

**Software Requirements**

**Specification**

**for**

**<SureFinventory-Detailed Control Application>**

****

**Version 0.1**

**Prepared by <Esteban, Matias, Abner & Benjamin>**

**<JABAS team>**

**<November 10th 2024>**

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

# Table of Contents

[**Table of Contents ii**](#_heading=h.gjdgxs)

[**Revision History ii**](#_heading=h.30j0zll)

1. [**Introduction 1**](#_heading=h.1fob9te)
   1. [Purpose 1](#_heading=h.3znysh7)
   2. [Document Conventions 1](#_heading=h.2et92p0)
   3. [Intended Audience and Reading Suggestions 1](#_heading=h.tyjcwt)
   4. [Product Scope 1](#_heading=h.3dy6vkm)
   5. [References 1](#_heading=h.1t3h5sf)
2. [**Overall Description 2**](#_heading=h.4d34og8)
   1. [Product Perspective 2](#_heading=h.2s8eyo1)
   2. [Product Functions 2](#_heading=h.17dp8vu)
   3. [User Classes and Characteristics 2](#_heading=h.3rdcrjn)
   4. [Operating Environment 2](#_heading=h.26in1rg)
   5. [Design and Implementation Constraints 2](#_heading=h.lnxbz9)
   6. [User Documentation 2](#_heading=h.35nkun2)
   7. [Assumptions and Dependencies 3](#_heading=h.1ksv4uv)
3. [**External Interface Requirements 3**](#_heading=h.44sinio)
   1. [User Interfaces 3](#_heading=h.2jxsxqh)
   2. [Hardware Interfaces 3](#_heading=h.z337ya)
   3. [Software Interfaces 3](#_heading=h.3j2qqm3)
   4. [Communications Interfaces 3](#_heading=h.1y810tw)
4. [**System Features 4**](#_heading=h.4i7ojhp)
   1. [System Feature 1 4](#_heading=h.2xcytpi)
   2. [System Feature 2 (and so on) 4](#_heading=h.1ci93xb)
5. [**Other Nonfunctional Requirements 4**](#_heading=h.3whwml4)
   1. [Performance Requirements 4](#_heading=h.2bn6wsx)
   2. [Safety Requirements 5](#_heading=h.qsh70q)
   3. [Security Requirements 5](#_heading=h.3as4poj)
   4. [Software Quality Attributes 5](#_heading=h.1pxezwc)
   5. [Business Rules 5](#_heading=h.49x2ik5)
6. [**Other Requirements 5**](#_heading=h.2p2csry)

[**Appendix A: Glossary 5**](#_heading=h.147n2zr)

[**Appendix B: Analysis Models 5**](#_heading=h.3o7alnk)

[**Appendix C: To Be Determined List 6**](#_heading=h.23ckvvd)

# Revision History

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

*The purpose of the following document is to define and structure the software requirements for the SureFinventory application, where we will explain the following:*

1. *Establish the functional requirements for SureFinventory to provide user authentication, inventory management, cash closing, basic accounting management and generate monthly reports.*
2. *Describe the non-functional requirements that ensure a secure, accessible, intuitive and customizable user experience, with language configuration options (English and Spanish) to facilitate adoption.*
3. *Delimit the scope of the system to specify that this application will cover the daily operations of the cafeteria, facilitating control and decision making.*

## Document Conventions

*As the main source of requirements we use the book Clean Code, the author of the book, Robert C. Martin, invites each of the group members to review and analyze a large amount of code, identify problems and learn how to solve them.*

## Intended Audience and Reading Suggestions

*The document is addressed to the profiles involved within the development of the SureFinventory application, we have 3 main recipients within the project, which are:*

1. *Developers: they are the ones who provide a detailed description of the functional and non-functional requirements necessary for the creation of SureFinventory must read all the specific requirements, focusing on the essential functionalities and quality standards of the system.*
2. *Project Managers: For managers, it is recommended to start with the introductory and product purpose sections to get an overview of the project scope and objectives.*
3. *End Users: They can review functional requirements and configurations to understand the application's operation and options.*

## Product Scope

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

## References

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# Overall Description

## Product Perspective

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## Product Functions

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

## User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## Operating Environment

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## User Documentation

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## Assumptions and Dependencies

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# External Interface Requirements

## User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on.*

*Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## System Feature 1

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

### Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

### Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

### Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs.*

*Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

### REQ-1:

### REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

*The software must be useful for daily work; about the data performance, an apliccation that can save several ammounts of data is needed, so it´s important to find an efficent way to save and pull data in a database located at the computer where the app will be running. In the other hand, talking about the users traffic, we just need the app to perform with one user at a time.*

*The focus in the performance requirement is essencially the speed at managing data.*

## Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## Security Requirements

*The application requires to have different types of users, including an admin that can controll everything in the system. Also is needed to have backups for the data and versions to retrieve in case of any human failure when using the platform.*

## Software Quality Attributes

*In this aspect the software needs to be intuitive for the user. Usefulness is the main objective in this project so the quality is going to adapt these terms.*

## Business Rules

*There are no business rules for the development of the softw*

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

# Appendix A: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

# Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

# Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*

*Second Interview*

***Good evening, Mrs. Mayra.****I’m here to ask you some questions regarding the previous interview we had. I need to fill out some more information. Now, I need a bit more specific information, as I mentioned in the message.  
I would like to know what processes you follow here and also, based on those processes, what requirements you would need the app to handle for you.  
To do that, I would need to understand the processes you have in place. For example, how do you manage the accounting for the day?  
How do you enter the daily orders? Let’s start there.*

***Good evening. As I explained last time, we do everything manually. We have sales notes and order notes.****So, in the order notes, we take all the customer orders, and that’s how we keep a sales record. Then, we use a notebook in which we manually record all the sales of the day, and we do a cash closing where we detail sales by card, cash, or transfers.  
And, well, we also record any expenses depending on the day, for example, purchases of products required by the cafeteria, but this can vary because, in general, we make purchases once a week.  
And this is also recorded manually.*

***Where do you make the purchases, Mrs. Mayra?****We use several methods. Some products are delivered to us, and payments are made via transfer. For other products, I go to the supermarket to buy them, and, well, I also record those purchases.  
And I also order online and have the products delivered here.*

***In this case, would you have the documentation, such as receipts, transfers, or card statements, for the products you purchase?****Yes, I have all the invoices, exactly.*

***Okay, perfect. I would need your help to provide this information so I can enter the expenses into the app.****It would really be independent of the payment method, but rather just the amount of the expense for the purchases made.  
So, apart from pantry items, what other expenses do you have?*

***Well, we have direct and indirect expenses. Direct expenses include products required by the cafeteria, employee salaries, and also the payments for basic services, like water and electricity.  
Is the location owned or rented?****No, it’s rented.****Okay, so that would be an additional cost too.****Yes, of course.****And do you also have costs related to renovations or repairs within the premises?****Yes, for example, we also have maintenance expenses for the cafeteria and the premises, and we also process documents, such as with the SRI (Tax Authority), which is annual.  
We also do renovations, and sometimes we sell products or items we have here, so we also need to replace that space.*

***Okay, but at the moment, do you not specify the expenses separately? You group them all together and then subtract them from the income? How do you report your final monthly income and expenses?****No, we detail our direct and indirect expenses. For example, maintenance and renovation costs are indirect expenses that the cafeteria incurs, and those are necessary.  
At the moment, are you keeping track of this in Excel or...?  
Everything is manual. Everything is done manually.  
Okay, I recall you mentioned an accounting book—could you send me the format you use?  
Yes, we keep a record of only the most relevant things, like the direct expenses the cafeteria needs.*

***Alright, I now have a clearer idea of what we would need for the app.****I’d also like to know, in terms of inventory, which products should be tracked?  
I understand you have a lot of products coming in and out, so what are the finished products that should be tracked?  
For example, how many cakes or coffee do you have? Could you tell me which products need to be inventoried to have more precision?*

***Yes, the dishes are specific, so we do an inventory of all the products and also track them with the purchases we make online.****For example, I buy packs of a dish we have on the menu, which is the "portavelos."  
I know that each pack has three portavelos, and I buy four packs a week, once a week, so I know how many will be sold.  
So, in that way, I keep track. On weekends, I also do a detailed record of the weekly sales to know which products I’ve sold.*

***Okay, you mentioned an Excel sheet before. Could you share that with me?****Yes, we have an Excel sheet where we track the products we have in the cafeteria. No problem.*

***Perfect, I think that’s basically the information I need.****We now know where you buy products, how payments are made, and how you’ve been managing things so far.  
So, what I understand is that you need a basic report, correct?  
I would also like to know, by the end of the month, do you know your net income or your profits, or have you had trouble identifying that benefit?*

***The truth is, it’s complicated for me. I’ve only managed it weekly, but I haven’t done a monthly report. The people who work here help me, but it’s very basic.***

***Now, let’s get into specifics. What would you need the app to do?****It’s mostly for inventory management and also for the orders. For example, when a customer comes in and orders a cappuccino, the staff sometimes says, "Let’s note it down," but for one order, they use a comanda. It’s quite informal, and it can’t be managed this way. The app would help visualize and track everything properly.*

***Would you need the app to have a username and password?****Yes.****And should there be profiles, one for the administrator and one for employees, or would it be just for you?****I think both would be necessary.  
Yes, that would be fine.*

***Perfect. We could also add features like inventory management, sales tracking, and updates.****As I understand, you make purchases weekly, so we could have a report before each purchase to show which products are running low. Would that be useful?****Yes, that way we can check which products are running out and which ones have been sold because we don’t keep track of that at the moment.***

***Okay, based on our previous conversation, I’ve compiled a list of what would be necessary to solve this.****The app should have an interface for entering daily orders, allowing you to close the cash register, showing the amount that should be in the cash drawer, the sales by card, and everything. This would make it easier to close the cash.  
It should also be connected to a database that can break down basic accounting, showing income and expenses, and generating a monthly report, which I believe you requested.****Yes.***

***Also, since I understand you are about to change the menu, it would be important to modify the product costs and the menu itself.****It wouldn’t be practical for me to make all the changes. It should be intuitive and easy for you to enter the settings and change the product prices if that works for you.****Yes.***

***Apart from all that, those would be the main functions of our program. I’d like to ask if there’s a specific color you’d like the app to have.****We use turquoise and red as our brand colors.****Perfect, turquoise and red it is.  
Would you prefer the app to be in any specific language?****Yes, it’s important for it to be available in both English and Spanish.****Great, we can set up a language toggle between English and Spanish.****If possible, there will also be design customization options, but language switching seems very feasible.*

***Putting all that together, I think I now have a solid list of requirements, and I have the necessary information.****I just need the rest of the details that we agreed to follow up on, such as how you manage everything manually, so we can prepare a format for automating these processes.*