**DUYTAN UNIVERSITY**



**INTERNATIONAL SCHOOL**

**--------------🙖🙐✰🙖🙐-------------**



**CMU-SE 100**

**INTRODUCTION TO SOFTWARE ENGINEERING**

**SYNTHESIS REPORT GROUP PROJECT**



**Mentor: Msc. Huy Nguyen Dang Quang**

***Team Member*:**

**Cường, Huỳnh Ngọc**

**Hoàng, Lê Hữu**

**Hào, Thái Nhật**

**Hiệp, Ngô Xuân**

**Duy,Thái Nhật**

**Group Project - Mentor:**

Name Signature Date

Da nang, 12/2023

|  |  |
| --- | --- |
| **Project Acronym** | FMS |
| **Project Title** | **A self-service coffee shop system**. | | | |
| **Start Date** | 10-Dec-2023 | **End Date** | 25-Dec-2023 | |
| **Lead Institution** | International School, Duy Tan University | | | |
| **Team Member** | **Name** | **Email** | | **Phone** |
|  | Cuong,Huynh Ngoc | huynhngoccuong12345678@gmail.com | | 0702333936 |
|  | Hao, Thai Nhat | haothai0701@gmail.com | | 0932590857 |
|  | Duy, Thai Nhat | thainhatduy2k5@gmai.com | | 0905418349 |
|  | Hoang, Le Huu | Lehuuhoang.hoanghoang@gmail.com | | 0337631023 |
|  | Hiep, Ngo Xuan | hiepxuan07@gmail.com | | 0326627614 |

**DOCUMENT NAME**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Title** | **Synthesis Report Group Project Document** | | |
| **Reporting Period** | 05 – Sep – 2022 | | |
| **Author(s)** | Team 5-–Group Project CMU-SE 100 | | |
| **Role** |  | | |
| **Date** |  | **Filename** | Team 5-–Group Project CMU-SE 100 |

**SIGNATURE**

***Document Approvals:*** *The following signatures are required for approval of this document.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mentor** | Huy, Nguyen Dang Quang | **Signature:** |  |
| **Date:** |  |
| **Team Lead** | Cuong, Huynh Ngoc | **Signature:** |  |
| **Date:** |  |
| **Team member(s)** |  | **Signature:** |  |
| **Date:** |  |
| Hao, Thai Nhat | **Signature:** |  |
| **Date:** |  |
| Duy, Thai Nhat | **Signature:** |  |
| **Date:** |  |
| Hoang, Le Huu | **Signature:** |  |
| **Date:** |  |
| Hiep, Ngo Xuan | **Signature:** |  |
| **Date:** |  |

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| **1.0** |  | **Make Synthesis Report Group Project Document** |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

Mục lục

[1. INTRODUCTION 6](#_Toc154356136)

[2. DESCRIBE BUSINESS PROCESSES OF RESTAURANT 6](#_Toc154356137)

[3. LIST OF PARTICIPANTS TO REQUIREMENTS GATHERING 6](#_Toc154356138)

[4. THE TECHNICALS USING TO REQUIREMENTS GATHERING 6](#_Toc154356139)

[5. ASSIGNMENT TEAM MEMBER TO REQUIREMENTS GATHERING 7](#_Toc154356140)

[6. INTRODUCTION PROJECT 7](#_Toc154356141)

[6.1. Project Objectives 7](#_Toc154356142)

[6.2. Project Scope 8](#_Toc154356143)

[6.3 Identify the Functional Requirement & Non-Functional Requirement 8](#_Toc154356145)

[7. THE USER REQUIREMENT ANALYSIS 10](#_Toc154356147)

[7.1 Business Function Diagram 10](#_Toc154356148)

[7.2 Context Model 10](#_Toc154356149)

[7.3 List of Actor 11](#_Toc154356150)

[7.4. List of Use Cases 11](#_Toc154356151)

[8. PROJECT PLAN 12](#_Toc154356152)

[9. TECHNICAL CONSTRAINTS 12](#_Toc154356153)

[9.1 Technical to develop: 12](#_Toc154356154)

[9.2 Environment: 12](#_Toc154356155)

[9.3 Another: 12](#_Toc154356156)

[10. ANALYSIS & MODELING REQUIREMENT 13](#_Toc154356157)

[10.1. Use Case Diagram 13](#_Toc154356158)

[10.2. List of Business Rule 13](#_Toc154356159)

[10.3. Activity Diagram 13](#_Toc154356160)

[11. INTERFACE DESIGN 16](#_Toc154356161)

[11.1. Introduction tool design: 20](#_Toc154356162)

[11.2 User Interface Design for ABC Function 20](#_Toc154356163)

[12. CONFIGURATION MANAGEMENT 24](#_Toc154356164)

[13. TESTING 26](#_Toc154356165)

[14. CONTRIBUTION 26](#_Toc154356166)

# 1. INTRODUCTION

- Highland coffee

- Highlands Coffee : is a famous coffee brand in Vietnam. This brand was established in 2002 and quickly became one of the leading coffee brands in Vietnam.

- Business areas : coffee and fast food.

- Address : 179 Nui Thanh, Hai Chau, Da Nang, Viet Nam.

- Phone : 0236 3530 005

# 2. DESCRIBE BUSINESS PROCESSES OF RESTAURANT

- Customers enter the establishment and scan the code on the display.

- Display the website on the customer's device.

- Customers can enter their Gmail to receive promotions and view popular menu items.

- Customers can place orders on the restaurant's homepage.

- Customers can add desired items to their shopping cart.

Top of Form

- Customers can make payments directly on their shopping cart.

Top of Form

- The system displays the payment form.

- Customers can choose to pay in cash or via bank transfer.

Top of Form

- The system notifies that the order has been successfully placed

- The system notifies about customer information

- The system notifies about the order information for the customer.

- The system sends a notification, and the customer comes to the counter to pick up their order

-

# 3. LIST OF PARTICIPANTS TO REQUIREMENTS GATHERING

*-* **Participants in collecting requirements**: Cường, Hiệp, Duy, Hào, Hoàng.

*-* ***Object to collect requirements:*** Store Management: The store manager can provide necessary information and specific requests related to the coffee shop's online selling website.

# 4. THE TECHNICALS USING TO REQUIREMENTS GATHERING

Method: Joint Application Design (JAD)

Description (mô tả):

Joint Application Design (JAD) is a requirements gathering method in which direct working sessions are organized with the active participation of key stakeholders such as customers, managers, and system users. During these working sessions, stakeholders engage in discussions, draw diagrams, and make on-the-spot decisions to clearly define project requirements.

Benefits (lợi ích):

Increased Interaction: JAD creates an opportunity for direct interaction among stakeholders. This interaction helps not only in detailing requirements but also fosters deep understanding and increased commitment from participants.

Creativity: JAD working sessions often blend creativity from various sources. New ideas can emerge when stakeholders collaborate, enhancing the quality and uniqueness of requirements.

In-Depth Understanding of Project Requirements: The direct involvement of key stakeholders ensures that everyone has a clear understanding of the project's goals and requirements. These working sessions promote consensus and shared understanding.

Quick Decision-Making: JAD allows for decisions to be made on the spot, reducing wait times and accelerating project development. Decisions are based on discussions and consensus within the team.

Reduced Misunderstandings: Due to direct participation, JAD helps minimize misunderstandings and misinterpretations between stakeholders. Everyone can work together to clarify requirements.

Establishing a Foundation for Development: JAD sessions help build a solid foundation for the subsequent development process. Participants have the opportunity to share opinions and shape the direction of the project.

Effective Communication: Direct interaction among stakeholders improves communication within the team and between project partners. JAD sessions often create an open environment for information exchange.

Overall, JAD is a flexible and powerful method, particularly effective when close collaboration and consensus are needed from all stakeholders*.*

# 5. ASSIGNMENT TEAM MEMBER TO REQUIREMENTS GATHERING

|  |  |  |
| --- | --- | --- |
|  | ***Member(s)*** | ***Tasks/Questions*** |
|  | *Cường,* | *Observe how customers in the shop approach and use the website to place orders. Observe staff guiding and serving customers* |
|  | *Hào, Duy* | *Ask the staff a question:*  *1. How convenient is ordering using the website?*  *2. Is it easy for customers to access the website?*  *3. The effectiveness of using the website to order* |
|  | *Hiệp, Hoàng* | *Ask the cashier a question:*  *1. When a customer orders food, how quickly is the order sent to the employee's account?*  *2. Can staff delete tables or move tables?*  *3. Each employee's account is different or the same account*  *4. How fast is the web access speed when the number of customers accessing is large?* |

# 6. INTRODUCTION PROJECT

6.1. Project Objectives

**Customers can easily use and place food orders directly through a QR code leading to the restaurant's website for payment and to make specific requests regarding their food ......... all on their mobile phones.**

**Employees and managers can easily view invoices and manage the number of dishes sold during the day through the management website.**

## 6.2. Project Scope

## The website project consists of various distinct functions designed to cater to customers in the process of ordering and making payments for food items through the available website prepared by the restaurant. This system aids both employees and managers in tracking products, orders, and restaurant information.

## 6.3 Identify the Functional Requirement & Non-Functional Requirement

## **a)** **High level Functional Requirement (FR) chưa làm**

|  |  |  |
| --- | --- | --- |
| FR1.1 | Title | Create New place |
| Stakeholder | Users who use jukebox for playing music |
| Description | If user is new one, the system will allow him/her establish an account and specify amount the money that he/she allocate from their credit card into Jukebox account. Otherwise, user can check Jukebox balance. |
| FR1.2 | Title | **Edit a Place’s Information** |
| Stakeholder | Users who use jukebox for playing music |
| Description | The system shall display list of songs grouped into music genres and will allow user select music genres and songs to play. Once the first song is selected the song will start playing while allowing user to make further choices. The system shall also record such selected and played song into database. User can deselect his/her songs from the current playlist and amount of deselecting shall refund. User can create account on one Jukebox and play music on another in distributed system. |

**b) List of Non-Functional Requirement:**

1. **Performance:**
   * **Response Time: The system should provide quick response times for user actions, such as placing an order or browsing the menu.**
   * **Throughput: The system should handle a certain number of transactions per minute during peak hours.**
2. **Scalability:**
   * **User Scalability: The system should be able to handle an increasing number of users as the cafe's popularity grows.**
   * **Data Scalability: The system should efficiently handle a growing database of menu items, orders, and customer information.**
3. **Availability:**
   * **Uptime: The system should have high availability, ensuring that it is operational and accessible to users most of the time.**
   * **Redundancy: Implement redundancy measures to minimize downtime in case of server failures or other issues.**
4. **Reliability:**
   * **Fault Tolerance: The system should gracefully handle errors and continue to operate without data loss.**
   * **Data Integrity: Ensure the accuracy and consistency of data throughout the system.**

**5. Security:**

* + **Data Encryption: Use encryption to secure sensitive information such as user credentials and payment details.**
  + **Access Control: Implement role-based access control to restrict unauthorized access to certain functionalities or data.**

**6. Usability:**

* + **User Interface Design: Ensure an intuitive and user-friendly interface for both customers and cafe staff.**

**7. Compatibility:**

**Cross-Browser Compatibility: Ensure the application works seamlessly across popular web browsers.**

**Mobile Responsiveness: Design the application to be responsive on various devices, including smartphones and tablets.**

1. **Network Requirements:**
   * **Bandwidth: Specify network bandwidth requirements to ensure smooth data transfer.**
   * **Latency: Minimize network latency to enhance user experience.**
2. **Environmental:**

**Deployment Environment: Specify the target environment for deployment, such as cloud infrastructure or on-premises servers.**

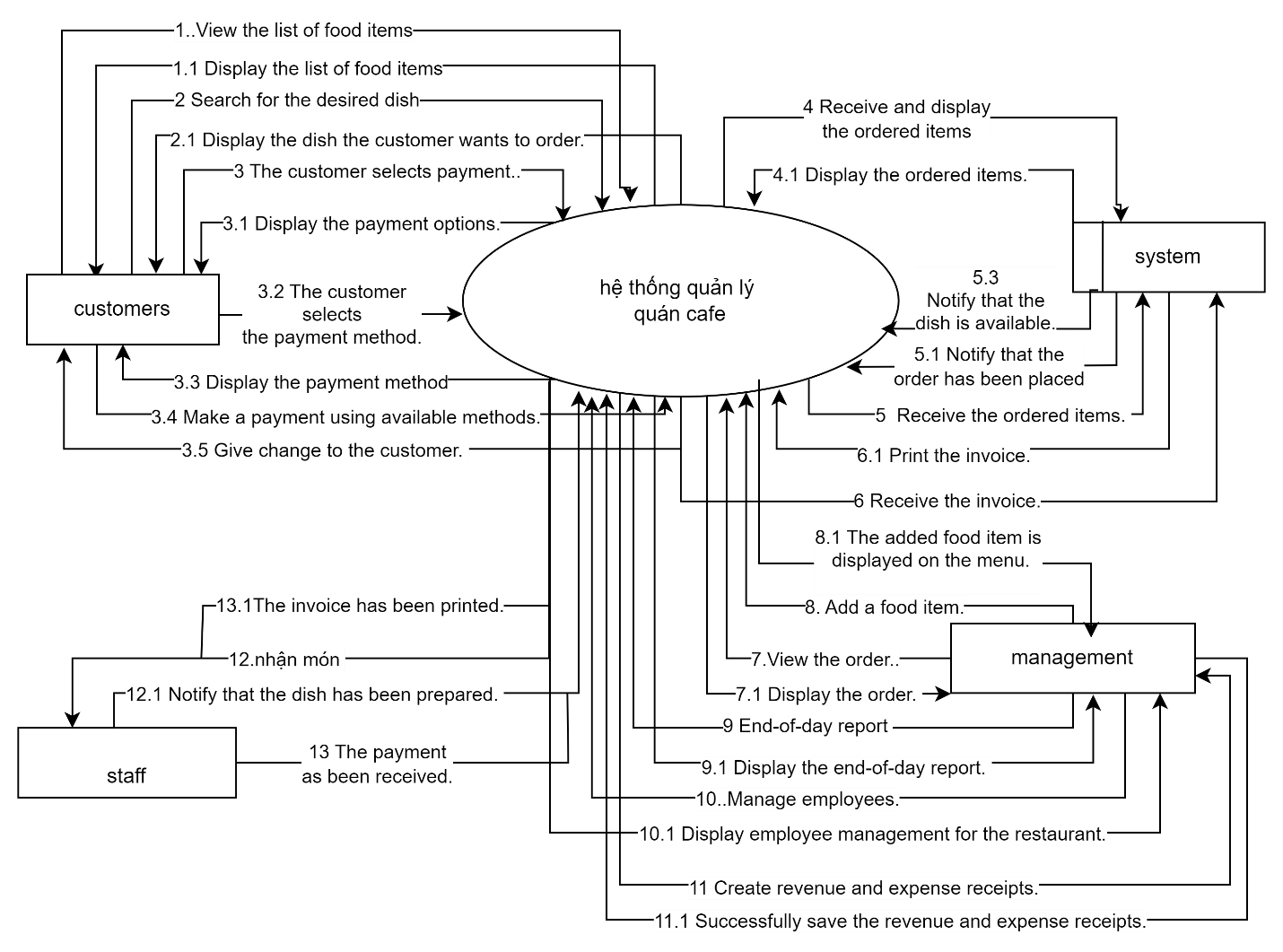
# 7. THE USER REQUIREMENT ANALYSIS

## 7.1 Business Function Diagram



## 7.2 Context Model

**a) Context Model**

****

**b) System Context Description**

|  |  |
| --- | --- |
| **Function** | **Description** |
| view menu | The feature aims to allow users, both customers, and cafe staff, to easily access and navigate the cafe's menu to explore available food and beverage items. |
| place an order | The feature is designed to facilitate the seamless process of customers selecting items from the menu, customizing their orders, and completing the transaction. This functionality aims to streamline the ordering process for both customers and cafe staff. |
| make payment | The feature facilitates the secure and seamless processing of financial transactions for confirmed orders. This functionality ensures that customers can complete their transactions with ease, while maintaining the security and integrity of their payment information. |
| receive the order | The feature is designed for cafe staff to efficiently manage and fulfill incoming orders. This functionality ensures that staff can acknowledge and process orders promptly, maintaining a smooth workflow in the cafe**.** |

|  |  |
| --- | --- |
| view invoice | The feature allows both customers and cafe staff to access and review detailed invoices for completed orders. This functionality is crucial for transparency in transactions and provides a clear breakdown of charges associated with each order. |
| Communicate with the staff | The feature enables users, both customers and cafe staff, to engage in real-time communication for inquiries, order updates, or issue resolution within the Quan's Cafe application. |
| *display menu list* | Menu display system for customers and wedding information staff |
| *input orders* | Facilitate the seamless process for customers to input their food and beverage orders, customize items, and add them to the shopping cart for checkout. |
| notify availability of orders | To keep both customers and cafe staff informed about the status of orders, ensuring transparency and efficient order fulfillment.. |
| print invoices | Ability to confirm orders have been received and are being prepared. Feature to send notifications to customers about their order status. |
| Receive the order | To streamline the process of receiving, acknowledging, and managing incoming orders, ensuring a smooth workflow for cafe staff. |
| Prepare the order | The "Prepare the Order" feature is designed to guide cafe staff through the process of preparing food and beverages for customer orders, ensuring accuracy, efficiency, and timely delivery or pickup. |
| Collect payment | The feature facilitates the secure and efficient processing of payments for customer orders within the Quan's Cafe application. |
| system management | The feature encompasses tools and functionalities to administer and maintain the overall operation, configuration, and health of the Quan's Cafe application |
| add items to the system | The "Add Items to the System" feature allows administrators to efficiently add new food and beverage items to the Quan's Cafe menu, ensuring that the system stays up-to-date with the latest offerings. |
| manage invoices | The feature allows administrators to efficiently handle and oversee the invoicing process within the Quan's Cafe application, ensuring accurate billing and proper financial management. |
| Employee management | The feature is designed to enable administrators to efficiently handle the various aspects of employee administration within the Quan's Cafe application. |
| View orders. | the "View Orders" feature provides a user interface for administrators to access and monitor customer orders within the Quan's Cafe application, facilitating effective order management and fulfillment. |
| Record revenue and expense receipts | the feature allows administrators to accurately document and manage financial transactions related to revenue generation and expenses within the Quan's Cafe application |
| Item management | The feature allows administrators to efficiently manage and maintain the menu items within the Quan's Cafe application, ensuring an up-to-date and organized menu for customers |

## 7.3 List of Actor

**1.customers**

**2.system**

**3.management**

**4.staff**

## 7.4. List of Use Cases

|  |  |  |
| --- | --- | --- |
| Use case ID | Use case name | Functional Req. |
| *UC.01* | *view menu* | *FR1.1* |
| *UC.02* | *place an order* | *FR1.2* |
| *UC.03* | *make payment* | *FR1.3* |
| *UC.04* | *receive the order* | *FR1.4* |
| *UC.05* | *view invoice* | *FR1.5* |
| *UC.06* | *Communicate with the staff* | *FR1.6* |
| *UC.07* | *display menu list* | *FR1.7* |
| *UC.08* | *input orders* | *FR1.8* |
| *UC.09* | *notify availability of orders* | *FR1.9* |
| *UC.10* | *print invoices* | *FR1.10* |
| *UC.11* | *Receive the order* | *FR1.11* |
| *UC.12* | *Prepare the order* | *FR1.12* |
| *UC.13* | *Collect payment* | *FR1.13* |
| *UC.14* | *system management* | *FR1.14* |
| *UC.15* | *add items to the system* | *FR1.15* |
| *UC.16* | *manage invoices* | *FR1.16* |
| *UC.17* | *Employee management* | *FR1.17* |
| *UC.18* | *View orders.* | *FR1.18* |
| *UC.19* | *Record revenue and expense receipts* | *FR1.19* |
| *UC.20* | *Item management* | *FR1.20* |

# 8. PROJECT PLAN

Chưa làm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Task Name** | **Duration (Days)** | **Start** | **Finish** | **Assign to** |
| **1** | **Initial** | **7** | **20/08/2022** | **26/08/2022** | **Nhan, Tuan, Huy, Huy** |
| 1.1 | Project Kick-off Meeting | 1 day | 20/08/2022 | 20/08/2022 | Nhan,Tuan,Huy,Huy |
| 1.2 | Discuss about project ideal | 1 day | 21/08/2022 | 21/08/2022 | Nhan,Tuan,Huy,Huy |
| 1.3 | Create Proposal Document | 1 day | 22/08/2022 | 22/08/2022 | Nhan,Tuan,Huy,Huy |
|  |  |  |  |  |  |

# 9. TECHNICAL CONSTRAINTS

## 9.1 Technical to develop:

* ***Languages****:*
* ***Server:***

## 9.2 Environment:

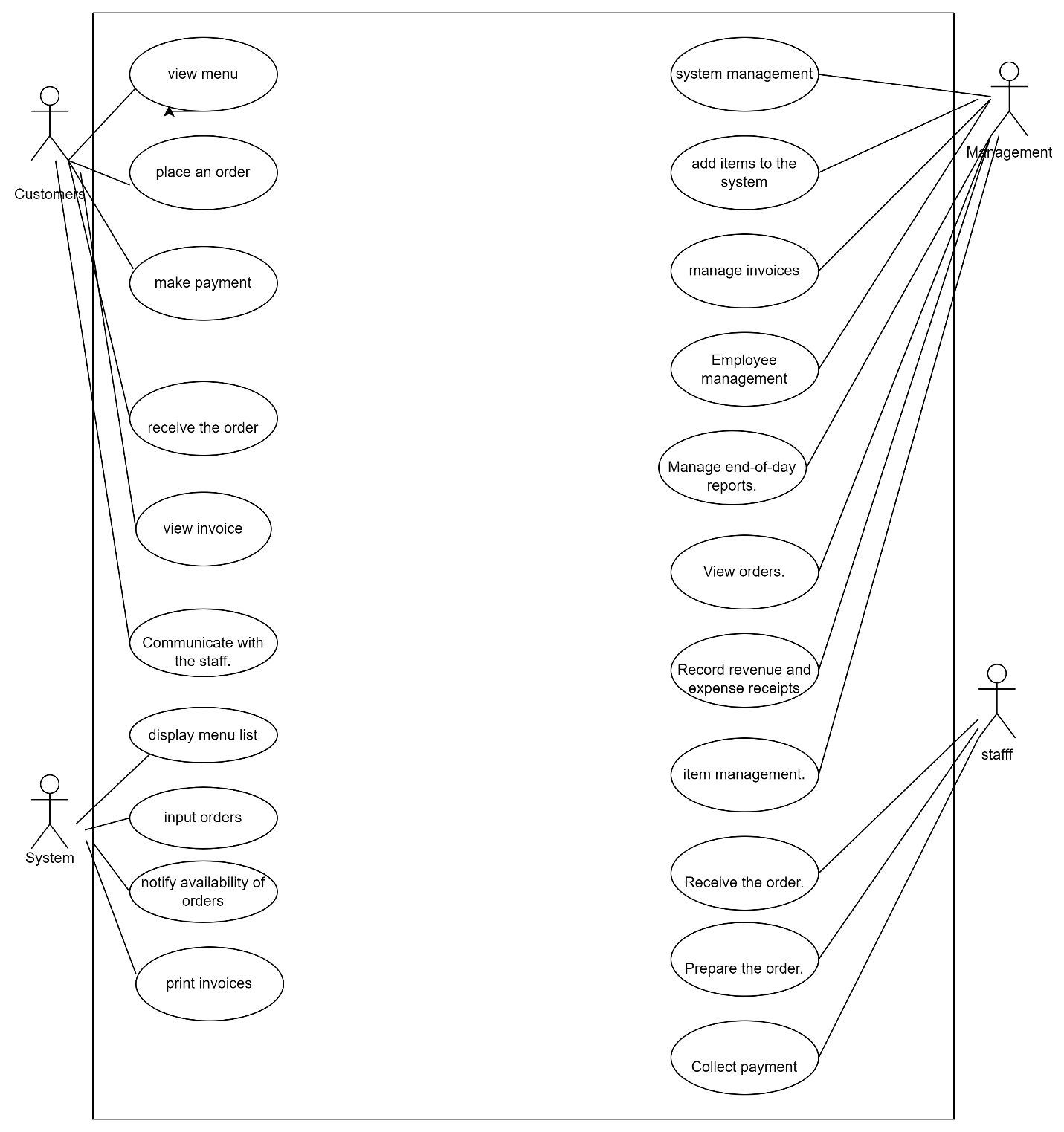
* ***Web browsers****: Google Chrome, Molliza Firefox, Apple Safari, Cốc Cốc, Microsoft Edge.*
* ***Environment:*** *Microsoft windows, Mac Os, Android, iOS,..*

## 9.3 Another:

* ***Management tool****:*
* ***Design tool****:*
* ***Manage Source code tool****: Git,Github*
* ***Test tool*** *:*

# 10. ANALYSIS & MODELING REQUIREMENT

## 10.1. Use Case Diagram

**

## 10.2. List of Business Rule

|  |  |  |
| --- | --- | --- |
| Code | Business rules | Use case |
| *BR1* | *Discount applied when customers purchase more than 5 orders.* | *UC.02* |
| *BR2* | *Purchase items during the promotional period or use discount codes distributed by the establishment.* | *UC.02* |
| *BR3* | For each order, 100 points will be added to the customer's account. Customers can use these points to redeem discount codes. | *UC. 02* |

## 10.3. Activity Diagram

**a) The steps performed of function abc….**

* **Payment function:**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| **1.** Customers add to cart. | **2.** Display shopping cart |
| **3.** The customer chooses to pay. | **4.** Proceed to payment form. |
| **5.** The customer enters all necessary information and selects the delivery method. | **6.** The system displays the payment methods. |
| **7** The customer selects the payment method and makes the payment. | **8** Display the invoice |

* **Search for items function:**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. Customer enters the name of the product they want to search for | 2. The system displays the item according to the customer's request |

**Login:**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| **The user accesses the page login information.** | **Displays the login page** |
| **Enter login information (username and password).** | **Check and verify information.** |
| **Click login** | **If yes, create a login session and redirect to the main page or personal page** |

**Register:**

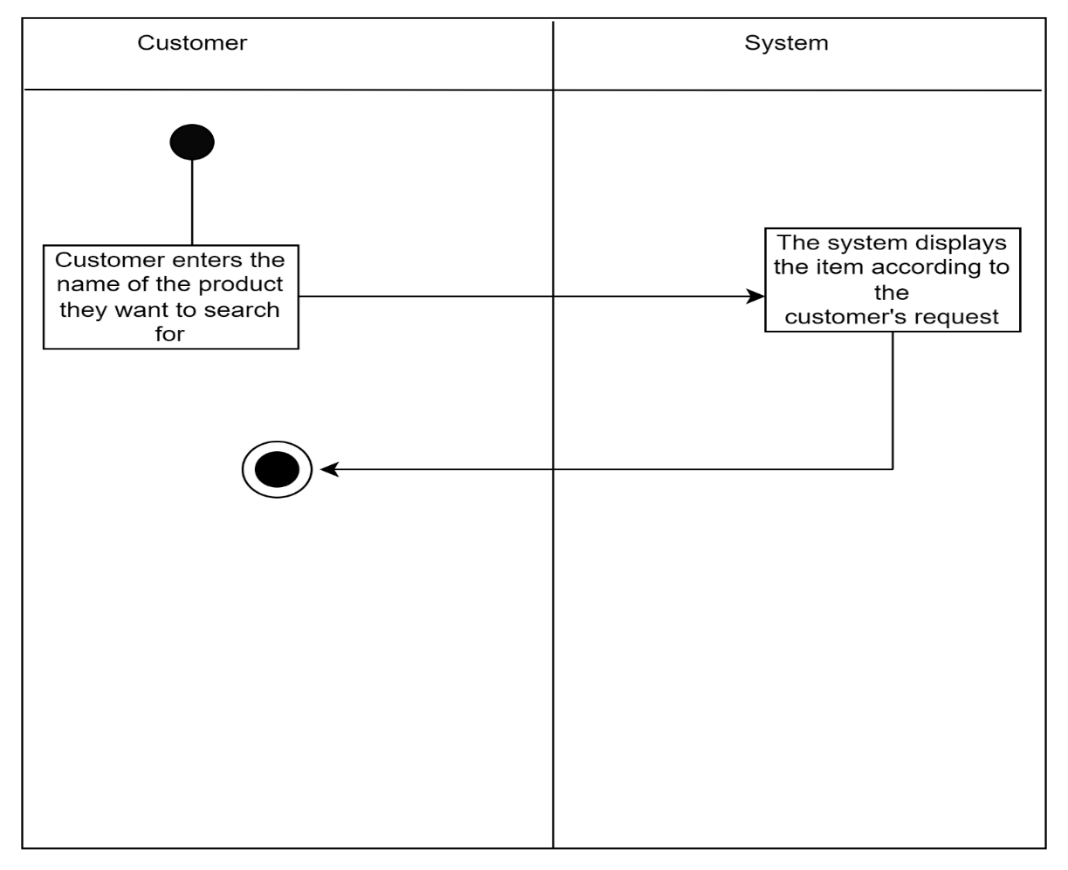
|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| **The user visits the registration page.** | **Displays the registration page** |
| **Enter the required information (name, email, password, etc.).** | **Check the validity of the information and confirm.** |
| **Click register** | **Create a new account if the information is valid.** |

**See menu list:**

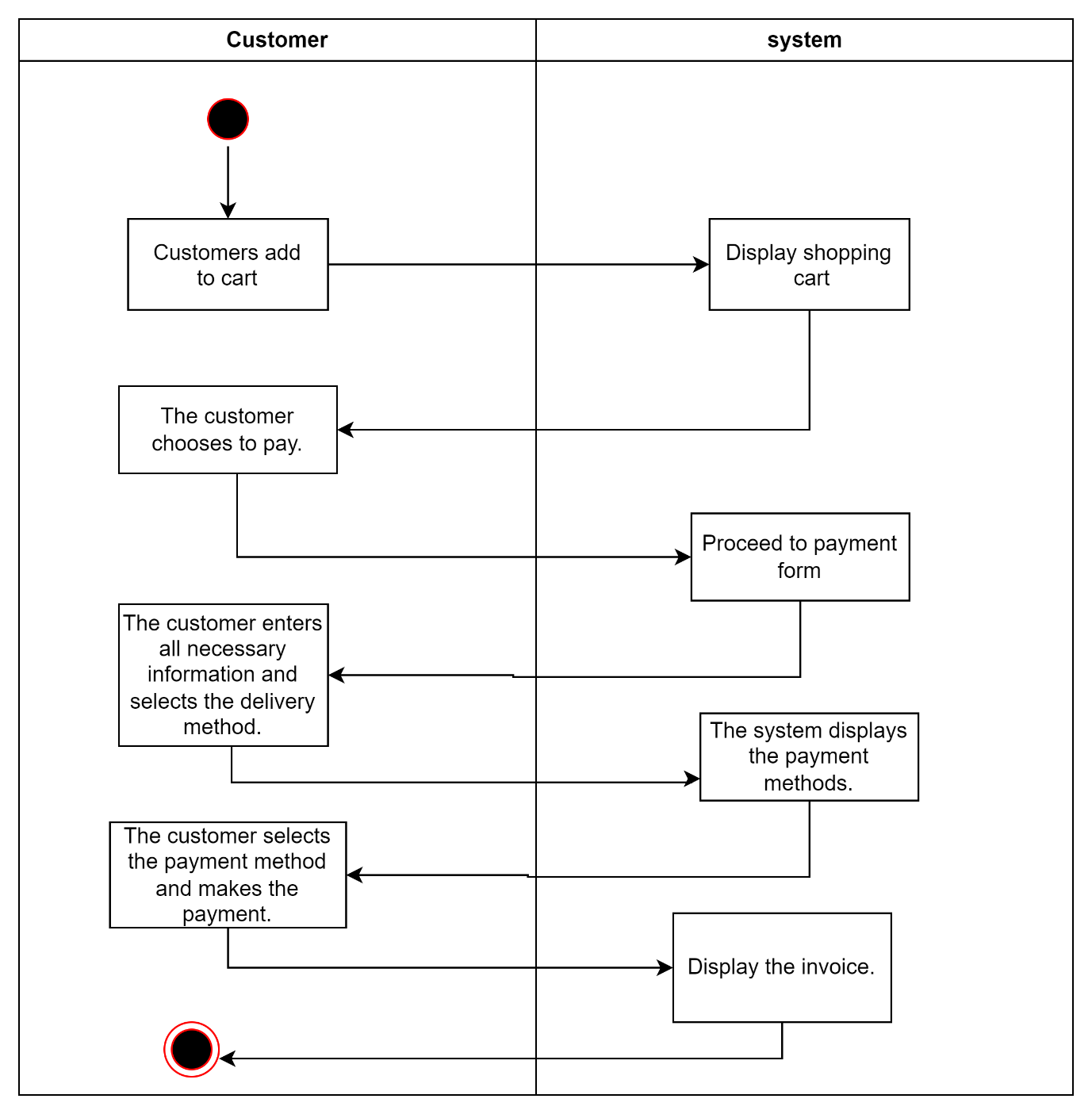
|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| **User logs into the system.** | **Show home page** |
| **Visit the drink list page.** | **Displays a list of drinks with information such as name, description, price.** |
| **Click search** | **Provide filtering or search options for user convenience** |
| **Choose a drink** | **Allow users to click on each drink to see detailed information.** |

**b) Activity Diagram of function abc….**

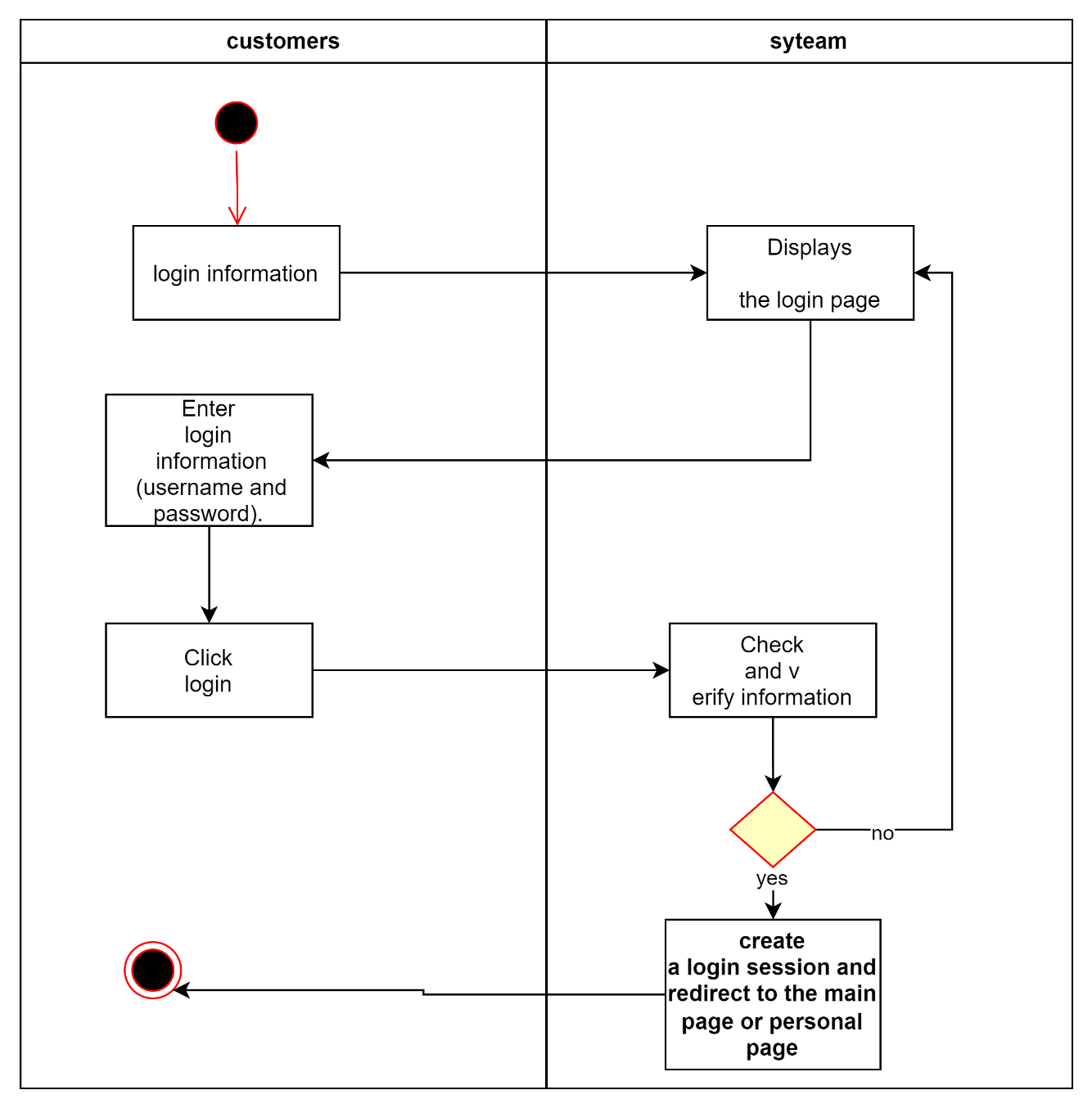
**Search for items function:**

**

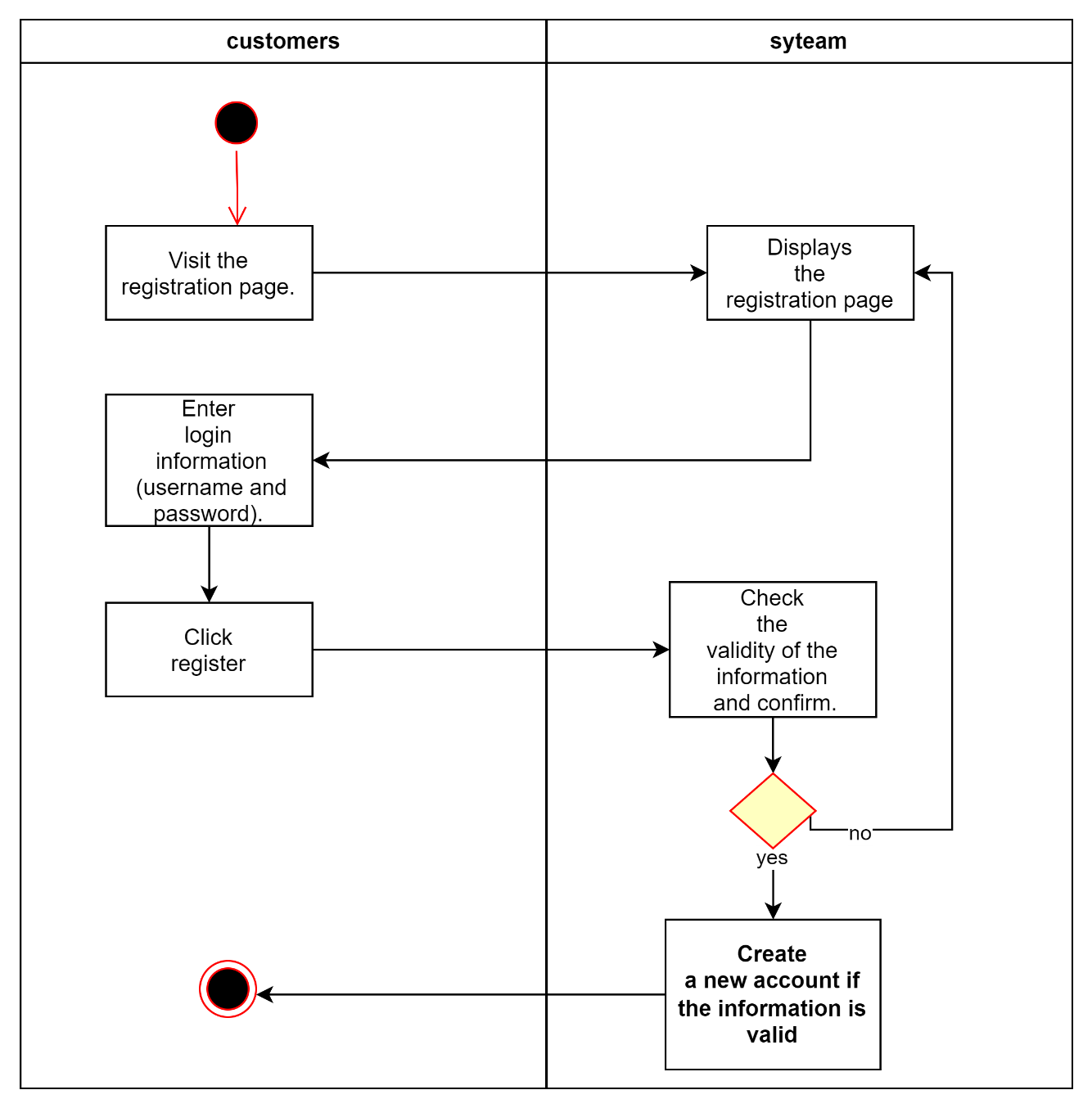
* **Payment function:**

****

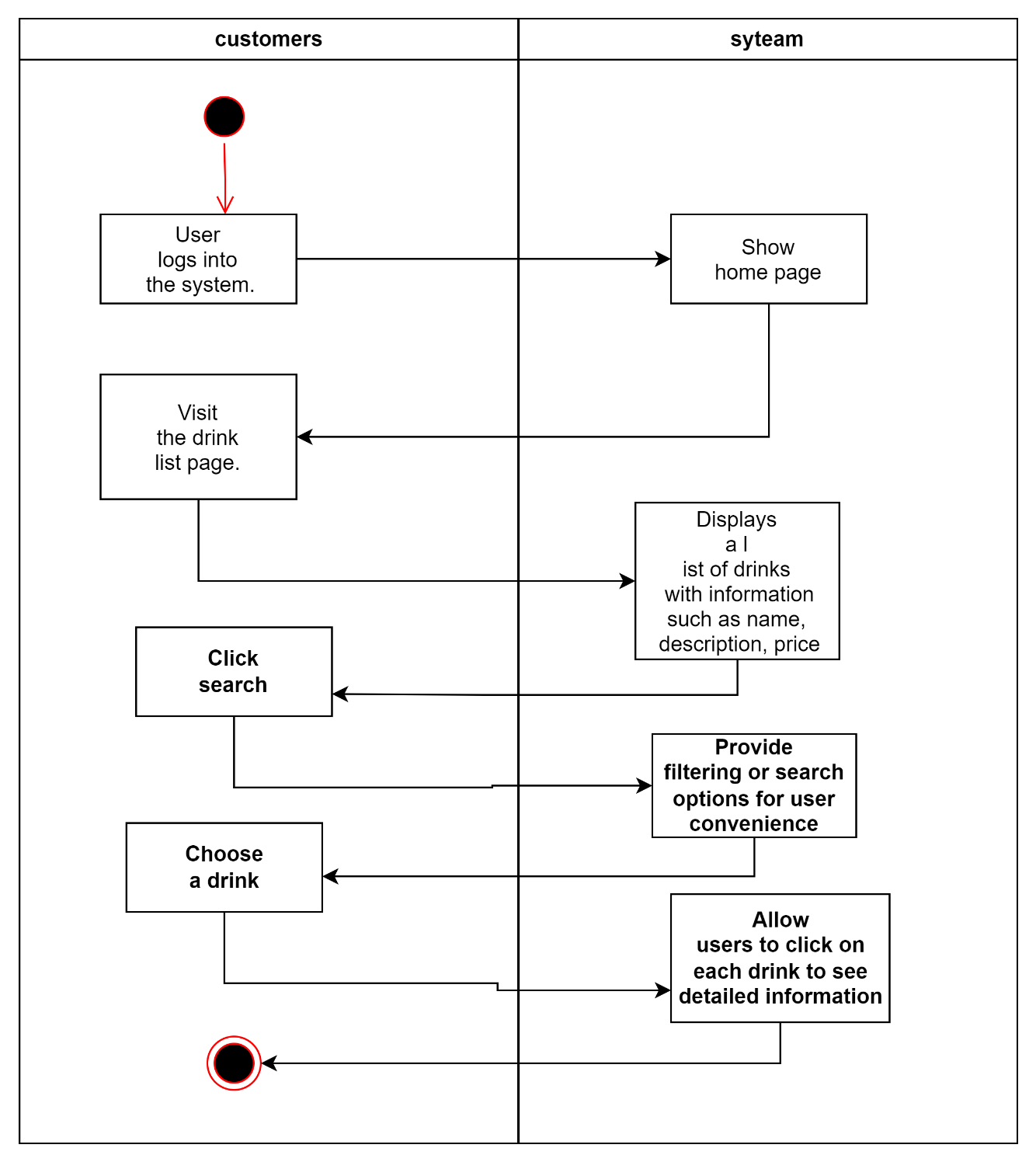
**Login:**

****

**Register:**

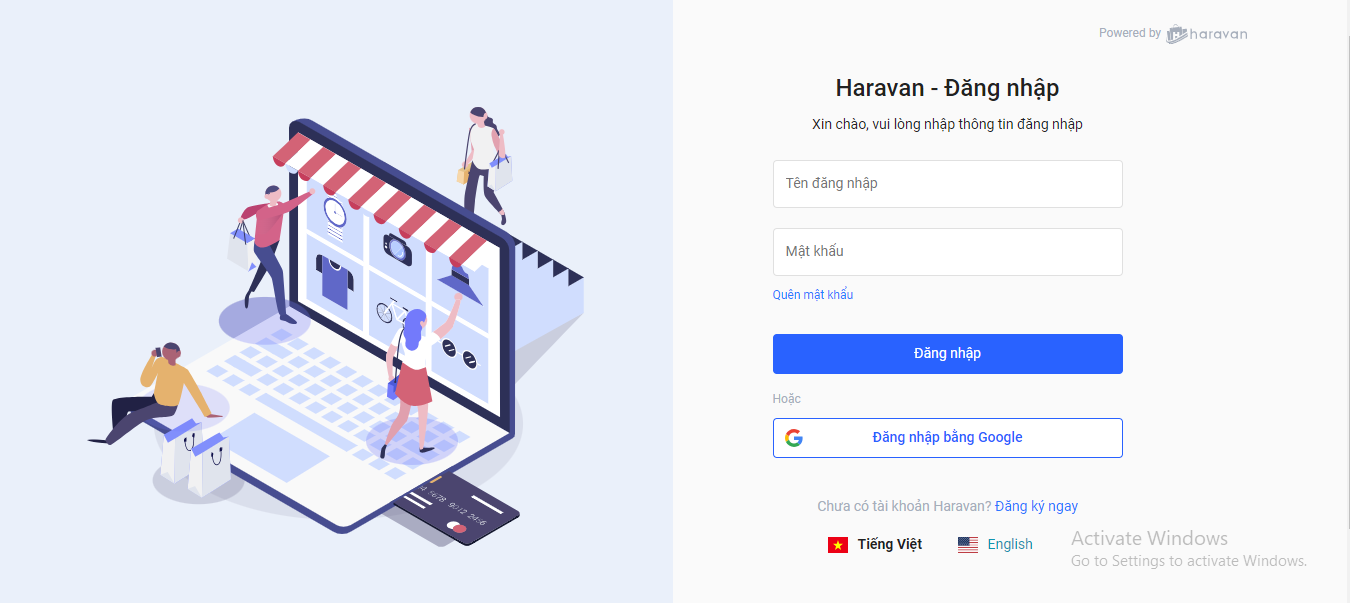
****

**See menu list:**

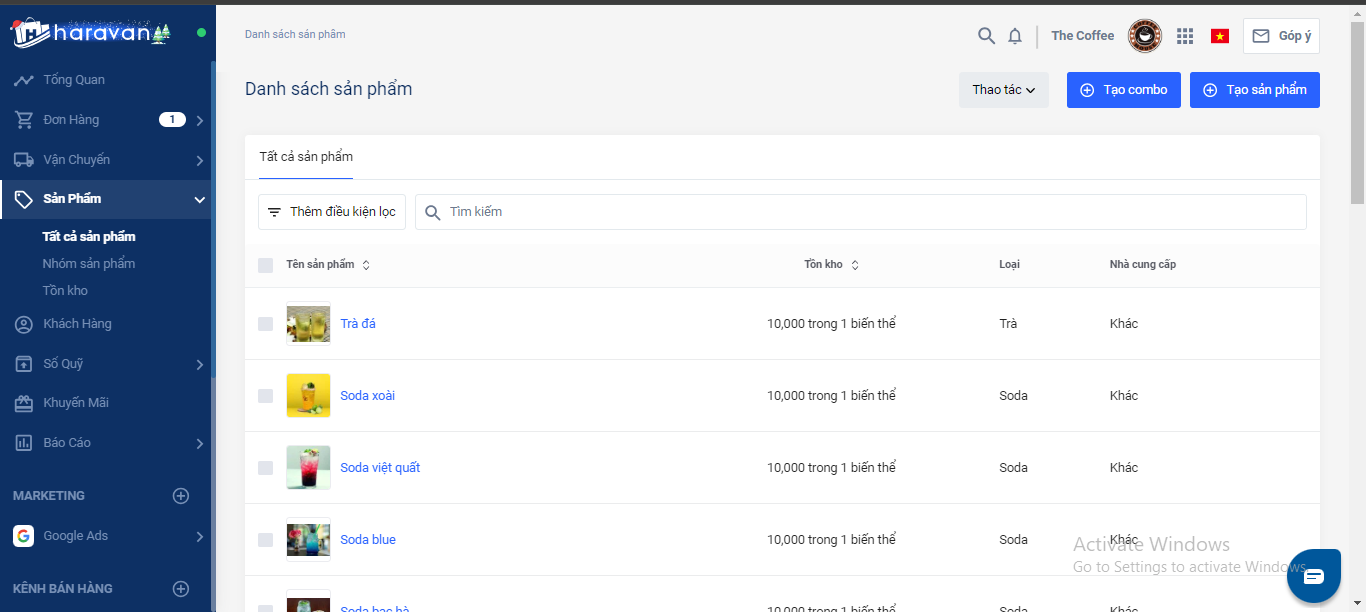
****

# 11. INTERFACE DESIGN

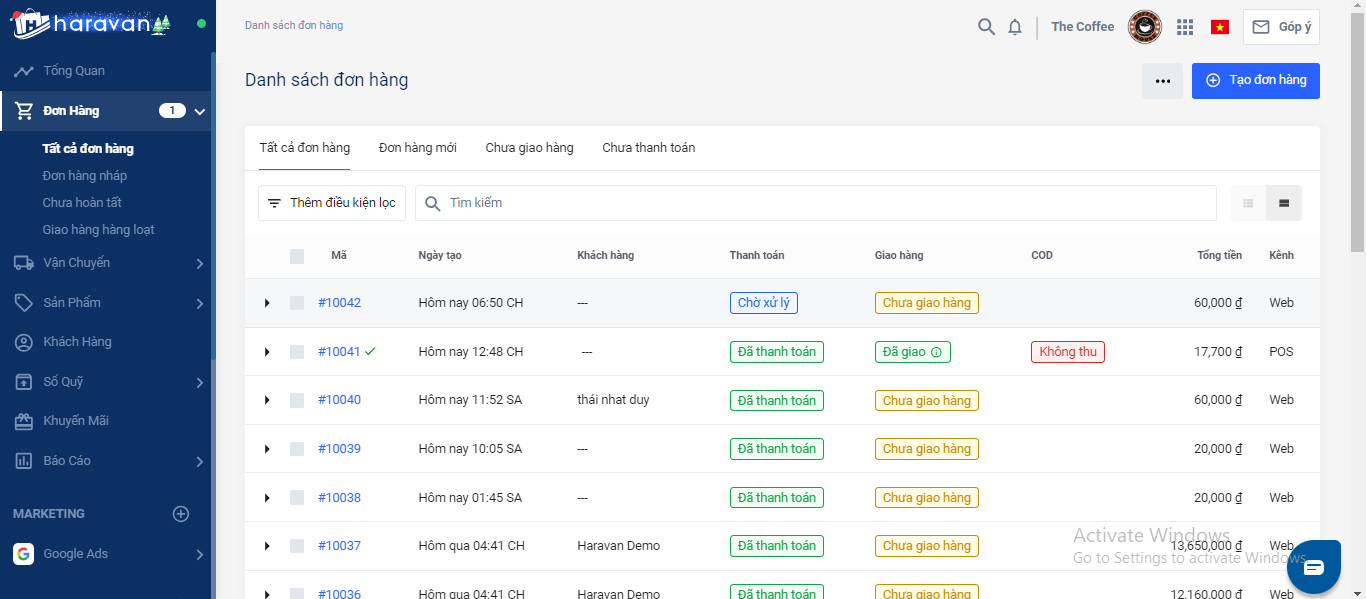
Interface for Login for admin Function:

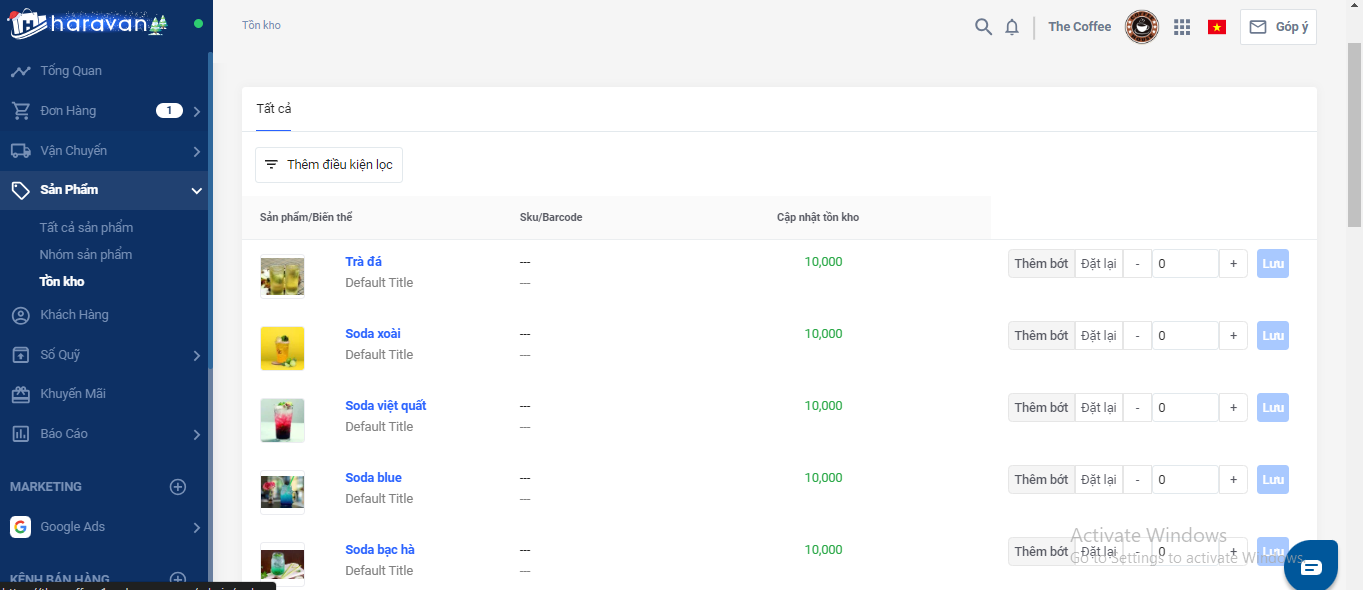


Interface for Magane menus and prices Function:

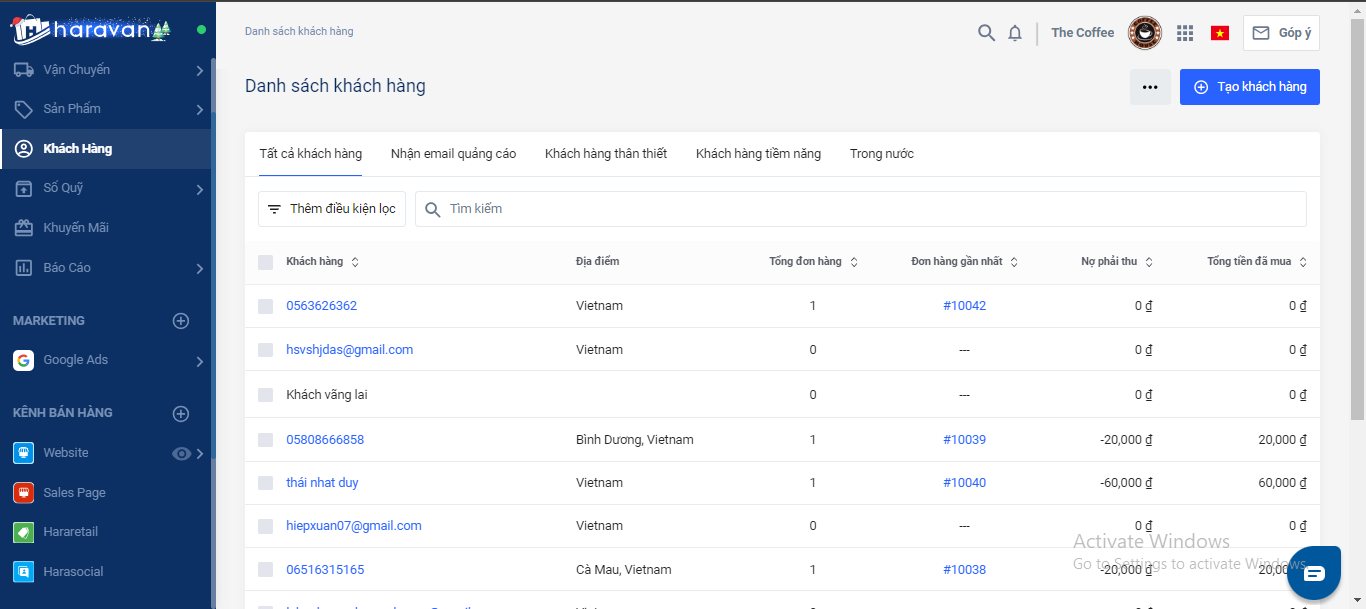


Interface for order management function:

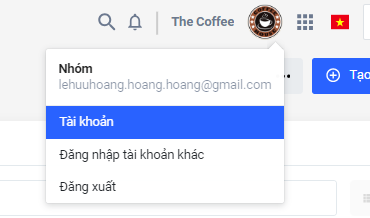


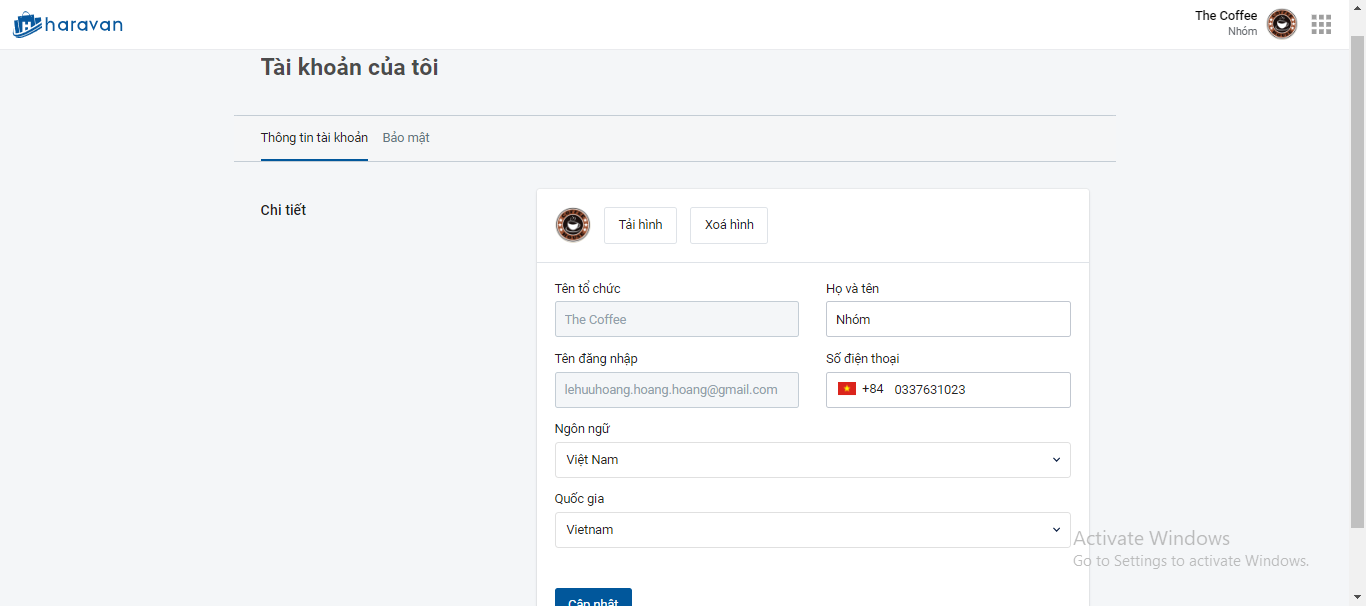
Interface for inventory management functions: 

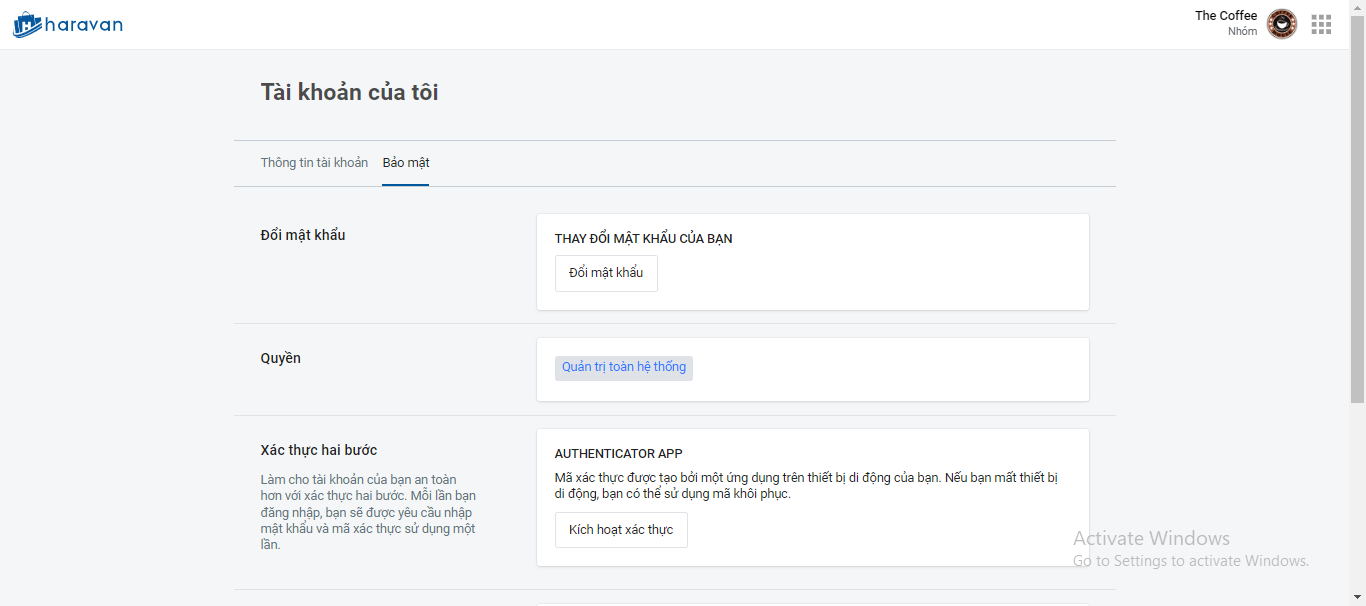
Interface for Customer management Function:



Interface for Account magagement Function:





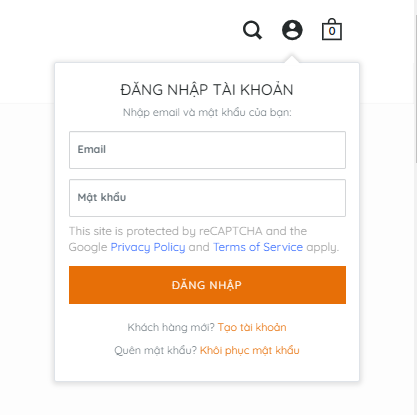


## 11.1. Introduction tool design:

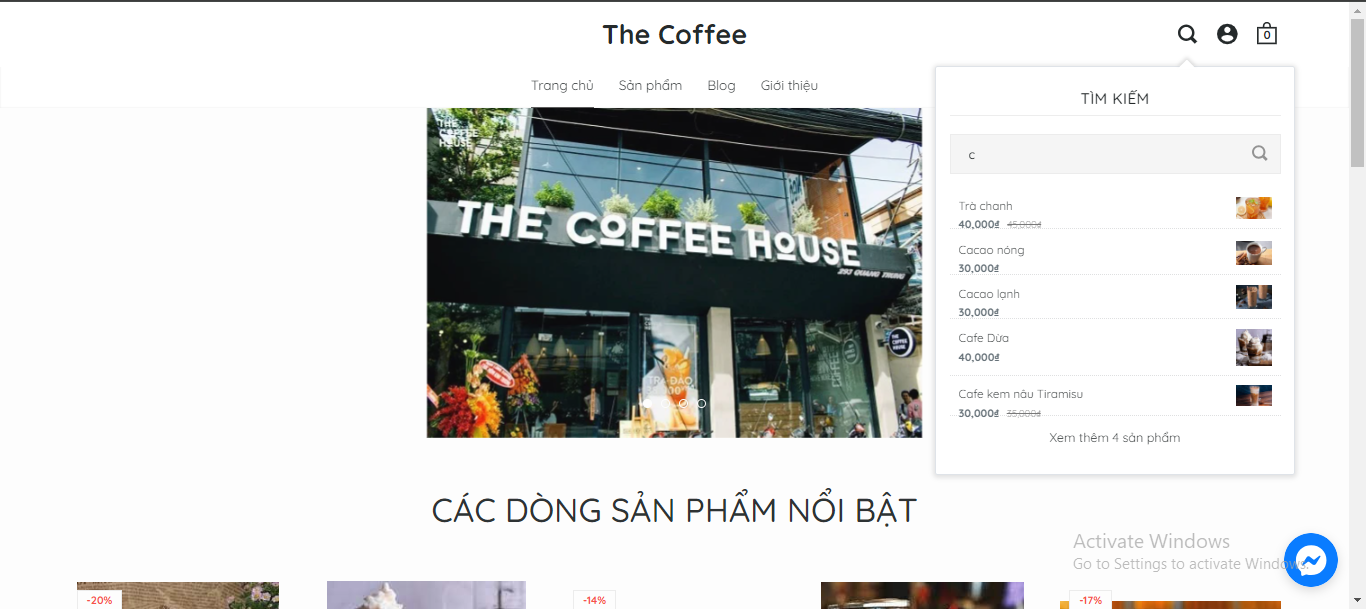
- Haravan

## 11.2 User Interface Design for ABC Function

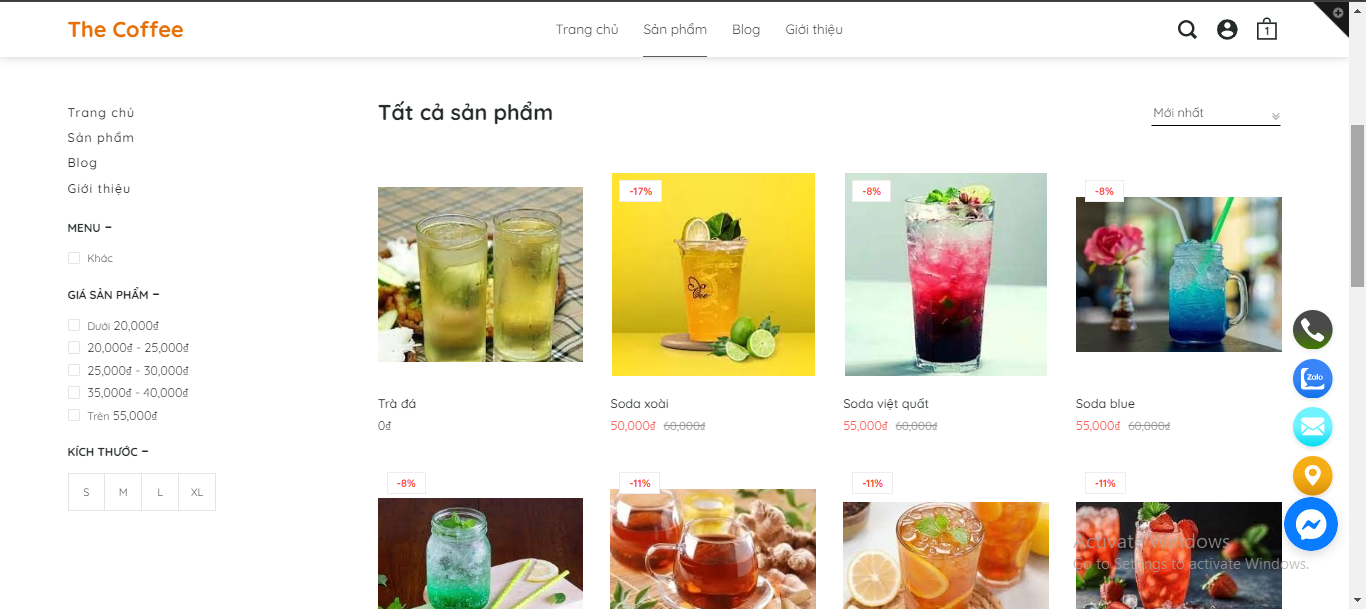
Interface for customer login function:



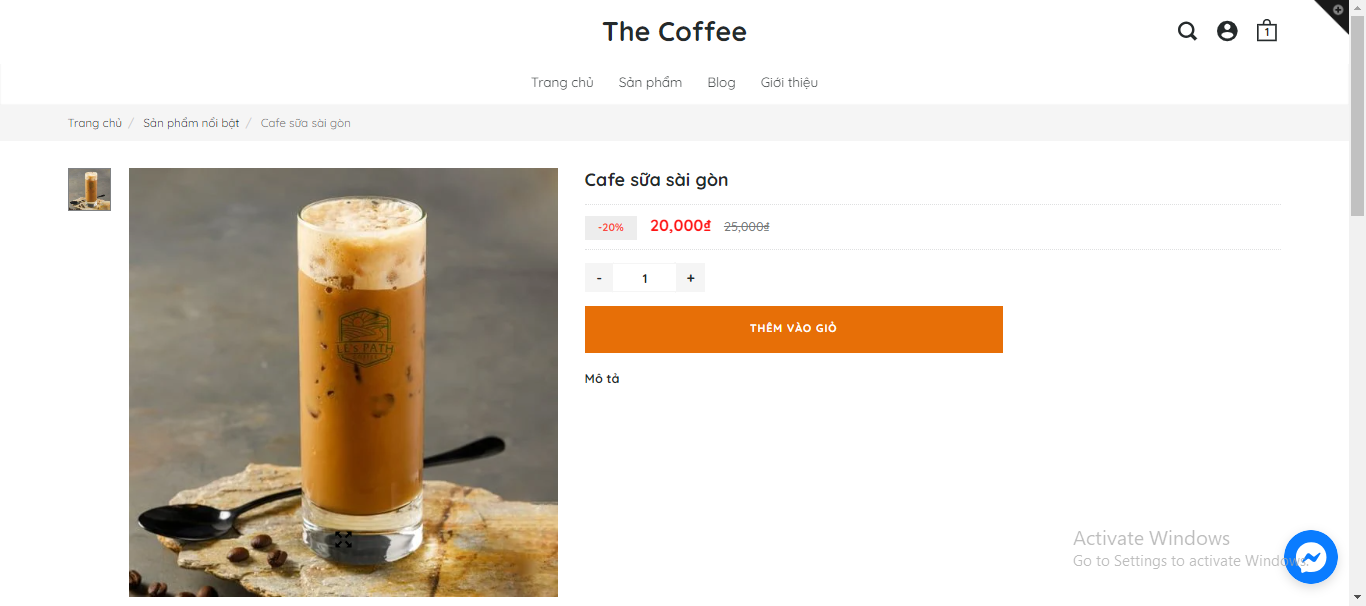
interface for searching function items:



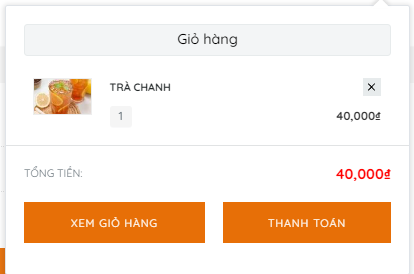
Product interface:



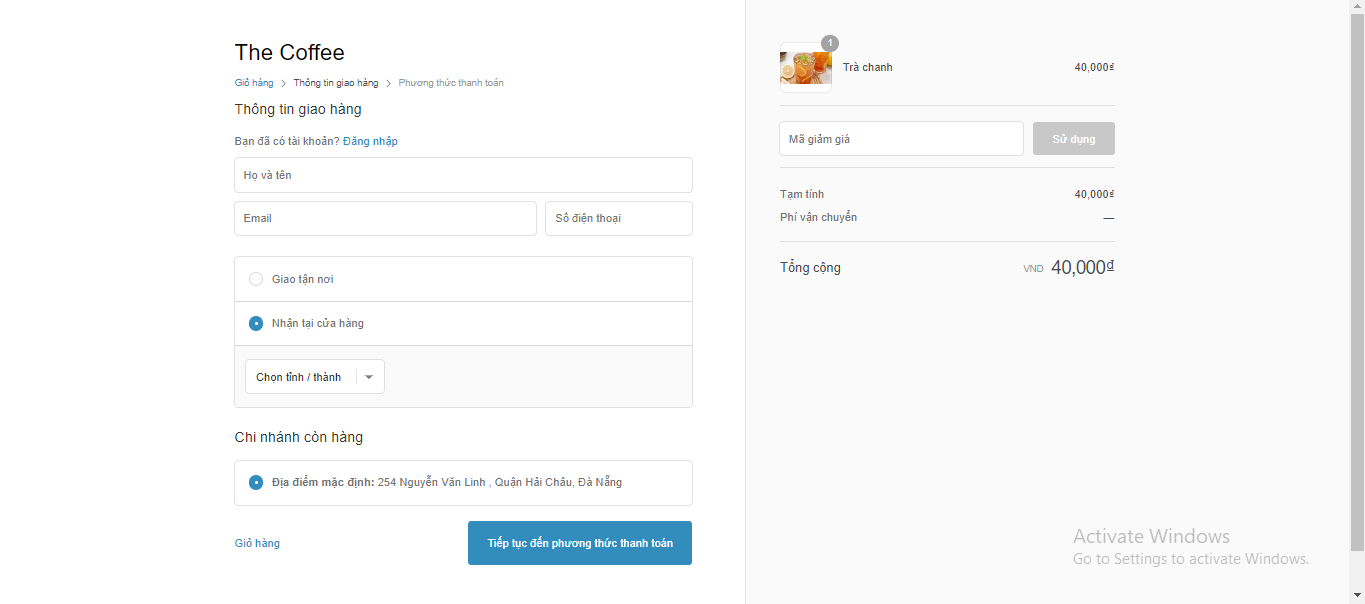
Interface for adding items:

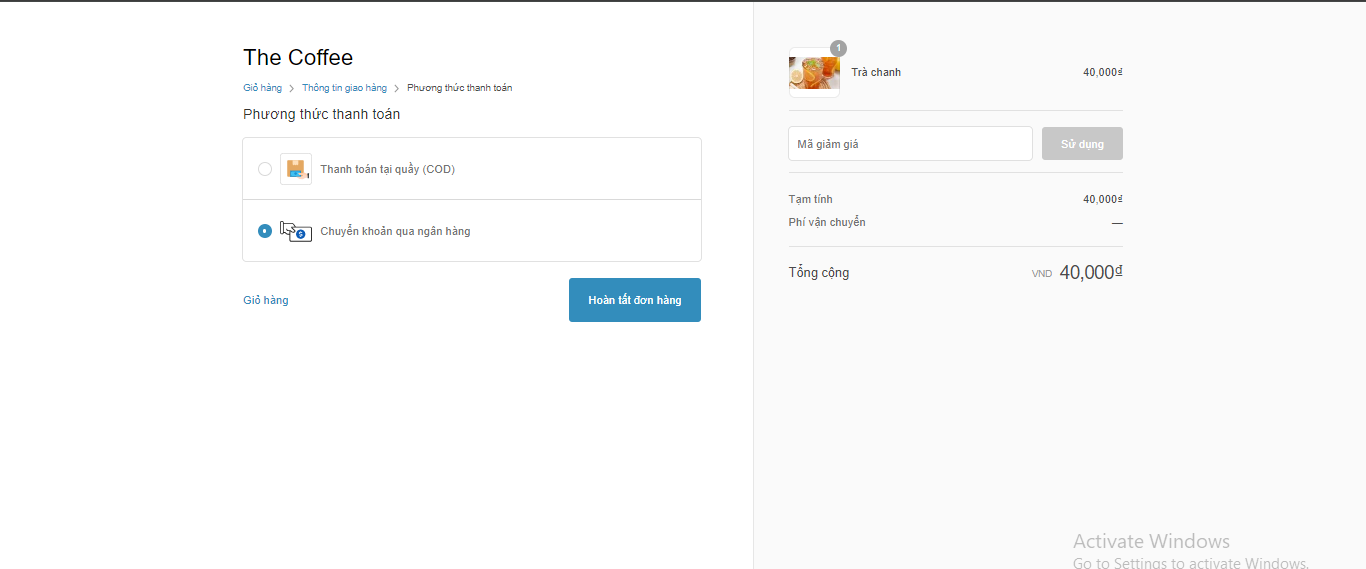


Shopping cart function interface:

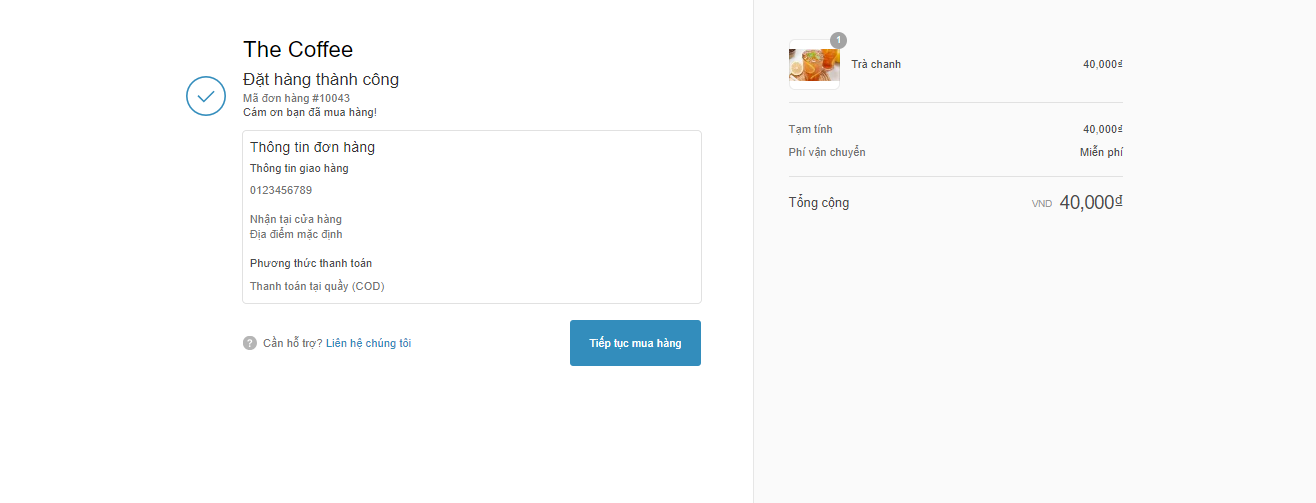


Interface for payment function:





Invoice printing interface:



# 12. CONFIGURATION MANAGEMENT

**a) Introduction the tool Source Code Management**

**GitHub** is a source code repository hosting service and project management system widely used in the open-source software community and software development projects. Here's an introduction to GitHub:

**1. Source Code Repository:**

GitHub provides a secure and remotely accessible location to store your project's source code.

Each project on GitHub has its own repository where you can store source code, documentation, images, and various types of files.

**2. Version Control:**

It employs a version control system to track changes in the source code over time.

Developers can create branches to work on features without affecting the main version of the project.

**3. Multi-Person Collaboration:**

GitHub allows multiple developers to collaborate on a project.

Features like pull requests facilitate reviewing and integrating changes from multiple sources.

**4. Issue and Project Tracking:**

Issue tracking helps in tracking and resolving problems, tasks, or features within the project.

Projects can be organized into sub-projects for more efficient management.

**5. CI/CD Integration:**

Continuous Integration and Continuous Deployment support automate testing and deployment when there are changes in the source code.

GitHub Actions is a powerful feature for configuring automated workflows.

**6. Community and Networking:**

GitHub fosters collaboration and communication among developers.

Users can follow projects, report issues, contribute code, and engage in discussions.

**7. Platform Integration:**

Supports integration with various development tools and services.

Capable of linking with services like Slack, Jenkins, Jira, and many others.

**8. Security and Permission Management:**

GitHub features security measures such as two-factor authentication, access control management, and permissions control.

**9. Searchability:**

Easy searchability for source code, issues, and projects on GitHub.

GitHub is not only a source code repository service but also a robust community that facilitates connection and collaboration within the software development community.

**b) Introduction Key Functional**

**GitHub** offers various key features for source code management and project collaboration. Here are some main functions:

**1. Repositories:**

Create a Repository: Establish a new repository to store source code and documentation.

Clone Repository: Copy a repository from GitHub to the local machine for development.

**2. Branching:**

Create a Branch: Generate a new branch to develop a feature without affecting the main version.

Merge Branches: Combine changes from one branch into another.

**3. Pull Requests:**

Open a Pull Request: Initiate a pull request to propose changes and integrate them into the main branch.

Review Code: Inspect and evaluate source code within a pull request.

**4. Issues:**

Create an Issue: Generate an issue to track tasks, problems, or features that need addressing.

Labeling and Milestones: Apply labels and set milestones to manage issues more effectively.

**5. Actions:**

Automate Workflow: Use GitHub Actions to automate testing, deployment, and other tasks.

Workflow Templates: Utilize templates to create automated workflows.

**6. Projects:**

Create a Project Board: Establish a project board to manage tasks and track progress.

Kanban Board: Use a Kanban board to move and manage task cards.

**7. Wiki and Pages:**

Create a Wiki: Document guides, tutorials, and other important information.

GitHub Pages: Publish static websites based on repository content.

**8. Security:**

Code Scanning: Scan source code to detect security vulnerabilities.

Dependency Graph: View and manage project dependencies.

**9. Collaboration and Discussions:**

Collaborators: Add collaborators to the project with specific permissions.

Discussions: Engage in direct discussions within issues or pull requests.

**10. Insights:**

Analytics: View information about project activity, contributions, and health.

Code Frequency: Examine a chart showing the frequency of source code changes over time.

**11. Settings:**

Repository Settings: Configure options such as access permissions, security, and integrations.

Branch Protection: Safeguard the main branch against unintended changes.

**c) The Screen Shot apply Source Code Management of team**

<Giao diện quản lý mã nguồn/tài liệu của nhóm>

# 13. TESTING

**a) Introduction to method testing apply of this project**

*Black box testing is a method of software testing in which the internal structure, design, and implementation details of the software being tested are not known to the tester. The tester focuses on input and output, examining the software's functionality without any knowledge of its internal code. The goal is to ensure that the software behaves according to its specifications.*

**b) Test Cases chưa có đủ**

*<Thiết kế test case cho các chức năng (05 chức năng bất kỳ), theo mẫu Test Case đính kèm>*

# 14. CONTRIBUTION

*<Team member contributed significantly to team's success (%)>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Participant**  *<Họ tên>* | **Role**  *<Vai trò>* | **Responsibilities**  *<Trách nhiệm cụ thể được phân công, đóng góp cho bài tập nhóm>* | **%**  *Mức độ (%) đóng góp cho bài tập nhóm* |
|  |  |  |  |  |