Software Requirements Specification

for

<Project>

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 2

1.1 Purpose 2

1.2 Document Conventions 2

1.3 Project Scope 2

1.4 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

3. System Features 2

3.1 System Feature 1 2

3.2 System Feature 2 (and so on) 2

4. Data Requirements 2

4.1 Logical Data Model 2

4.2 Data Dictionary 2

4.3 Reports 2

4.4 Data Acquisition, Integrity, Retention, and Disposal 2

5. External Interface Requirements 2

5.1 User Interfaces 2

5.2 Software Interfaces 2

5.3 Hardware Interfaces 2

5.4 Communications Interfaces 2

6. Quality Attributes 2

6.1 Usability 2

6.2 Performance 2

6.3 Security 2

6.4 Safety 2

6.5 [Others as relevant] 2

7. Internationalization and Localization Requirements 2

8. Other Requirements 2

Appendix A: Glossary 2

Appendix B: Analysis Models 2

Revision History

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

# Introduction

<The introduction presents an overview to help the reader understand how the SRS is organized and how to use it.>

## Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers.>

## Document Conventions

<Describe any standards or typographical conventions used, including the meaning of specific text styles, highlighting, or notations. If you are manually labeling unique requirement identifiers, you might specify the format here for anyone who needs to add one later.>

## Project Scope

<Provide a short description of the software being specified and its purpose. Relate the software to user or corporate goals and to business objectives and strategies. If a separate vision and scope or similar document is available, refer to it rather than duplicating its contents here. An SRS that specifies an incremental release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision. You might provide a high-level summary of the major features the release contains or the significant functions that it performs.>

## References

<List any documents or other resources to which this SRS refers. Include hyperlinks to them if they are in a persistent location. These might include user interface style guides, contracts, standards, system requirements specifications, interface specifications, or the SRS for a related product. Provide enough information so that the reader can access each reference, including its title, author, version number, date, and source, storage location, or URL.>

# Overall Description

<This section presents a high-level overview of the product and the environment in which it will be used, the anticipated users, and known constraints, assumptions, and dependencies.>

## Product Perspective

## 

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product and describe their pertinent characteristics. Some requirements might pertain only to certain user classes. Identify the favored user classes. User classes represent a subset of the stakeholders described in the vision and scope document. User class descriptions are a reusable resource. If available, you can incorporate user class descriptions by simply pointing to them in a master user class catalog instead of duplicating information here.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform; operating systems and versions; geographical locations of users, servers, and databases; and organizations that host the related databases, servers, and websites. List any other software components or applications with which the system must peacefully coexist. If extensive technical infrastructure work needs to be performed in conjunction with developing the new system, consider creating a separate infrastructure requirements specification to detail that work.>

## Design and Implementation Constraints

<Describe any factors that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing or memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; programming language requirements or restrictions.>

## 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, reuse expectations, 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 outside its control.>

# System Features

## Order Feature

<

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | 6. Order Product | | |
| Created By: | Tran Nhat Minh | Date Created: | 19/01/2021 |
| Primary Actor: | Customer | Secondary Actors: | Clothes Inventory System |
| Trigger: | User has chosen a product and click add to cart | | |
| Description: | A customer accesses the Clothes Web System from the website and views the clothes, selects clothes items, and add it to cart. After all customer add address and phone number and clothes to be delivered to them. | | |
| Preconditions: | 1. Guest has logged in | | |
| Postconditions: | 1. Clothes order is stored in database with a status of “Accepted” 2. Update quantity clothes in database | | |
| Normal Flow: | 1. Clothes Web System display clothes and discount. 2. Customer click to cloth for view more detail. 3. Clothes Web System display cloth pictures, cloth size, cloth quantity, and cloth price. 4. Customer clicks add product to cart. 5. Clothes Web System display number of products in Cart. 6. Customer clicks to cart to prepare payment. 7. Clothes Web System display number and detail of clothes in cart. 8. Customer specifies payment method. 9. Clothes Web System display total price include the price of clothes and payment method. 10. Customer confirms order. 11. Clothes Web System confirms acceptance of the order. 12. Clothes Web System stores order to database sends order information to store. | | |
| Alternative Flows: | 1. If cart already existed, Clothes Web System return to old cart. 2. If cart not existed, Clothes Web System create new cart. 3. If quantity in database is less than quantity clothe in order, Clothes Web System notify to customer to change and can’t confirm order. | | |
| Exceptions: | Quantity of clothe in database is less than quantity of clothe in order | | |
| Priority: | High | | |
| Frequency of Use: | Every time the customer wants to buy a clothes | | |
| Business Rules: | 1. All cloth in the order must be paid for by using the same payment method. 2. All cloth in the order must delivered in the time. 3. Order price is calculated as the sum of each cloth item price times the quantity of that clothe item ordered, plus a delivery charge if a meal is delivered outside the free delivery zone. 4. The price of delivered is free:  * If distance less than 500km and the total price of clothe is more than 500.000 vnd * If distance less than 1000km and the total price of clothe is more than 1.000.000vnd  1. Network transmissions that involve financial information or personally identifiable information require 256-bit encryption. 2. Payment method is  * Cash On Delivery (COD) * Visa * Digital Wallet | | |
| Other Information: |  | | |
| Assumptions: | Assume that 10 percent of Custoemr will order the clothe in normal. | | |

## Calculate Salary Feature

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | Calculate salary | | |
| Created By: | Văn Đức Huy | Date Created: | 19/01/2021 |
| Primary Actor: | Staff of store | Secondary Actors: | Staff account Database |
| Trigger: | Automatic at the end of the month | | |
| Description: | Calculate salary of every staffs of the store base on monthly sales | | |
| Preconditions: | 1. Have a staff account 2. Work at the store | | |
| Postconditions: | 1. The system calculate salary for staff at the end of the month 2. Staff receive salary at the bank account | | |
| Normal Flow: | 1. The system checks every staff account and get the monthly sales 2. The system will automatic calculate the salary of staff 3. Notification for staff 2 days after transfer money 4. The system transfers the money into the staff bank account base on the account number in the staff information | | |
| Alternative Flows: | 1. Invalid staff account because the staff is terminated when he/she quit the job  2. If the staff doesn’t add the number account information the system must notify him/her | | |
| Exceptions: | 1. Can’t transfer the money -> in this case the store owner will receive the notify and pay the salary for staff in cash  2. The money being rollback because the bank account doesn’t exist -> notify staff by email  3. Number account isn’t true -> notify for staff by email or at the login session of staff | | |
| Priority: | Medium | | |
| Frequency of Use: | Once a month | | |
| Business Rules: | The system will calculate income of staff = Position salary (P1) + Competency salary (P2) + Performance salary (P3) + Sales salary (% of sales) | | |
| Other Information: | Store owner can modify this salary formula at any time | | |
| Assumptions: |  | | |

System Feature 2 (and so on)

# Data Requirements

<This section describes various aspects of the data that the system will consume as inputs, process in some fashion, or create as outputs.>

## Logical Data Model

## 

## Data Dictionary

* Shopping Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Element | Description | Composition or Data Type | Length | Values |
| CustomerID | Number ID of the customer who buy the clothes | Integer |  | Auto increase – INDENTITY(1,1)  Start from 1 |
| Customer Phone | Telephone number of the customer who ordered cloth | Alphabetic | 20 |  |
| Customer Address | Address of the customer for delivery | Alphabetic | 200 |  |
|  |  |  |  |  |
| CategoryID | Number ID of the category of cloth | Integer |  | Auto increase - INDENTITY(1,1)  Start from 1 |
| Category Name | Name of the category which cloth attached | Alphabetic | 50 |  |
| ClothID | Number ID of the cloth | Integer |  | Auto increase - INDENTITY(1,1)  Start from 1 |
| Cloth Name | Name of the cloth | Alphabetic | 50 |  |
| Cloth Color | Color of the cloth | Alphabetic | 50 |  |
| Cloth Image | Image detail of the cloth | Alphabetic | 200 |  |
| Cloth Quantity | Quantity left of the cloth which customer can order | Integer |  |  |
| Cloth Size | Size of the cloth which user choose when they order | Alphabetic | 20 |  |
| Order Detail Quantity Cloth | Quantity of cloth user want to order |  |  | Maximum quantity = quantity presently in inventory |
| Order Detail Size | Size of cloth user want to order | Alphabetic | 20 |  |
| Order Total Price | Total price of order | Numeric, Dollars and Cents |  |  |
| OrderID | Unique ID of the order which created when customer confirm the order | Alphabetic | 20 | Auto generate format: YYYYMMdd-hhmmss-xxxx (xxxx is a number generate) |
| Order Date | The date of order when user order | Date |  | Default is current date |
| Order Payment Method | How the customer is paying for a meal he ordered | Alphabetic | 20 | Cash on delivery, cash, credit card, debit card |
| Order Delivery Cost | Price of the delivery | Numeric, Dollars and Cents |  |  |

* Calculate Staff Salary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Element | Description | Composition or Data Type | Length | Values |
| StaffID | Store ID of Staff | Varchar | 50 |  |
| StaffName | Store Name of Staff | Nvarchar | 50 |  |
| Phone | Store phone number of Staff | Varchar | 20 | Validate when insert just accept number |
| BankNumber | Store bank number of staff | Varchar | 30 | Validata when insert  Just accept number and (-) |
| SalaryDate | Store salary date of staff | Date(MM-DD-YYYY) |  |  |
| Salary | Store monthly salary of staff | Real |  |  |
| monthlySales | Store the quantitfy of product of this staff sale in month | Int |  | Validate when insert |

## Reports

<If your application will generate any reports, identify them here and describe their characteristics. If a report must conform to a specific predefined layout you can specify that here as a constraint, perhaps with an example. Otherwise, focus on the logical descriptions of the report content, sort sequence, totaling levels, and so forth, deferring the detailed report layout to the design stage.>

## Data Acquisition, Integrity, Retention, and Disposal

<If relevant, describe how data is acquired and maintained. State any requirements regarding the need to protect the integrity of the system's data. Identify any specific techniques that are necessary, such as backups, checkpointing, mirroring, or data accuracy verification. State policies the system must enforce for either retaining or disposing of data, including temporary data, metadata, residual data (such as deleted records), cached data, local copies, archives, and interim backups.>

# External Interface Requirements

<This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.>

## 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.>

## Software Interfaces

<Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the intercomponent communications. Identify data that will be exchanged between or shared across software components. Specify nonfunctional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.>

## Hardware Interfaces

<Describe the characteristics of each interface between the software and hardware (if any) components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used. List the inputs and outputs, their formats, their valid values or ranges, and any timing issues developers need to be aware of. If this information is extensive, consider creating a separate interface specification document.>

## Communications Interfaces

<State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.>

# Quality Attributes

## Usability

<Specify any requirements regarding characteristics that will make the software appear to be “user-friendly.” Usability encompasses ease of use, ease of learning; memorability; error avoidance, handling, and recovery; efficiency of interactions; accessibility; and ergonomics. Sometimes these can conflict with each other, as with ease of use over ease of learning. Indicate any user interface design standards or guidelines to which the application must conform.>

## Performance

<State specific performance requirements for various system operations. If different functional requirements or features have different performance requirements, it's appropriate to specify those performance goals right with the corresponding functional requirements, rather than collecting them in this section.>

## Security

<Specify any requirements regarding security or privacy issues that restrict access to or use of the product. These could refer to physical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.>

## Safety

<Specify requirements that are concerned with possible loss, damage, or harm that could result from use of the product. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the product must conform.>

## [Others as relevant]

<Create a separate section in the SRS for each additional product quality attribute to describe characteristics that will be important to either customers or developers. Possibilities include availability, efficiency, installability, integrity, interoperability, modifiability, portability, reliability, reusability, robustness, scalability, and verifiability. Write these to be specific, quantitative, and verifiable. Clarify the relative priorities for various attributes, such as security over performance.>

# Internationalization and Localization Requirements

<Internationalization and localization requirements ensure that the product will be suitable for use in nations, cultures, and geographic locations other than those in which it was created. Such requirements might address differences in: currency; formatting of dates, numbers, addresses, and telephone numbers; language, including national spelling conventions within the same language (such as American versus British English), symbols used, and character sets; given name and family name order; time zones; international regulations and laws; cultural and political issues; paper sizes used; weights and measures; electrical voltages and plug shapes; and many others.>

# Other Requirements

<Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections. >

Appendix A: Glossary

<Define any specialized terms that a reader needs to know to understand the SRS, including acronyms and abbreviations. Spell out each acronym and provide its definition. Consider building a reusable enterprise-level glossary that spans multiple projects and incorporating by reference any terms that pertain to this project.>

Appendix B: Analysis Models

<This optional section includes or points to pertinent analysis models such as data flow diagrams, feature trees, state-transition diagrams, or entity-relationship diagrams. You might prefer to insert certain models into the relevant sections of the specification instead of collecting them at the end.>