**SOFTWARE**

**REQUIREMENTS**

**SPECIFICATION**

**Prepared for**

Electronic device business website

**Prepared by**

Nguyễn Dương Tùng – 19522494

Nguyễn Võ Đức Thắng – 19522213

Vương Hoàng Trung – 19522438

Đặng Hải Hoàng Sơn - 19522131

**Version 1.0**

**Lecture: NGUYỄN THỊ THANH TRÚC**

**Acknowledgement**

First and foremost, Our team would like to express our sincere thanks to lecturer Nguyen Thi Thanh Truc for her enthusiastic teaching and guidance during the implementation of the subject project. The knowledge gained during the project implementation as well as throughout the course is extremely valuable, contributing significantly to the completion of the project.

Our team wishes you health, happiness and success.

**Project Team.**

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Description** |
| 01/11/2021 | 1.0 | Nguyễn Dương Tùng | Project Initialization |
|  |  |  |  |
|  |  |  |  |

**Approval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Approver Name** | **Position** |
| 01/11/2021 | 1.0 | Nguyễn Thị Thanh Trúc | Project Owner |
|  |  |  |  |
|  |  |  |  |

This document has been approved as the official Software Requirements Specification for *“Electronic device business website”* software application, and accurately reflects the current understanding of software requirements specification. Following approval of this document, requirement changes will be governed by the project’s change management process, including impact analysis, appropriate reviews, and approvals.

**DOCUMENT CONTENT**

[1. Introduction 6](#_Toc106132210)

[**1.1.** **Purpose** 6](#_Toc106132211)

[**1.2.** **Scope** 6](#_Toc106132212)

[**1.3.** **Technical Requirements From Customer** 6](#_Toc106132213)

[**1.4.** **Intended Audiences and Document Organization** 6](#_Toc106132214)

[2. Functional Requirement 8](#_Toc106132215)

[**2.1.** **Use-case diagram** 8](#_Toc106132216)

[**2.2.** **Use-case specification** 8](#_Toc106132217)

[**2.2.1.** **User authentication** 8](#_Toc106132218)

[**2.2.1.1.** **Sign In** 9](#_Toc106132219)

[**2.2.1.2.** **Sign Up** 10](#_Toc106132220)

[**2.2.2.** **Sales management** 11](#_Toc106132221)

[**2.2.2.1.** **Find Product** 11](#_Toc106132222)

[**2.2.2.2.** **Add/remove to/from Cart** 11](#_Toc106132223)

[**2.2.2.3.** **Add/remove to/from Wish list** 12](#_Toc106132224)

[**2.2.2.4.** **Pay for order** 13](#_Toc106132225)

[**2.2.2.5.** **Verify Payment** 14](#_Toc106132226)

[**2.2.2.6.** **Compare Product** 15](#_Toc106132227)

[**2.2.2.7.** **CRUD Product** 16](#_Toc106132228)

[**2.2.2.8.** **CRUD Promotion** 17](#_Toc106132229)

[**2.2.2.9.** **CRUD Order** 18](#_Toc106132230)

[**2.3.** **Actor description** 19](#_Toc106132231)

[1.1. **Context diagram** 20](#_Toc106132232)

[**1.3.** **Workflow** 25](#_Toc106132233)

[**1.3.1.** **Sequence diagram** 25](#_Toc106132234)

[29](#_Toc106132235)

[**1.3.2.** **State diagram** 29](#_Toc106132236)

[**1.4.** **Business rules** 36](#_Toc106132237)

[2. Non-functional requirement 47](#_Toc106132238)

[**2.1.** **Security matrix** 47](#_Toc106132239)

[**2.2.** **Performance requirement** 48](#_Toc106132240)

[**2.3.** **Implementation requirement** 49](#_Toc106132241)

[3. Other Requirement 49](#_Toc106132242)

[**3.1.** **Archive function** 49](#_Toc106132243)

[**3.2.** **Security Function** 50](#_Toc106132244)

[4. System Requirement 50](#_Toc106132245)

[**4.1.** **Permission level** 50](#_Toc106132246)

[**4.2.** **Actor group** 51](#_Toc106132247)

[**4.3.** **User interface** 51](#_Toc106132248)

[**4.4.** **Technical concern** 52](#_Toc106132249)

[5. Installation 52](#_Toc106132250)

[6. Development Direction & Improvement 52](#_Toc106132251)

[7. Conclusion 52](#_Toc106132252)

[8. Appendixes 52](#_Toc106132253)

[**8.1.** **Term** 53](#_Toc106132254)

[**8.2.** **Message** 53](#_Toc106132255)

[**8.3.** **Issue list** 54](#_Toc106132256)

1. **Introduction**
   1. **Purpose**

This Software Requirements Specification and Design document contains the software requirements of *“Electronic device business application that supports product recommendations” which developed on web platform.*

This document and other documents such as Business Requirement Document, Architecture Design Document will be used in the development and finalization of the product.

This document, technically, define how the application will work. Developers will base on this document during the development, division of work, and operation of the new application.

* 1. **Scope**
* The final products solve the requirements of electronic goods management, electronic goods purchase & sale management, and algorithms that support product recommendations according to user information. In addition, a real-time database system supports the correctness of the system when purchasing and managing products.
* Final products support shop’s customer user to buy goods and shop’s staff user to manage system.
  1. **Technical Requirements From Customer**
* Framework: Angular
* Language: JS, TS, HTML, CSS
* Development Software: XAMPP, VS Code
  1. **Intended Audiences and Document Organization**

This document is intended for:

* **Development team:** Responsible to develop detailed design, implement and perform unit test, integration test and system test for the migrated application
* **Data Migration team:** Responsible to create data migration scripts and perform data migration for the application.
* **Documentation Team:** Responsible to writing User Guide for the application.
* **User Acceptance Testing team**: Responsible to conduct user acceptance test sessions with end users.

Below are main sections of the document:

**1. Introduction**: This section describes the general introduction of this document.

**2. Functional Requirements**: This section describesthe functional requirements in detail.

**3. Non-functional Requirements:** This section describes the non-functional requirements of this application such as user access and security, interfaces, screens, and performance.

**4. Other Requirements:** This section describes other requirements such as archive or security audit function.

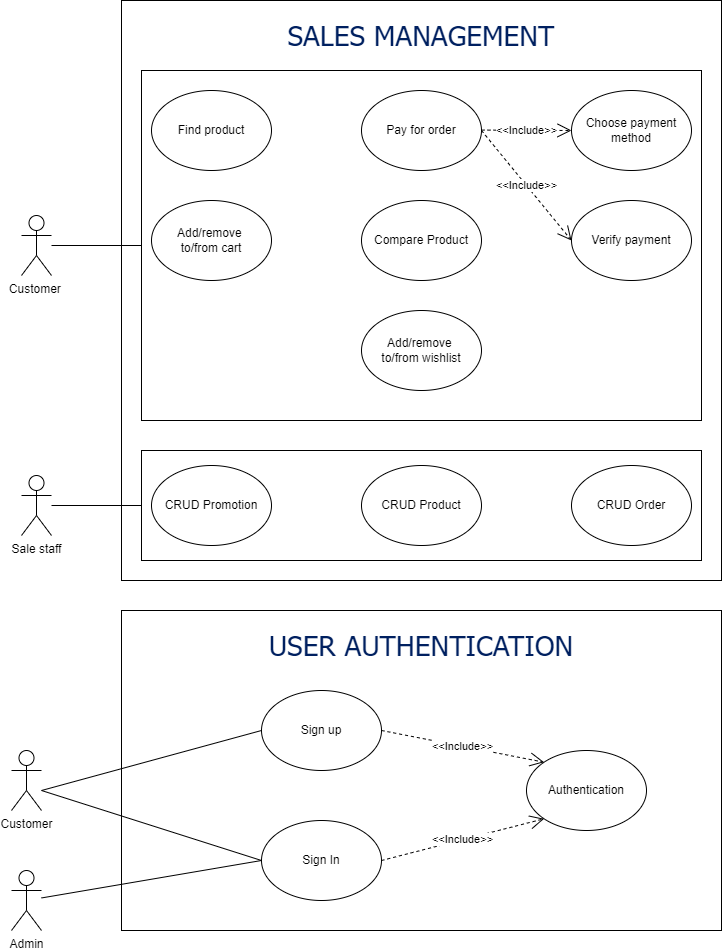
**5.** **Application Design:** This section describes the design of SharePoint application.

**6. Appendixes**: This section describes other requirements for this application and other supporting information for this document**.**

* 1. **References**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Title** | **Version** | **File name/Link** | **Description** |
| 1 | Luật doanh nghiệp 2020 | 2020 |  |  |
| 2 | Thông tư 47 – 2014/TT/BCT | 2014 |  |  |
| 3 | Nghị định 185 – 2013/ND/CP | 2013 |  |  |

1. **Functional Requirement**
   1. **Use-case diagram**

****

* 1. **Use-case specification**
     1. **User authentication**
        1. **Sign In**

|  |  |
| --- | --- |
| **Name** | **Sign in** |
| **Scenario** | Users want to access to system, login to system using email and password. |
| **Description** | This use case occurs when the user wants to use the system, verify the information, and assign the corresponding permissions to the user. After login process, redirect to home page of system. |
| **Actor** | Staff (sale staff, inventory staff, manager), system’s admin, customer |
| **Trigger** | Users click login, enter email and password then submit. |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website and access to Login function. |
| **Post-condition** | **Success:** Users have system access rights according to their respective account type  **Fail:** Show notification and captcha verification before authenticating again |
| **Flow of events** | 1. User access to the system website corresponding to their role (customer or staff)  2. User click “Login” button  3. Redirect to Login page  4. User input the username and password then submit  5. System check the validity of entered email and password  6. If user is existed in system, redirect to homepage of system. |
| **Exception conditions** | If the entered email and password is invalid, system will give notify to ask them try again.  If the entered email and password don’t exist in system, system will give notify to ask them try again.  If the user has forgotten the password, they can click “Forgot password” button |

* + - 1. **Sign Up**

|  |  |
| --- | --- |
| **Name** | **Sign up** |
| **Scenario** | Customer users want to purchase product in system, but they don’t have account to login to system. |
| **Description** | This use case occurs when the user wants to create account to use the system, this use case includes email confirmation to verify the user information. After signup process, redirect to home page of system. |
| **Actor** | Customer |
| **Trigger** | Customer user click sign up, enter email, password, and some basic information such as: name, gender then submit. |
| **Related use cases** | Confirm information via email, Activate account |
| **Pre-condition** | The user has accessed the system's website and access to Sign up function. |
| **Post-condition** | **Success:** Users have system access rights.  **Fail:** Show notification and captcha verification before start signup authenticating again |
| **Flow of events** | 1. User access to the store website.  2. User click “Sign up” button on any screen.  3. Redirect to Sign up page  4. User input the username, password and some basic information then submit  5. System check the validity of entered email, password, and inputted information.  6. Show email verification waiting page.  6. If user is activated, redirect to homepage of system. |
| **Exception conditions** | If the entered email and password is existed, system will give notify to ask them to login.  If the entered email, password, basic information is not valid, system will give notify to ask them try again. |

* + 1. **Sales management**
       1. **Find Product**

|  |  |
| --- | --- |
| **Name** | **Find Product** |
| **Scenario** | Customer users need to find product to view or buy. |
| **Description** | This use case occurs when the user wants to find product in system. After find product process, system return relevant searched keyword product. |
| **Actor** | Customer |
| **Trigger** | User click search box, input keyword, and submit |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website. |
| **Post-condition** | **Success:** Products result relevant to search keyword.  **Fail:** Show notification that can’t find relevant product |
| **Flow of events** | 1. User access to system website  2. Input to search box and click search  3. System show relevant product result |
| **Exception conditions** | If the entered keyword has wrong spelling, auto correct it, and show result. |

* + - 1. **Add/remove to/from Cart**

|  |  |
| --- | --- |
| **Name** | **Add/remove to/from Cart** |
| **Scenario** | Users want to interact with cart, support them to purchase process, |
| **Description** | This use case occurs when the user interacts with cart to support their purchase process. They can pay for their cart any time if their product available. |
| **Actor** | Customer |
| **Trigger** | Users click “Add to cart” or “Remove from cart” in cart page or product page. |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website. |
| **Post-condition** | **Success:** Product added/removed to/from cart.  **Fail:** Show notification to notify them. |
| **Flow of events** | 1. User access to product page or cart page  2. Click to “add to cart” or “remove from cart” or “plus” or “minus” icon button  3. Product will be added or removed from cart |
| **Exception conditions** | If the product is not available, show notification to notify user. |

* + - 1. **Add/remove to/from Wish list**

|  |  |
| --- | --- |
| **Name** | **Add/remove to/from Cart** |
| **Scenario** | Users want to interact with wish list, help them hold what products they want to buy later |
| **Description** | This use case occurs when the user interacts with wish list to support their purchase process. They can move their items to their cart any time if their product available. |
| **Actor** | Customer |
| **Trigger** | Users click “Add to cart” or “Remove from cart” in cart page or product page. |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website. |
| **Post-condition** | **Success:** Product added/removed to/from cart.  **Fail:** Show notification to notify them. |
| **Flow of events** | 1. User access to product page or cart page  2. Click to “add to cart” or “remove from cart” or “plus” or “minus” icon button  3. Product will be added or removed from cart |
| **Exception conditions** | If the product is not available, show notification to notify user. |

* + - 1. **Pay for order**

|  |  |
| --- | --- |
| **Name** | **Pay for order** |
| **Scenario** | System needs to verify user email |
| **Description** | This use case occurs when the user signs up for account or access forgot password function, verify the information. After confirm process, activate account and redirect to home page of system. |
| **Actor** | Staff (sale staff, inventory staff, manager), customer |
| **Trigger** | User click login, enter email and password then submit. |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website and have signed up or accessed forgot password. |
| **Post-condition** | **Success:** User’s email is verified for account.  **Fail:** Show notification and captcha verification before authenticating again |
| **Flow of events** | 1. User have signed up or accessed to forgot password function.  2. System send verification link to user email  3. User open mailbox and click verification link  4. User email or user identity is verified |
| **Exception conditions** | If the entered email is invalid, system will give notify to ask them try again.  If the entered email and password don’t exist in system, system will give notify to ask them try again. |

* + - 1. **Verify Payment**

|  |  |
| --- | --- |
| **Name** | **Confirm information via email** |
| **Scenario** | System needs to verify user email |
| **Description** | This use case occurs when the user has chosen product and want to purchase it, this process verifies the user information, product information, payment information. After this process, if succussed, order will be created an involve. |
| **Actor** | Customer |
| **Trigger** | Users click “buy now” button on product screen or cart screen |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website and have signed into system. Have chosen at least 1 product |
| **Post-condition** | **Success:** System created an involve corresponding to an order.  **Fail:** Show notification and captcha verification before start pay for order process again |
| **Flow of events** | 1. User have signed into system.  2. Click “buy now” in product screen or cart screen  3. Input shipping information and method  4. Select payment method and click purchase  5. Redirect to accepted payment method  4. Show the payment result |
| **Exception conditions** | If the payment methos is not available, show notification dialog  If the third-party payment service return fails result, show notification dialog. |

* + - 1. **Compare Product**

|  |  |
| --- | --- |
| **Name** | **Add/remove to/from Cart** |
| **Scenario** | Users want to compare between different product |
| **Description** | This use case occurs when the user want to compare between different product to find out what product is suitable for them to buy |
| **Actor** | Customer |
| **Trigger** | Users click “Compare” at specific product |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website. |
| **Post-condition** | **Success:** Products comparation screen is showed  **Fail:** Show notification to notify them. |
| **Flow of events** | 1. User access to product page or cart page  2. Click to “compare” icon button on specific page  3. Product comparation screen is showed |
| **Exception conditions** | If the product is not available, show notification to notify user. |

* + - 1. **CRUD Product**

|  |  |
| --- | --- |
| **Name** | **CRUD Product** |
| **Scenario** | Staff users want to CRUD Product for customer to buy. |
| **Description** | This use case occurs when the user wants to CRUD Product for customer to buy. After CRUD Product process, the product is created and show to customer |
| **Actor** | Admin |
| **Trigger** | User access to product management screen.  Users click “Create/Update/Delete” button on purchase bill management screen. |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website and have signed into system. Having access to product management screen. |
| **Post-condition** | **Success:** Product is created and ready to be showed to customer  **Fail:** Show notification dialog to notify user |
| **Flow of events** | 1. User have signed into system  2. User access to product management screen management screen and click Create/Update/Delete” button.  3. System show product information form  4. User input the information.  5. Product is created and ready to be showed to customer. |
| **Exception conditions** | If user input is not valid, show notification dialog to notify them.  If product has been existed, show notification dialog to notify them. |

* + - 1. **CRUD Promotion**

|  |  |
| --- | --- |
| **Name** | **CRUD Promotion** |
| **Scenario** | Staff users want to CRUD Promotion event to push their sales. |
| **Description** | This use case occurs when the user wants to CRUD Promotion event to push their sales. After CRUD promotion process, their promotion will be applied to their selected product and customer user. |
| **Actor** | Admin |
| **Trigger** | User access to promotion management screen.  Users click “Create/Update/Delete” button on promotion management screen. |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website and have signed into system. Having access to promotion management screen. |
| **Post-condition** | **Success:** Promotion event is created and applied to selected product, selected group of customers.  **Fail:** Show notification dialog to notify user |
| **Flow of events** | 1. User have signed into system  2. User access to promotion management page and click Create/Update/Delete” button.  3. System show promotion information form  4. User input their information.  5. Promotion event is recorded. |
| **Exception conditions** | If user input is not valid, show notification dialog to notify them.  If promotion have been existed, show notification dialog to notify them. |

* + - 1. **CRUD Order**

|  |  |
| --- | --- |
| **Name** | **CRUD Order** |
| **Scenario** | Staff users want to CRUD Order for offline order or online order. |
| **Description** | This use case occurs when the user wants to CRUD Order offline order or online order. After CRUD Order (involve) process, the purchase bill for corresponding order is created. |
| **Actor** | Admin |
| **Trigger** | User access to purchase bill management screen.  Users click “Create/Update/Delete” button on purchase bill management screen. |
| **Related use cases** | Null |
| **Pre-condition** | The user has accessed the system's website and have signed into system. Having access to purchase bill management screen. |
| **Post-condition** | **Success:** Purchase bill is created and ready to be printed  **Fail:** Show notification dialog to notify user |
| **Flow of events** | 1. User have signed into system  2. User access to purchase order management screen management screen and click Create/Update/Delete” button.  3. System show order information form  4. User input the information.  5. Purchase bill is created and ready to be printed. |
| **Exception conditions** | If user input is not valid, show notification dialog to notify them.  If recommendation product has been existed, show notification dialog to notify them. |

* 1. **Actor description**

|  |  |  |
| --- | --- | --- |
| **No.** | **Actor** | **Description** |
| **1** | Admin | The most powerful user in the system, having access to all functions in the system and full access to CRUD functions |
| **5** | Customer | Users access the system to make purchases and purchase-related operations. |

* 1. **Context diagram**

Graphical user interface, application

Description automatically generated

**Diagram 9:** Context - System Overview

* 1. **Domain model**Diagram

     Description automatically generated

**Diagram 10:** Domain - User

**Diagram

Description automatically generated**

**Diagram 11:** Domain - Supply Partner Management

**Diagram

Description automatically generatedDiagram

Description automatically generated**

**Diagram 13**: Domain - Branch Store Management

**Diagram 12:** Domain - Inventory Management

**Diagram

Description automatically generated**

**Diagram 14**: Domain - Customer Management

**Diagram

Description automatically generated**

**Diagram 15:** Domain - Staff Management

Diagram

Description automatically generated

**Diagram 16:** Domain – Product sale – after-sale Management

* 1. **Workflow**
     1. **Sequence diagram**

**Note:**

* User must login before access any function of system
* **Diagram

  Description automatically generated**The functions of the system are decentralized according to the type of account

**Diagram 17:** Sequence - CRUD

**Diagram

Description automatically generated**

**Diagram 18:** Sequence - After-sale

**Diagram, schematic

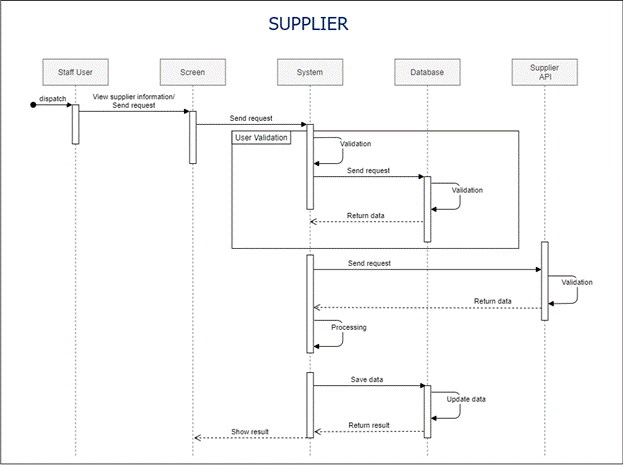
Description automatically generated**

**Diagram 19:** Sequence - Purchase & Payment

**Diagram, schematic

Description automatically generated**

**Diagram 20:** Sequence – Branch Store



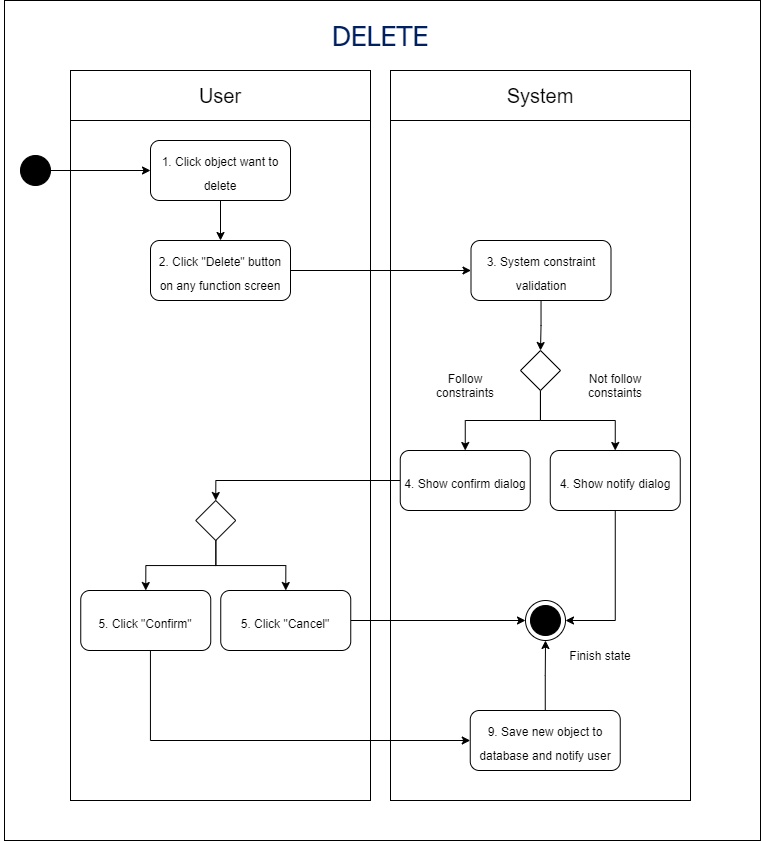
**Diagram 21:** Sequence – Branch Store

* + 1. **State diagram**

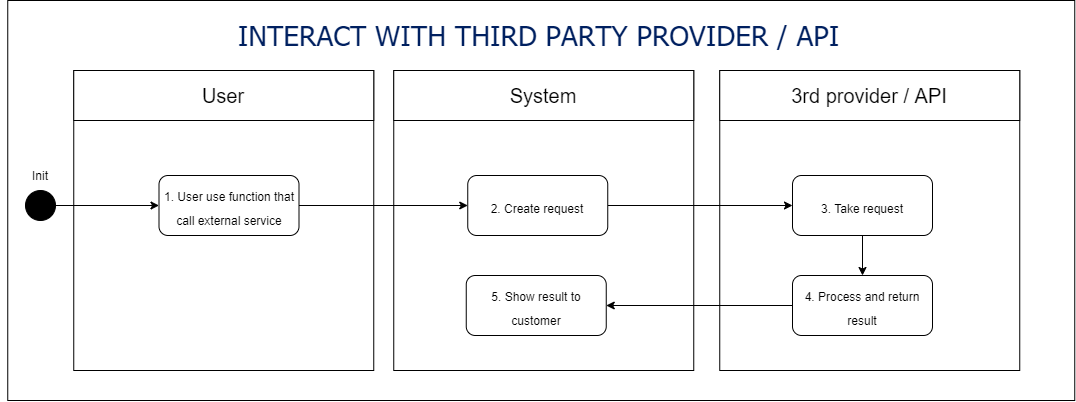
**Diagram

Description automatically generated**

**Diagram 22:** State - Create/Update

****

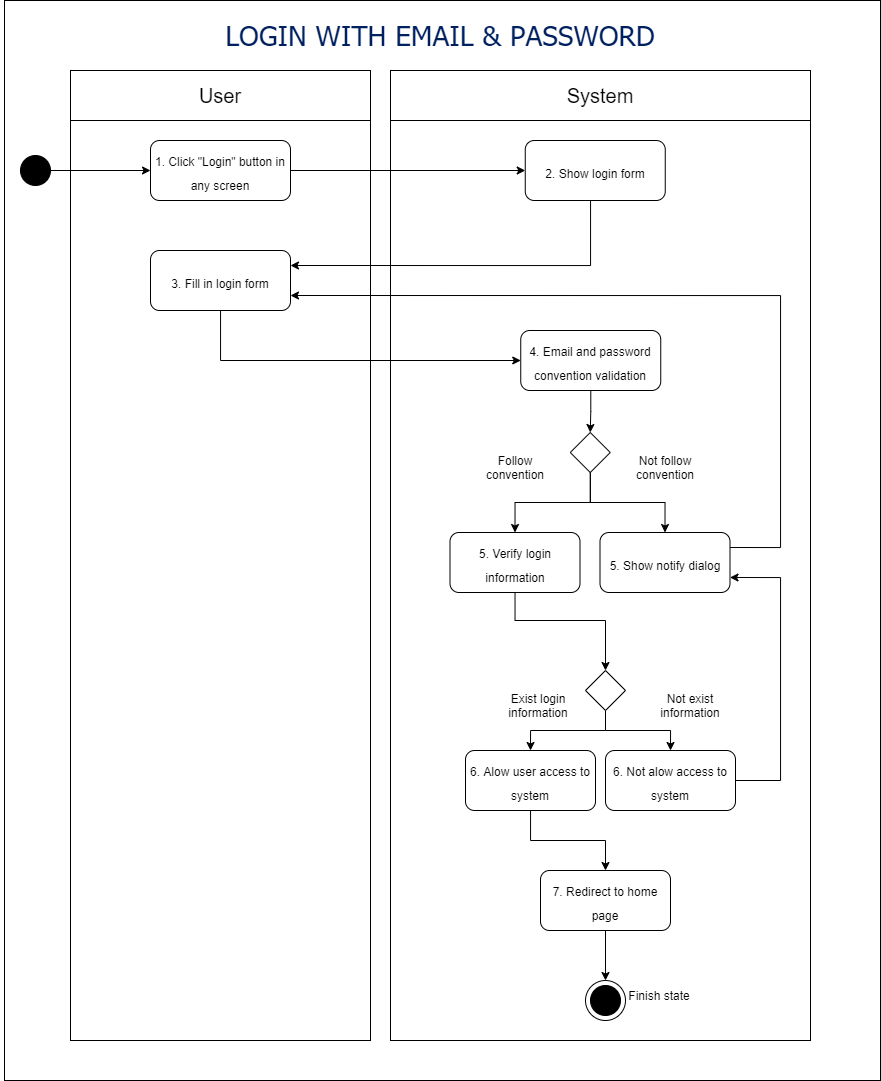
**Diagram 23:** State - Delete

**Diagram

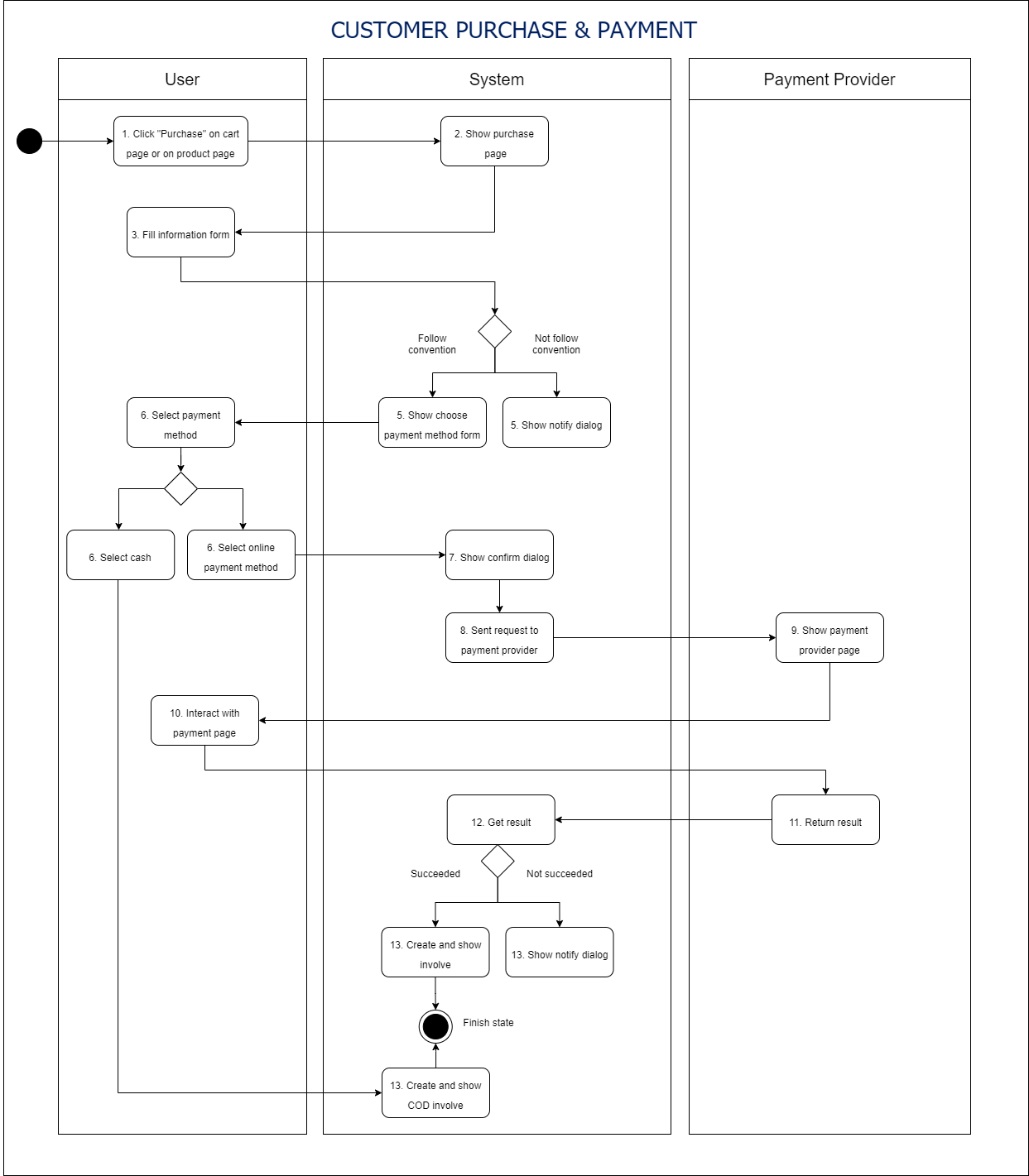
Description automatically generated**

**Diagram 25:** State - Interact with Third Party Provider / API

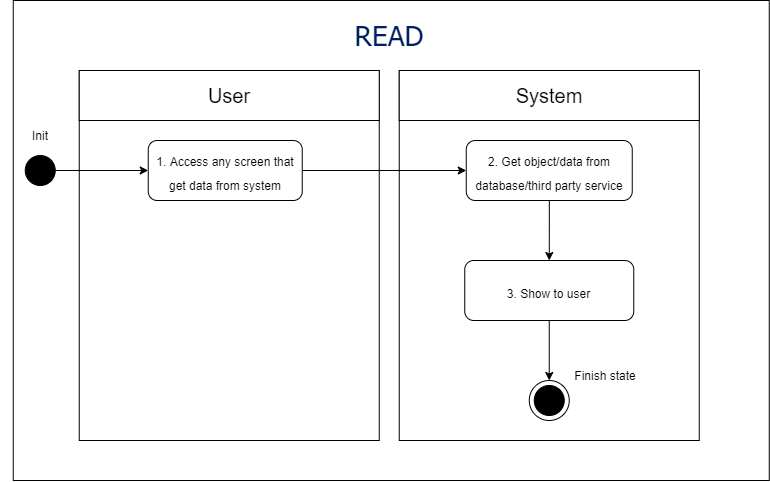
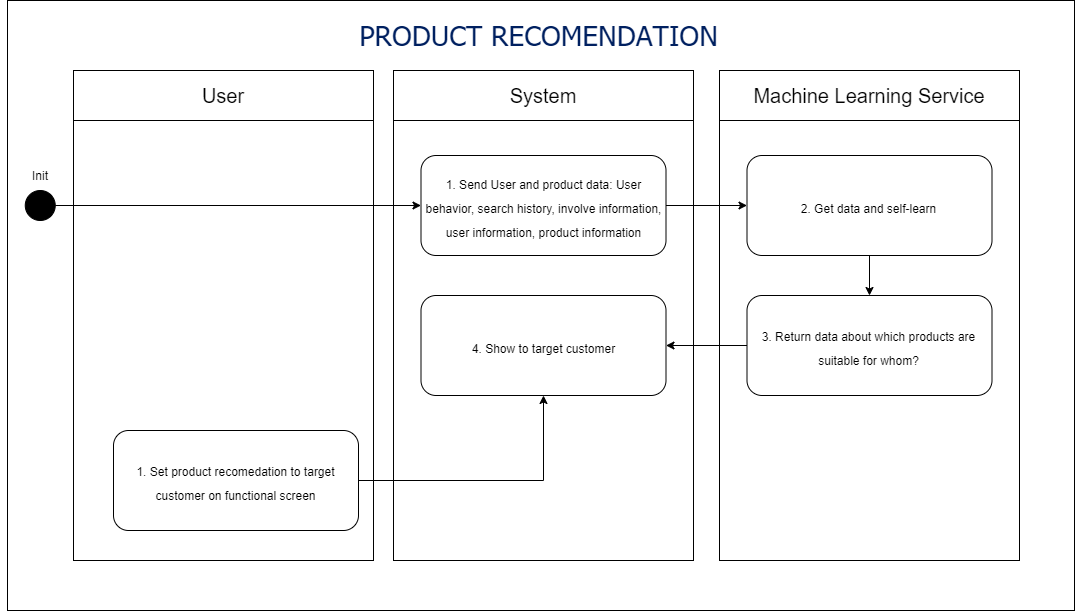
**Diagram 24:** State - Login with Social Network Provider

****

**Diagram 26:** State - Login with Email & Password

****

**Diagram 27:** State - Customer Purchase & Payment

****

**Diagram 29:** State - Product Recommendation

**Diagram 28:** State - Read

**Diagram

Description automatically generated**

**Diagram 30:** State - Sign Up

* 1. **Business rules**

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| **(1)** | BR1 | **User authentication:**   * **Sign up.**   **Customer:**   * When users click on “register” for an account, redirect to the **register page**, they need to provide basic information such as name, email, and password in register form as well as they can register via Google or Facebook login services method provided by third party provider. * The register process goes like description below:   + Checking **[User email]**, if **[User email]** in right form, continue checking **[User password]** form, Else, show **“MSG1”** dialog to notify user.   + Checking **[User password]**, if **[User password length]** > 8 and **[User password]** does not contains special character, space, continue the process, else, show **“MSG1”** dialog to notify user.   + Checking **[User email]** provided via email text field or social authentication service provider.   + If **[User email]** have existed in system:   Show **“MSG2”** dialog and redirect to **sign in page** if they click “no” else stay on **register page**.   * + Else:   Save inputted information or information from social authentication service provider to system.  Show email verification code text field and send verify code to **[User email]** with template below:   |  |  | | --- | --- | | Subject | Email verification for **[Store name]**‘s customer register | | Body | Dear **[User name],**  We have received your register account request. You only need one more step to activate for account. Fill the code below to required email verification code on register page:  **[Random Generated Code]**  Regard,  [Store name] team. |  * + Checking **[Email Verification Code]**, if **[Email Verification Code] = [Random Generated Code],** notify to user register process succeed, save **[User]** to system, and redirect to **home page**, else, show **“MSG3”** dialog.   **Sale staff, inventory staff, manager:**   * Sale staff, inventory staff, manager can’t create account by themselves. * System’s admin user will create their account on staff management page by give all information about staff such as name, email, account type, gender, birthday, and other background information. **[User email]** and **[User password]** will be check like customer process. * System will send and email to user to activate account like template below:  |  |  | | --- | --- | | Subject | Account activation for [Staff position] position in **[Store name]** | | Body | Dear **[User name],**  We have created for you an account for **[Staff position]** position in **[Store name]**, you will use this account to access our system for your work. Please check your information below:   * Name: **[User name]** * Age: **[User age]** * Position: **[User position]** * Address: **[User address]** * Phone number: **[User phone number]**   Please click the link below to confirm your information and activate your account.  **[Generated Link]**  If something wrong, please contact us by reply to this email.  Regard,  **[Store name]** team. |  * Account only can be used after user click [Generated Link] to confirm their information. * **Log in:** * When user click “Log in”, redirect to **login page.** * User can login with email and password, or with Google, Facebook authentication services. * The login process goes like description below:   + Checking **[User email]** provided via email text field or social authentication service provider.   + If **[User email]** have existed in system:   Checking **[User email]** and **[User password]**, or **[User Information]** provided by third party, if it match with system information, redirect to home page corresponding to each type of account, else, show **“MSG4”** dialog to notify user.   * + Else show **“MSG5”** dialog to notify user if user click “yes” redirect to **sign up page**, else, stay on current page. |
| **(2)** | BR2 | **CRUD:**   * All CRUD function will run **[User validation]** process. After that checking **[User Input]** and **[Object Constraint].** Process only submitted when all validation above passed. * CRUD function will show **“MSG6”** dialog to user to confirm. * CRUD process after finished will save timestamp and information about user who start CRUD function. * Object that be updated or deleted will move to stored database and can be recovered in 30 days.   **Create:**   * Object will have unique **[ID]** and can’t be changed in future. * If there are many create requests to system at the same time, system will randomly choose a request to process and, another request will randomly put in queue. * **Create process:** * User click **“Create”** button on specific functional screen. * Show object information form to fill in. * User click “Create” on object information screen, system will check everything user have filled in:   If all of it follow the required form constraints and system constraints, show confirmation notify dialog **“MSG6”** to confirm request.  If user click “Yes”, start saving new object process, else, turn off dialog and do nothing.  Else, show **“MSG1”** dialog to user to correct.  **Read:**   * Data return to user will be automatically refresh when have changed in database to make sure show to latest data. * If something when wrong with refresh data, show timestamp of data snapshot to user, and prevent user start create, update, delete function. * **Read process:** * User access to any function that contain data from database. * System try to get data from database and show to user, if anything went wrong, show timestamp on screen for latest local data and notify to user via **“MSG7”** dialog.   **Update:**   * If there are many update requests to system at the same object and at the same time. Update requests will be cancelled and notify to users. * **Update process:** * User click **“Update”** button on specific functional screen. * Show object information form to fill in. * User click “Update” on object information screen, system will check everything user have filled in:   If all of it follow the required form constraints and system constraints, show confirmation notify dialog **“MSG6”** to confirm request.  If user click “Yes”, start saving new object process, else, turn off dialog and do nothing.  Else, show **“MSG1”** dialog to user to correct.  **Delete:**   * If there are many delete requests to system at the same object and at the same time. Delete requests will be cancelled and notify to users. * **Delete process:** * User click **“Delete”** button on specific functional screen. * If all of it follow the required form constraints and system constraints, show confirmation notify dialog **“MSG6”** to confirm request.   If user click “Yes”, start saving new object process, else, turn off dialog and do nothing. |
| **(3)** | BR3 | **Statistic & Report:**   * This function will statistics data in the system according to a number of pre-selected areas from the user. * Several types of graphs can be created to represent data visually. * Users can export data in spreadsheet, text, pdf formats. * **Statistic & Report process:**   + User click **“Static & Report”** button on specific functional screen.   + Show list of filed, area, period that can be generate statistic & report to user to choose.   + Show confirmation dialog **“MSG6”** if user click “Yes”, continue process, else, turn of dialog and do nothing.   + System get data from database, processing data and show to user.   + User can click **“Print”** button to print report.   + User can click **“Save”** and choose format to save report on local. |
| **(4)** | BR4 | **Purchase & Payment:**   * When customer user has chosen product and want to purchase, they click “Buy now”on **product page,** or “Purchase” on **cart page**, they will jump to Purchase & Payment process**.** * **Purchase & Payment process:**   + Get all **[Product Information], [User Information]** from database, calculate the cost.   + User choose payment method and delivery method. If they chose online payment method:   Checking availability of **payment provider API:**  If it’s available:  Send payment request and continue payment process on payment provider page.  Get return of result from payment provider API. If it’s succeeded create involve, shipping order and update store database, else, show “MSG8” dialog.  Else, show “MSG9” dialog.  Else if user chose “Cash on delivery” method:  Checking availability of **shipping provider API:**  If it’s available:  Send create order request.  Get return of result from shipping provider API. If it’s succeeded, keep tracking status of good until it paid to succeeded create involve and update store database.  Else, show “MSG9” dialog. |
| **(5)** | BR5 | **Shipping:**   * All product in paid involve or COD involve will be sent to shipping partner to ship to customer. * Shipping partner will update status of order via shipping API in real-time for customer and staff. * With COD involve, when shipping partner API notify delivery succeeded, system will create paid involve and update store database. |
| **(6)** | BR6 | **After-sale:**   * User have bought product can access after sale function for product such as: product feedback, warranty request, support request, return product request. * **Rate & give feedback process:** * User choose product, checking **[User Information],** if they have bought that product allow them access rate & give feedback function. * When user click “Rate & give feedback” button. Function panel appear. * User set rate star and fill in feedback text filed. * User click “Post” button, system will check constraint, if pass update database, else show “MSG1” dialog * **Send request & contact for support:**   + User click “Support” button   + System will check availability of store staff, if there is any staff online, initial connection to this staff and customer, else, show “MSG10” dialog.   + Message send to staff will include product from page that customer call request function.   + Staff and customer can communicate, and staff will solve customer problem via this function   + If customer exit before confirmation of staff, show “MSG6” dialog.   + Every information in this section will be recorded and send to customer email.   + When the problem have solved, staff click “Solved” button to mark this request have solved.   + System update database and related information |
| **(7)** | BR7 | **Supplier:**   * Store can send request to supplier via supplier partner API.   **Supplier request process:**   * + User access supplier page   Checking availability of **supplier provider API:**  If it’s available:  Send request and wait for result.  Get result data and show to user   * Else, show “MSG11” dialog. * **Supplier request process:**   + User click “Request” button on supplier information page.   Checking availability of **supplier provider API:**  If it’s available:  Send supplierrequest and wait for result.  If request have solved, update store database, else, mark request as pending.  Else, show “MSG11” dialog. |
| **(8)** | BR8 | **Branch store:**   * Store can send request to supplier via supplier partner API.   **Branch store view information process:**   * + User access supplier page   Checking availability of **supplier provider API:**  If it’s available:  Send request and wait for result.  Get result data and show to user   * Else, show **“MSG11”** dialog. * **Branch store request process:**   + User click “Request” button on supplier information page.   Checking availability of **supplier provider API:**  If it’s available:  Send supplierrequest and wait for result.  If request have solved, update store database, else, mark request as pending.  Else, show **“MSG11”** dialog. |
| **(9)** | BR9 | **Input convention (in regex):**   * **Email :**   ^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$   * **Password - minimum eight characters, at least one letter, one number and one special character:**   ^(?=.\*[A-Za-z])(?=.\*\d)(?=.\*[@$!%\*#?&])[A-Za-\d@$!%\*#?&]{8,}$   * **Text filed:**   ([A-Z])\w+   * **Random verification code:**   [\w\d]{6}   * **Phone number:**   ^0[\d]{9} |
| **(10)** | BR10 | **Product recommendation:**   * System sends **[User information], [User Behavior Data]** (search history, time spend on specific product, click map), **[Product Information]** to third party service machine learning. * Third party service will process and return dataset about what product will be potential for specific group of users. * Staff can manually create product recommendation to specific group of users. |

1. **Non-functional requirement**
   1. **Security matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function** | **Customer** | **Sale staff** | **Inventory staff** | **Manager** | **System’s admin** |
| **Product & sale management** |  |  |  |  |  |
| Find product | **x** | **x** | **x** | **x** | **x** |
| Pay for order | **x** |  |  |  |  |
| Choose payment method | **x** |  |  |  |  |
| Verify payment | **x** |  |  |  |  |
| Rate product | **x** |  |  |  |  |
| Add/remove to/from cart | **x** |  |  |  |  |
| Request support | **x** |  |  |  |  |
| Give feedback | **x** |  |  |  |  |
| CRUD Promotion |  |  |  | **x** | **x** |
| CRUD Recommendation Product |  |  |  | **x** | **x** |
| CRUD Purchase Bill |  | **x\*** |  | **x** | **x** |
| Send message to customer |  | **x** |  | **x** | **x** |
| CRUD Report |  | **x\*** |  | **x** | **x** |
| **Inventory management** |  |  | **x** | **x** | **x** |
| CRUD Report |  |  | **x\*** | **x** | **x** |
| CRUD Contract |  |  | **x\*** | **x** | **x** |
| **Shipping management** |  | **x** | **x** | **x** | **x** |
| CRUD Shipping Order |  | **x** | **x** | **x** | **x** |
| Sent request to shipping partner |  | **x** | **x** | **x** | **x** |
| View shipping information | **x** | **x** | **x** | **x** | **x** |
| CRUD Report |  | **x** | **x** | **x** | **x** |
| **Branch store management** |  |  |  | **x** | **x** |
| View branch store information |  |  |  | **x** | **x** |
| Send request |  |  |  | **x** | **x** |
| **User management** |  |  |  |  | **x** |
| Sign up | **x** |  |  |  | **x** |
| Sign in | **x** | **x** | **x** | **x** | **x** |
| Authentication | **x** | **x** | **x** | **x** | **x** |
| Create staff account |  |  |  |  | **x** |
| **Supply partner management** |  |  |  | **x** | **x** |
| CRUD Supplier |  |  |  | **x** | **x** |
| Request import product |  |  |  | **x** | **x** |
| CRUD Report |  |  |  | **x** | **x** |
| **Finance & administrative management** |  |  |  | **x** | **x** |
| CRUD Report |  |  |  | **x** | **x** |
| CRUD Policy document |  |  |  | **x** | **x** |

**Note:**

* x: Full permission to perform action.
* x\*: Update, delete permission on object that user created only.
  1. **Performance requirement**

**Number of users:**

* Number of concurrent users: 1000-2000 users.
* Number of business users: 300 users.

**Data volume:**

* Number of document: 10/file size.
* Data growth rate: 100% per year.

**Availability:**

* 24/7 in at least 99.01%

**Usage frequency:**

* Hourly
  1. **Implementation requirement**

**Location:**

* This system will implement in Vietnam.
* Database service: Azure.

**Maintain frequency:**

* + 2-4 times / a month

**Maintain timeframe:**

* + From 12 am to 3 am.

1. **Other Requirement** 
   1. **Archive function**

|  |  |  |
| --- | --- | --- |
| **List name** | **Actor** | **Condition** |
| Product | Customer user, staff user, admin user | Actor can achieve detail information of product if product is visible (not hidden) to them. |
| Order | Customer user, staff user, admin user | Customer can achieve detail information of their order.  Staff & admin can achieve detail information of customer order with order ID. |
| Contract, report | Manager user | Manager user can achieve detail information about C report exist in system. |

* 1. **Security Function**

**Authorization:** Determine what type of actor the user is and assign the corresponding permissions to the user. Some objects and functions can only be accessed by a specific actor type.

**User Identifier:** Every user must verify via the verification code sent to the email. Customer accounts or staff accounts created by admin users both need to be activated via email to identify users and avoid account spam.

**System logs:** All CRUD operations are recorded in Logs so that users can look up information. Logs are stored in the system for 6 months and automatically deleted if there is no storage operation. Besides, all errors arising in the system (if any) are stored in logs for future repair and upgrade.

**Anomaly detection and self-protection mechanism:** When the system has abnormal operations, the system can automatically notify the management user or the admin user. There are auto-break or auto-correct mechanisms for pre-scripted operations

1. **System Requirement**
   1. **Permission level**

|  |  |  |
| --- | --- | --- |
| **Permission level** | **Permission** | **Description** |
| Full control | Full access permission to system. | Actor can access all system functions as well as all objects in the database |
| Partial restriction | Partial access permission to system. Detail in 3.1. Security Matrix | Part of the system's functions can be accessed, decentralized according to the user's role in the store. |
| Read only | Read permission | Actors can only view data in authorized parts of the system |
| Customer | Permission related to the customer's purchasing process | Customers can access an area in the system dedicated to purchase product. Customers cannot access the management areas of the system |

* 1. **Actor group**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Permission Level** | **Description** |
| System’s Admin | Full control | The user has the highest store management role. |
| Staff | Partial restriction  Read only | For functions directly related to the user's role. Actors can have partial permissions in operations.  For functions that are not directly related to the user's role, the actor can only view information or cannot. |
| Customer | Partial restriction | These actors can access the sales area of the system to make purchases. Unable to access the management area. |

* 1. **User interface**

|  |  |
| --- | --- |
| **UI Name** | **Description** |
| Full control | The interface displays all the functions available in the system to the user. |
| Partial restriction | The interface hides some components, disable some functions. Some system areas are inaccessible. |
| Read only | The interface only displays information, other functions are hidden. |
| Customer | The user interface is separate from the management interface. |

* 1. **Technical concern**
* System overload during peak hours.
* Third party services are not available.
* Growth rate too high.
* Internet connection speed
* Simultaneous operations on an object

1. **Installation**

In actual deployment, the environment should be set as follows:

* XAMPP
* Angular
* PHP My Admin

1. **Development Direction & Improvement**

* Improved user experience in terms of response time, error rate.
* Edit and update functions to suit reality after putting the product into practice.
* Expanding the functions of product recommendation, after-sale customer care, promotion,...
* Link with social networking platforms to support sales and purchases easily.
* Develop a mobile interface that is user-friendly and can be accessed and interacted quickly.

1. **Conclusion**

Over the course of the project, the team gained a better understanding of the software development, operation, maintenance and update process.

To develop a software is not only code and implementation but includes many steps after that to ensure the software is operated correctly and in accordance with reality.

1. **Appendixes**
   1. **Term**

|  |  |  |
| --- | --- | --- |
| **No.** | **Term/Appendixes** | **Description** |
| 1 | Authentication | Verify user information before being authorized to use the system. |
| 2 | Authorization | The action or process of authorizing or being authorized. Decentralize functions corresponding to the actor's role |
| 3 | Logs | The log file stores the events in the system. |
| 4 | Helper | A helper class is used to assist in providing some functionality |
| 5 | Third party services | An external software or services provides a service or technology |
| 6 | Partner API | Application Programming Interface provided by partner |
| 7 | MSG | Message |
| 8 | CRUD | Create – Read – Update - Delete |

* 1. **Message**

|  |  |  |
| --- | --- | --- |
| **Dialog** | **Dialog message** | **Button** |
| MSG1 | Some flied below are not in right format:  **[Fields name]** | Ok |
| MSG2 | User with **[User email]** email address has existed in system. Do you want to login instead? | Yes/No |
| MSG3 | Your filled-in email verification code is wrong, please try again! | Ok |
| MSG4 | Your filled-in authentication information is wrong, please try again! | Ok |
| MSG5 | User with **[User filled-in email]** email address has not existed in system. Do you want to sign-up instead? | Ok |
| MSG6 | Are you sure to perform this **[Function name]** action? | Yes/No |
| MSG7 | Something went wrong in system. Data is not synchrony now. This data was synchronized at **[Timestamp]** | Ok |
| MSG8 | Payment failed because [Error Message]. Please try again. | Ok |
| MSG9 | **[Payment method name]** payment services are not available now. Please try again or try other method! | Ok |
| MSG10 | Sorry, there is no online support staff available right now, please try again later or send email to us via **[Store email]**! | Ok |
| MSG11 | **[API Service Name]** services are not available now. Please try again later! | Ok |

* 1. **Issue list**

|  |  |  |
| --- | --- | --- |
| **No.** | **Issue** | **Description** |
| 1 | System overload | Several emergencies runs simultaneously in progress which exceed the number of available ambulances to respond, resulting in a delay in dispatch and/or response of an ambulance. |
| 2 | Page not found | Function redirect to not available page |
| 3 | Crash | There was an error in the system that resulted in a wrong response and caused the system to stop suddenly on the user interface |
| 4 | Not response | System does not return data information. |
| 5 | Low resolution element | Slow network speed, low quality of uploaded images, low user hardware support... the above factors lead to low-resolution user interfaces. |
| 6 | Long loading time | Low internet speed, data is too heavy, user's hardware processing speed is low... the above factors cause long loading time |
| 7 | Wrong information | Incorrect processing system can cause information falsification |