EVSU STORE Requirements Gathering

1. Introduction

# 1.1 Purpose

EVSU STORE is a techstore centered on the sale of hardware components. This document describes the Software Requirement Specification (SRS) for the administration and recommendation software. It will outline all of the agreed upon features of the software.

# 1.2 Product Scope

This SRS is addressed to the store manager, to begin with the development of the object-oriented application called EVSU\_Store, whose main objective is to manage the different store’s administrative processes (Inventory, Sales, Purchase History and Customer Information) and the provision of recommendations regarding suitable vendors for the procurement of hardware components, as well as the organization of said components according to their compatibility, in preparation for the assembly of a complete personal computer.

# 1.3 Definitions, acronyms, and abbreviations

There are a variety of terms used in this SRS relating to the software being created.

* Hardware: Physical components of a computer system.
* PC: Personal Computer, a computer system designed for individual use.
* API: Application programming interface is a set of definitions and protocols used to design and integrate application software.
* UI: User interface

# 1.4 References

* IEEE 830-1998, IEEE Recommended Practice for Software Requirements Specifications. IEEE, 1998.
* Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
* Sommerville, I. Software Engineering. 10th ed., Pearson, 2015.
* Open Web Application Security Project (OWASP). (2023). OWASP Top 10. Recuperado el 11 de mayo de 2023, de <https://owasp.org/www-project-top-ten/>

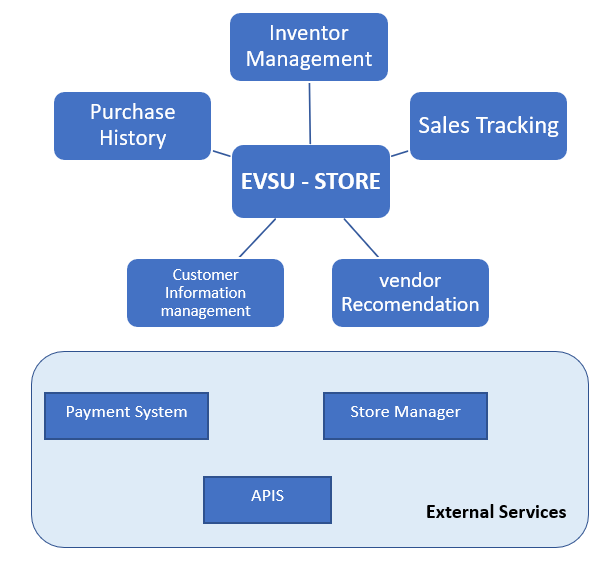
# 1.5 Overview

This SRS is organized as follows: Section 2 provides an overview of the EVSU\_Store software, including the product perspective, customer desired features, limitations, and dependencies. Section 3 delves into specific software requirements such as external interfaces, features, performance requirements, database requirements, design constraints, and software system attributes. Section 4 provides supporting information and appendices.

2. Overall description

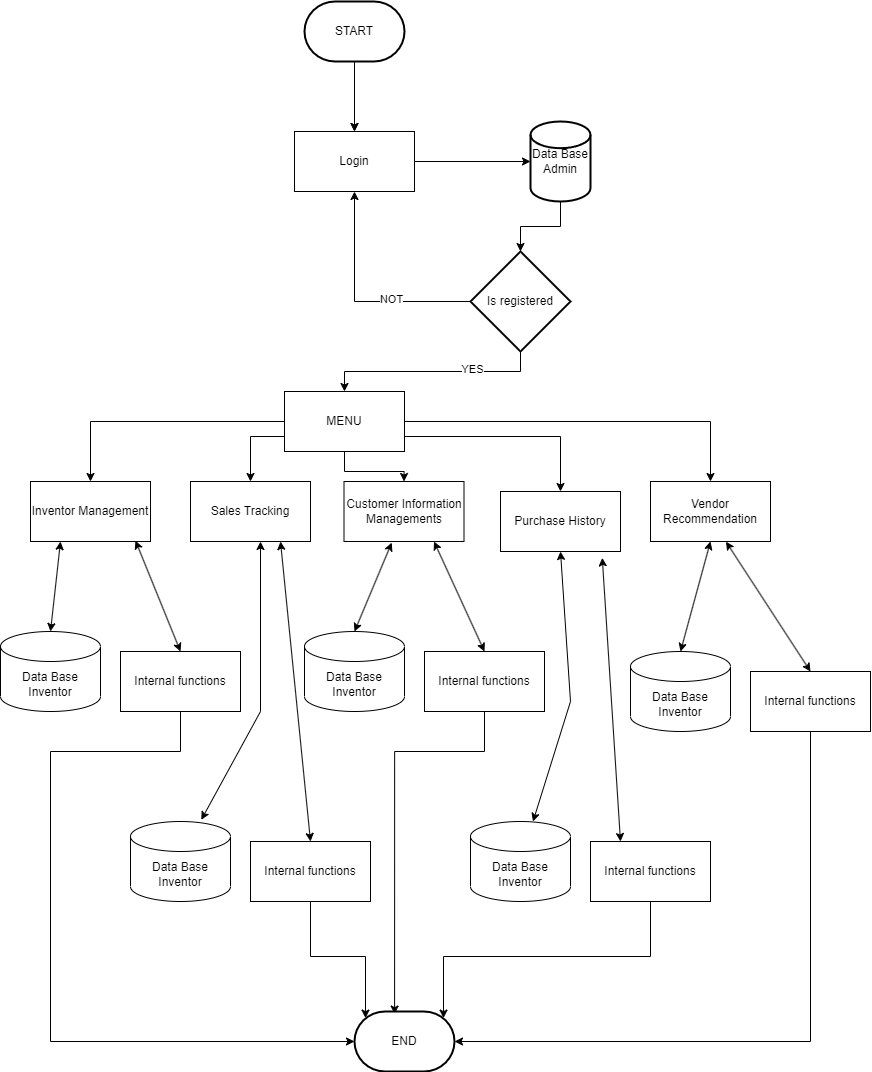
# 2.1 Product perspective

EVSU\_Store is a standalone software application to be developed to manage the operations of the STOEVSU RE store. Its goal is to improve store efficiency and productivity by automating key administrative tasks such as inventory management, sales, purchase history, and customer information.



# 2.2 Product functions

The main functions of the software include: inventory management, sales tracking, purchase history recording, customer information management and recommendation of suitable suppliers for the acquisition of hardware components. You will also organize hardware components by compatibility to make it easy to assemble a complete personal computer.



# 2.3 User characteristics

The personnel of the EVSU\_Store are mainly the store manager and the sales staff. They have basic computer skills and knowledge of the hardware components sold in the store.

# 2.4 Constraints

* The system must be compatible with Windows and Linux operating systems.
* The system must be developed in an object-oriented programming language, in this case java.
* The database must be non-MySQL, so other alternatives such as MongoDB must be used.
* The system must have a certain degree of security, sensitive data, such as the user's personal information and payment details, must be encrypted during transmission and storage.
* Also implement strong authentication mechanisms (such as strong passwords, two-factor authentication) to verify the identity of users before allowing them access.

# 2.5 Assumptions and dependencies

* It is assumed that all personnel will have access to a computer with the necessary specifications to run the software.
* The software depends on the availability and proper functioning of the MongoDB database. Although initially only text files will be handled to simulate a database that allows us to obtain information from the console.

# 2.6 Apportioning of requirements

The software functionalities will be divided into modules to facilitate its development and maintenance. Modules will include: inventory management, sales tracking, purchase history recording, customer information management, and supplier recommendation. Payment system, online store and maybe the use of API to generate personalized text automatically.

3. Specific requirements

# 3.1 External interfaces

* Api: It is planned to use text generation APIs to create reviews of the products sold in the store.
* User Interface: Intuitive and easy to use design.
* Database interface: interaction with the MongoDB database to store and retrieve information.

# 3.2 Functions

* Inventory management: add, remove and update products.
* Sales Tracking: record sales and generate reports.
* Customer Information Management: Store and update customer information.
* Vendor Recommendation: Suggest vendors based on component compatibility and quality.
* Algorithm to generate compatibility according to the components you buy.
* Implement a payment system.
* Implement a text generator to automate the creation of reviews and thus improve the business.
* You should handle different classes that contain various unique objects, since managing a store requires you to specify which item is for sale.
* Different types of data are formed since the project tightened it in the same way, the encapsulation of functions will be essential to have a better control of the code.

# 3.3 Performance requirements

The system must be able to handle multiple clients simultaneously without degrading performance.

# 3.4 Queries to the database must be efficient and fast.

Indexes should be used on the relevant columns to speed up database queries, and Using the cached object model will be used, so by caching the database objects, you can reduce access times and improve application efficiency.

Queries can also be optimized for the number of fields that are requested, preventing necessary queries and ensuring that the proper filter and sort clauses are used.

# 3.5 Logical database requirements

The data model to be used for the database will be specified. Specifying the data types, restrictions and validations that will be applied to the fields of the database, and the tables and relationships that will be used in the database will be defined.

# 3.6 Design constraints

The system will use design patterns to ensure code quality and maintainability. The use of frameworks and libraries will be applied to speed up development and ensure compatibility and interoperability with other systems. Regarding scalability and performance, the software must be able to handle large volumes of data or a high workload.

Regarding security, it will be handled by specific restrictions in terms of authentication, authorization, encryption and session management.

# 3.7 Software system attribute

The system will include aspects such as portability, usability, scalability, security, performance, efficiency, integration, reliability, among others. In addition, system attributes will be specific and measurable so that the success of the project can be objectively evaluated.

# 3.8 Organizing the specific requirements

This Software will focus the organization of specific requirements as follows:

Identify the main stakeholders and their needs, that is, understand the needs of the client, the end user and any other person of interest to this system. We will group the requirements by functionality, this can help to ensure that all the requirements are covered and can also make it easier to manage and implement the software.

Requirements will be prioritized, having a clear idea of ​​which requirements are most important to the client and end users. With the verification and validation of the requirements through verification and validation tests to ensure that the established requirements are met.

4. Supporting information

Detailed support information will be included to ensure that users have access to necessary documentation and help, such as user manuals, online tutorials, how-to videos, FAQs, and online support. It will also specify the contact methods available to users if they need additional assistance, such as email addresses, telephone numbers, opening hours, etc. And information about available support levels will be provided.

# 4.1 Table of contents and index

[**1.1 Purpose**](#_am4j7qbbfs4y) **1**

[**1.2 Product Scope**](#_tjgct9euajvs) **1**

[**1.3 Definitions, acronyms, and abbreviations**](#_2ykn6vbl08cb) **1**

[**1.4 References**](#_iam8cxlosmj8) **1**

[**1.5 Overview**](#_5mvarkmfkew8) **2**

[**2.1 Product perspective**](#_6e3yskt1mdz2) **2**

[**2.2 Product functions**](#_2wks1rx6upyu) **3**

[**2.3 User characteristics**](#_akvkwzqqs1fz) **4**

[**2.4 Constraints**](#_oizhgo3fjmmc) **5**

[**2.5 Assumptions and dependencies**](#_4as87maethk7) **5**

[**2.6 Apportioning of requirements**](#_af5vka4zbpoi) **5**

[**3.1 External interfaces**](#_fa8b3kwevb5a) **6**

[**3.2 Functions**](#_zf8mssr9k44t) **6**

[**3.3 Performance requirements**](#_p80v4hpcr8kl) **6**

[**3.4 Queries to the database must be efficient and fast.**](#_sdmxujoeh3tw) **7**

[**3.5 Logical database requirements**](#_rse8qngm3bkx) **7**

[**3.6 Design constraints**](#_pem80sfdkd0f) **7**

[**3.7 Software system attribute**](#_g3f06kczgqha) **7**

[**3.8 Organizing the specific requirements**](#_x64pg2m4oa4t) **7**

[**4.1 Table of contents and index**](#_ftvtkwlp0587) **8**

[**4.2 Appendixes**](#_nzlhg2jsrjyi) **9**

# 4.2 Appendixes

**1. What device do you want to realize the software?**

**2. What kind of computer do you have?**

**3. What special name do you want for your software?**

**4. Does it have any specific functionality for proper functioning?**

**5. What is the volume of inventory you would like the software to handle?**

**6. What type of product do you sell, do you think it is necessary to categorize them?**

**7. Do you need software to manage product information?**

**8. What do you hope to improve with the software?**

**9. The system must enter with some other existing software, such as accounting and logistics systems.**

**10. Do you need the system to manage customer information, such as contact data, purchase history?**

**11. Does the system have to be able to manage promotions, discounts, special offers?**

**12. How is the system expected to handle taxes and billing?**

**13. Is the software required to support online selling and shipment tracking?**

**14. Are there specific legal compliance requirements that the software must meet?**

5. References

Pressman, R. S. (2014). Ingeniería del Software: Un enfoque práctico (8va ed.). McGraw Hill.

Sommerville, I. (2015). Ingeniería de Software (10ma ed.). Addison-Wesley.

Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.

Fowler, M. (2018). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley Professional.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.