Software Requirements Specification

for

<Online Bakery>

Version 1.0 approved

Prepared by Group 1

12/01/2020

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 of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, reference, system proposals and an overview of the SRS.

## Purpose

The purpose of this document is to collect, analyze and provide a consistent and complete description of the software requirements: Online Bakery. The Online Pie System (OB) helps manage all online transactions in a systematic order, which is beneficial for both the seller and the customer. Help managers more easily in marketing, bringing products to consumers, expanding business markets. Users can access the product easier, more convenient in product selection.

## Document Conventions

The following document conventions are followed during the preparation of this SRS:

* All keywords related to scholars are in italic type.
* All folder headlines are in bold type
* Comments and proper names may be enclosed in parentheses

## Project Scope

This system is convenient for computerizing all bakery activities. It focuses on seller (server side) and customer (client side) to provide online sales and related activities. The seller can add and modify cakes, manage information about the available cakes in the store (how many pieces are available, what is available, what is out of stock ...), business management sales, customer information management, delivery management, statistics.

## References

The topic is referenced and designed based on the material:

* <https://www.academia.edu/>
* <https://www.slideshare.net/>
* [https://www.slideshare.net](https://www.slideshare.net/)
* [https://fr.scribd.com](https://fr.scribd.com/)
* [https://sourceforge.net](https://sourceforge.net/)

# Overall Description

## Product Perspective

● This website provides a bakery service for customers and a management system for sellers.

● It aims to motivate customers to order cakes by offering enhanced features:

* Chatbox to interact with customers
* Voice recognition to support special use cases such as customer bound or illiterate
* Personalized product suggestions to match customers Taste.
* Designing products according to customers' requirements.
* This website is a Java web-based application, so any device with a web browser and Internet connection will be able to evaluate.

● Chatbox and speech recognition are created using the Microsoft Awareness Service and Google Text Transfer API, so these features will depend on the availability of Microsoft and Google third-party services.

## 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.>

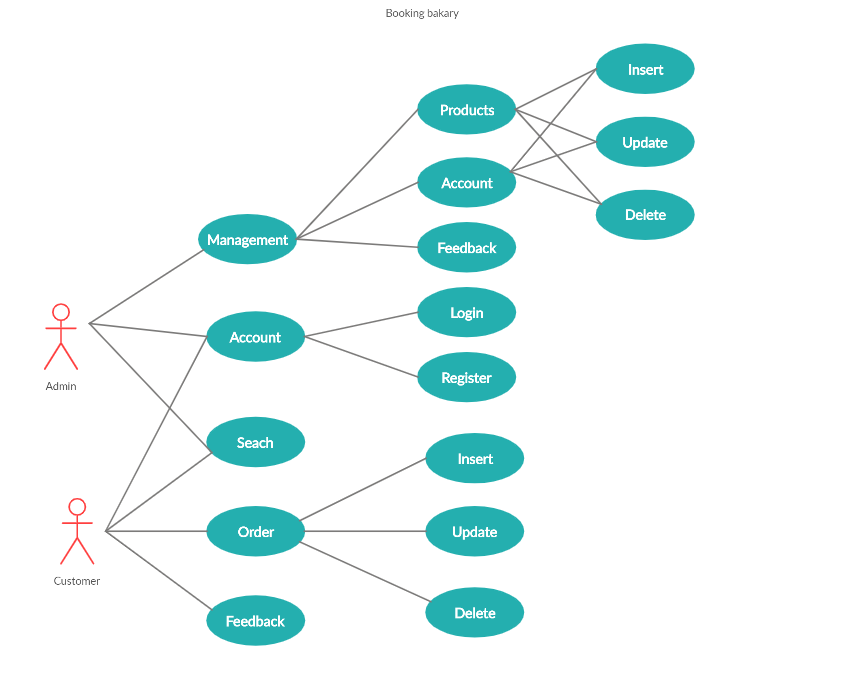
## Design and Implementation Constraints

<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

Use case diagam here

System feature tree here

Context diagram here

……..

## System Feature Manage Account

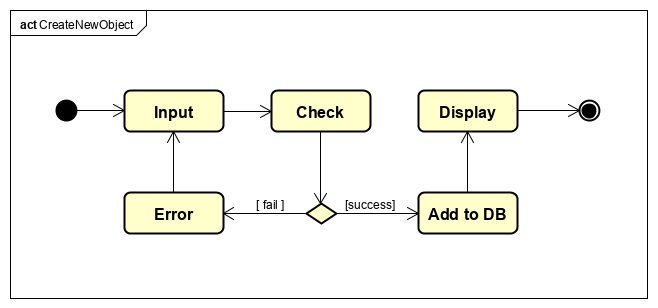
### Description

Writing short description about manage account here

### Stimulus/Response Sequences

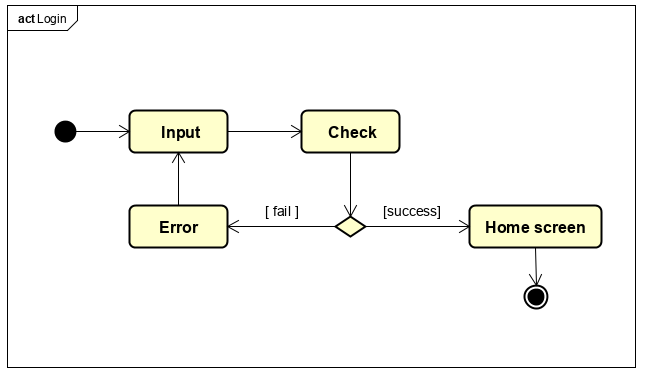
List all event and FAP system must response with the even. Should write in table formate

### Functional Requirements Create new Account

**

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | Create new Account | | |
| Created By: | Group 1 | Date Created: | 09/01/2020 |
| Primary Actor: | User | Secondary Actors: |  |
| Trigger: | None | | |
| Description: | Show the users how to create a new account | | |
| Preconditions: | Everyone can access | | |
| Postconditions: | Everyone can access | | |
| Normal Flow: | 1. The system requires the user to enter information 2. The User will enter email , name and password and click Sign up 3. The system check data input and allows user to access the system. | | |
| Alternative Flows: | None | | |
| Exceptions: | The system will give you an error if you have not filled in the information | | |
| Priority: | None | | |
| Frequency of Use: | Unlimited | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

### Function Requirements Login



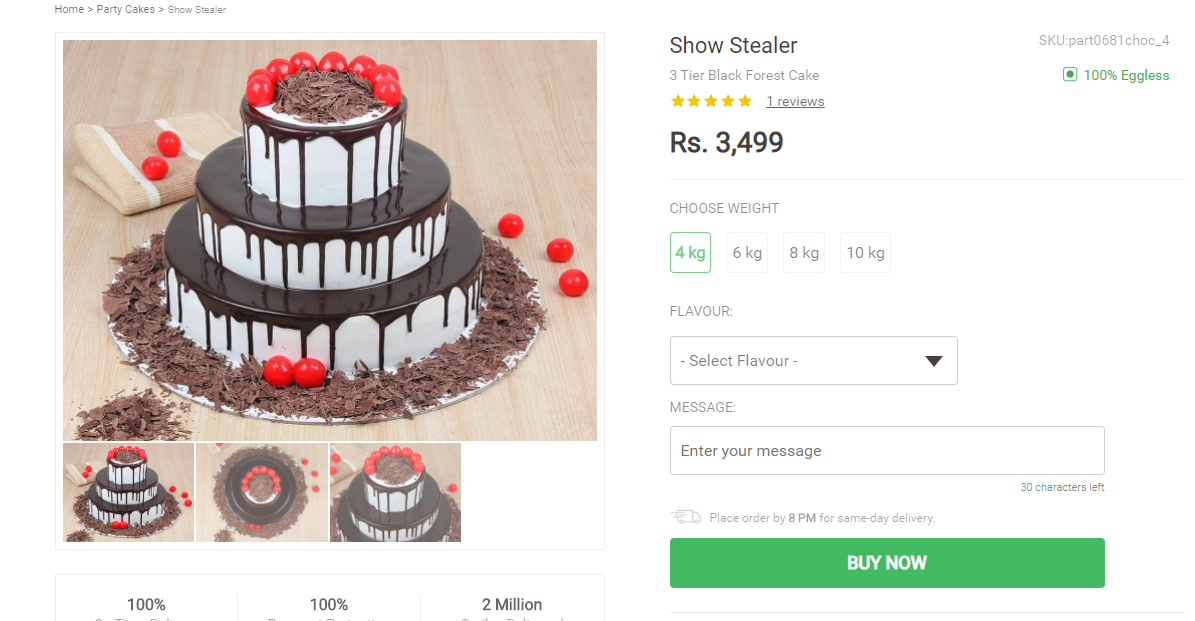
|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | Login | | |
| Created By: | Group 1 | Date Created: | 09/01/2020 |
| Primary Actor: | User | Secondary Actors: |  |
| Trigger: | No | | |
| Description: | Show the users how to login the system | | |
| Preconditions: | User must have account | | |
| Postconditions: | None | | |
| Normal Flow: | 1. The system requires the user to enter username and password 2. The User will fill in and click Sing In 3. The system check data input and allows user to access the system | | |
| Alternative Flows: | None | | |
| Exceptions: | An error message will appear if username or password you entered is invalid | | |
| Priority: | None | | |
| Frequency of Use: | Unlimited | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

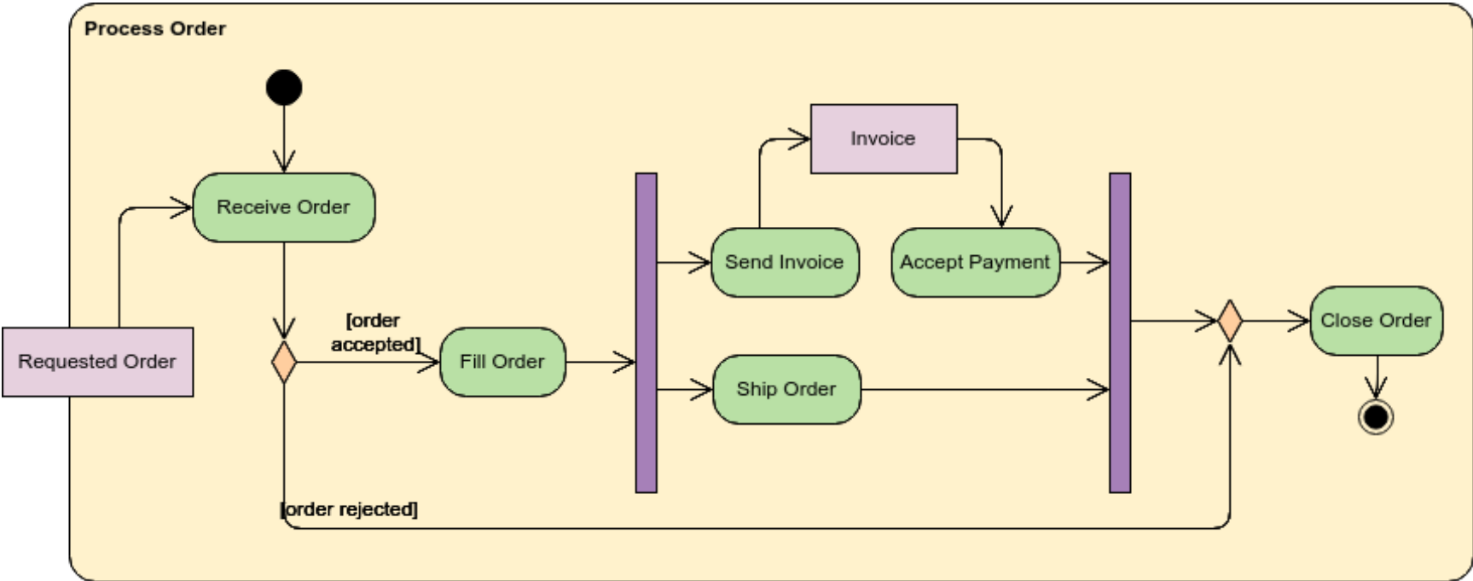
## System Feature Order

### Description

### Stimulus/Response Sequences

**3.2.3 Function Requirements Order**

****



|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | Order | | |
| Created By: | Group 1 | Date Created: | 09/01/2020 |
| Primary Actor: | User | Secondary Actors: |  |
| Trigger: | No | | |
| Description: | Show the users how to order products | | |
| Preconditions: | User must have account | | |
| Postconditions: | None | | |
| Normal Flow: | 1. The system requires the user to do procedure for ordering 2. The User will fill in and click Buy Now 3. The system check data input and send the user some feedback | | |
| Alternative Flows: | None | | |
| Exceptions: | An error message will appear if user not provide enough information | | |
| Priority: | None | | |
| Frequency of Use: | Unlimited | | |
| Business Rules: | None | | |
| Other Information: | None | | |
| Assumptions: | None | | |

## System Feature Insert/Update/Delete

### Description

### Stimulus/Response Sequences

### Function Requirements Insert/Update

### Function Requirements Delete

Draw the GUI of the screen import grade function and insert here

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | Import grade | | |
| Created By: | TrungNT | Date Created: | 12/2/2009 |
| Primary Actor: | Admin | Secondary Actors: |  |
| Trigger: |  | | |
| Description: |  | | |
| Preconditions: | The user must already have the valid account in the FAP System. | | |
| Postconditions: | After finish this function, the account must be updated and store in the database of the FAP system | | |
| Normal Flow: | 1. Fisrt, the admin user must access the screen create new account | | |
| Alternative Flows: |  | | |
| Exceptions: |  | | |
| Priority: |  | | |
| Frequency of Use: |  | | |
| Business Rules: |  | | |
| Other Information: |  | | |
| Assumptions: |  | | |

### function update grade

Draw the GUI of the screen import grade function and insert here

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | Import grade | | |
| Created By: | TrungNT | Date Created: | 12/2/2009 |
| Primary Actor: | Admin | Secondary Actors: |  |
| Trigger: |  | | |
| Description: |  | | |
| Preconditions: | The user must already have the valid account in the FAP System. | | |
| Postconditions: | After finish this function, the account must be updated and store in the database of the FAP system | | |
| Normal Flow: | 1. Fisrt, the admin user must access the screen create new account | | |
| Alternative Flows: |  | | |
| Exceptions: |  | | |
| Priority: |  | | |
| Frequency of Use: |  | | |
| Business Rules: |  | | |
| Other Information: |  | | |
| Assumptions: |  | | |

### function delete grade

# 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

<A data model is a visual representation of the data objects and collections the system will process and the relationships between them. Include a data model for the business operations being addressed by the system, or a logical representation for the data that the system itself will manipulate. Data models are most commonly created as an entity-relationship diagram.>

## Data Dictionary

<The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, you're better off storing the data dictionary as a separate artifact, rather than embedding it in the middle of an SRS. That also increases its reusability potential in other projects.>

## 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.>