, and efficient for all users, including teachers, students, and staff.

3. Specific Requirements

Inventory and component loans.

- *That the inventory is affected when lending/returning components.*
- *That the student can access their loan sheet and see what they owe.*
- *That the management system is as practical as writing on a sheet of paper. It cannot be cumbersome.*
- *To have a box indicating the teacher who lent and/or received the component. Crossout the borrowed component, do not delete it from the sheet.*
- *Maintain the loan sheet for teachers.*
- *Being able to access loans from the previous year.*
- *Having cloud support for inventory so that teachers can access and see what we have.*

Budget.

- *Being able to create the budget based on the inventory.*
- *That the budget is created with those inventory items whose remaining quantity is 45% or more.*

3.1 External Interface Requirements

Security: The system should have security features such as authentication and access control to protect against unauthorized access to data. The system should also have backup and recovery mechanisms in place to prevent data loss in case of system failure.

Reporting: The system should allow for the generation of reports and data analysis to provide insights into inventory levels, loan history, and other relevant metrics. Reports should be customizable and exportable to standard formats such as PDF or Excel.

3.1.1 User Interfaces

The system requires user accounts to be set up for each teacher, student, and staff member who will use the system. User account information, including login credentials and personal information, must be stored securely and accessible only to authorized personnel. The students' accounts will be automatically created adding their data in the models section of the project, using for this task an already prepared excel provided by the school directives.

3.1.2 Hardware Interfaces

The system requires a computer or mobile device with internet connectivity to access the software interface. The hardware should be capable of running modern web browsers and have adequate storage to save data.

3.1.3 Software Interfaces

The system requires a modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge to access the software interface. The software must be compatible with the latest versions of these web browsers.

3.1.4 Communications Interfaces

The system should allow for the import and export of data in standard formats such as CSV, Excel, or JSON. This will enable data to be easily transferred between the system and other software applications.

3.2 Functional Requirements

Inventory management: The system should allow the user to add, edit, and delete products from the inventory. Each product should have a unique identifier, name, description, stock level, location, and other relevant information.

Stock level monitoring: The system should monitor stock levels for each product and notify users when stock levels fall below a specified threshold. Users should be able to view the current stock level of a product at any time.Also the teachers should be able to calculate the yearly renewal of the stock based on the dollar prices in that actual moment.

Product categorization: The system should allow the user to categorize products by type, brand, or other relevant categories to make it easier to find products and generate reports.

Loan management: The system should allow the user to loan products to students and staff and keep track of loan history. The user should be able to see which products have been loaned, to whom, and when they are due to be returned.

Loan request: The system should allow the student to request a loan of a product, and the teacher should be able to approve or reject the request.

Checkout: The system should allow the user to add products to a checkout chart and process the checkout. The system should calculate the total price of the products and deduct them from the inventory stock level.

Reporting: The system should allow the user to generate reports on inventory levels, loan history, checkout history, and other relevant metrics. Reports should be customizable and exportable to standard formats such as PDF or Excel.

Search: The system should allow the user to search for products by name, description, or other relevant criteria to find specific products quickly.

User roles: The system should have different user roles such as administrator, teacher, and student, with different levels of access to different features.

Notifications: The system should send notifications to users when a loan request is approved, when a product is due for return, or when stock levels fall below a specified threshold.

3.5 Non-Functional Requirements

The non-functional requirements include usability, performance, reliability, security, scalability, accessibility, interoperability, maintainability, availability, and compliance. These requirements ensure that the system provides a high-quality user experience, meets industry standards, and is secure, reliable, and scalable to meet the needs of a growing user base over time.

3.5.1 Performance

The system should respond quickly to user requests, provide real-time information, and handle a considerable volume of transactions without delays or errors.

3.5.2 Reliability

The system should be reliable, available 24/7(or be available at least in schooltime), and have a minimum downtime for maintenance and updates.

3.5.3 Availability

The system should have a high level of availability, with a minimum uptime of 99%, and a backup and disaster recovery plan in place.

3.5.4 Security

The system should be secure, with appropriate measures in place to protect sensitive data and prevent unauthorized access, data loss, or data corruption.

3.5.5 Maintainability

The system should be maintainable, with well-documented code, easy-to-maintain software architecture, and an efficient update process.

3.5.6 Portability

The system should be interoperable with other systems, applications, and devices, using open standards and APIs to facilitate integration.

3.7 Logical Database Requirements

Inventory Management: The system must have a database that can store information about the products available in the inventory, such as product name, description, quantity, price, and availability.

Loan Management: The system must have a database that can store information about the loan transactions, such as loan date, return date, borrower name, and borrower ID.

User Management: The system must have a database that can store information about users, such as their name, contact information, and user ID.

Authorization Management: The system must have a database that can store information about user roles, access rights, and privileges.

Transaction Management: The system must have a database that can store information about transactions, such as sales, loans, and returns.

Reporting: The system must have a database that can store information required to generate reports, such as transaction history, inventory levels, and loan status.

Backup and Recovery: The system must have a database backup and recovery mechanism to ensure that data is safe and available in case of a system failure or disaster.

Performance: The database must be designed for optimal performance, including query optimization, indexing, and caching, to ensure that the system operates efficiently.

3.8 Other Requirements

Mobile Compatibility: The system may need to be compatible with mobile devices, to allow users to access the system on the go.

Multi-language Support: The system may need to support multiple languages, to accommodate users who speak different languages.Also we are tracing a plan to include disabled people so we can hoard all the user experience possibilities.

Customization: The system may need to provide customization options, such as customizable fields or reports, to meet the specific needs of different users or organizations.In this case we might be able to let other academics modify the system, in case something goes wrong in the time we can't do software maintenance.