

**FPT UNIVERSITY**

Software Development Project Document

**STUDENT DEAL**

|  |  |
| --- | --- |
| Class | JS1701 |
| Mentor | Mr. Lâm Hữu Khánh Phương |
| Group member | Nguyễn Gia Khiêm  Vũ Thị Bích Phương  Đậu Hoàng Khôi  Hồ Đắc Nhân Tâm  Dương Tuấn Kiệt |

– Ho Chi Minh, July 2023 –

# **Table of contents**

[**Table of contents** 2](#_Toc130714862)

[**Acknowledgement** 4](#_Toc130714863)

[**Definition and Acronyms** 5](#_Toc130714864)

[**I.** **Project Introduction** 6](#_Toc130714865)

[**1.** **Overview** 6](#_Toc130714866)

[**1.1** **Project Information** 6](#_Toc130714867)

[**1.2** **Project Team** 6](#_Toc130714868)

[**2.** **Product Background** 6](#_Toc130714869)

[**3.** **Business Opportunity** 7](#_Toc130714870)

[**4.** **Software Product Vision** 7](#_Toc130714871)

[**II.** **Software Requirements Specification** 8](#_Toc130714872)

[**1.** **Overall Description** 8](#_Toc130714873)

[**1.1** **User Requirements Overview** 8](#_Toc130714874)

[**1.1.1** **Guest** 8](#_Toc130714875)

[**1.1.2** **Customer** 8](#_Toc130714876)

[**1.1.3** **Admin** 8](#_Toc130714877)

[**1.2** **Business Rules** 8](#_Toc130714878)

[**2.** **User Requirements** 9](#_Toc130714879)

[**2.1** **System Actor** 9](#_Toc130714881)

[**2.1.1** **Guest** 9](#_Toc130714882)

[**2.1.2** **Customer** 9](#_Toc130714883)

[**2.1.3** **Admin** 9](#_Toc130714884)

[**2.2** **Use Case Diagram** 10](#_Toc130714885)

[**2.2.1** **Guest and Customer** 10](#_Toc130714886)

[**2.2.2** **Admin** 11](#_Toc130714887)

[**2.3** **Use Case List** 11](#_Toc130714888)

[**2.4** **Use Case Specification** 13](#_Toc130714889)

[**3.** **Functional Requirements** 45](#_Toc130714890)

[**3.1** **System Functional Overview** 45](#_Toc130714894)

[**3.1.1** **Screen Flow** 45](#_Toc130714895)

[**3.1.2** **Screen Description** 46](#_Toc130714896)

[**3.1.3** **Entity Relationship Diagram** 47](#_Toc130714897)

[**4.** **Non-Functional Requirements** 48](#_Toc130714898)

[**4.1** **External Interface** 48](#_Toc130714900)

[**4.1.1** **Hardware Interface** 48](#_Toc130714901)

[**4.1.2** **Software Interface** 48](#_Toc130714902)

[**4.1.3** **User Interface** 48](#_Toc130714903)

[**4.1.4** **Communication Interface** 48](#_Toc130714904)

[**4.2** **Quality Attribute** 48](#_Toc130714905)

[**4.2.1** **Performance** 48](#_Toc130714906)

[**4.2.2** **Usability** 48](#_Toc130714907)

[**4.2.3** **Security** 48](#_Toc130714908)

[**III.** **Software Design Description** 49](#_Toc130714909)

[**1.** **Overall Description** 49](#_Toc130714910)

[**1.1** **Assumptions** 49](#_Toc130714911)

[**2.** **Overall System Architecture** 49](#_Toc130714912)

[**3.** **Data and Database Design** 50](#_Toc130714914)

[**3.1** **Database Design** 50](#_Toc130714916)

[**3.2** **Data File Design** 51](#_Toc130714917)

[**3.2.1** **Brand Table** 51](#_Toc130714918)

[**3.2.2** **Cart Table** 51](#_Toc130714919)

[**3.2.3** **Category Table** 51](#_Toc130714920)

[**3.2.4** **Customer Table** 52](#_Toc130714921)

[**3.2.5** **Feedback Table** 52](#_Toc130714922)

[**3.2.6** **Manager Table** 53](#_Toc130714923)

[**3.2.7** **MembershipTier Table** 53](#_Toc130714924)

[**3.2.8** **Orders Table** 53](#_Toc130714925)

[**3.2.9** **OrdersDetail Table** 54](#_Toc130714926)

[**3.2.10** **Get vouchermentMethod Table** 54](#_Toc130714927)

[**3.2.11** **Product Table** 54](#_Toc130714928)

[**3.2.12** **ShippingMethod Table** 55](#_Toc130714929)

[**3.2.13** **ProductSize Table** 55](#_Toc130714930)

# **Acknowledgement**

We extend our sincere appreciation and gratitude to all those who have played a vital role in the successful completion of this project to establish an Student Deal.

We also extend our heartfelt thanks to our project mentor Mr. Lam Vu Khanh Phuong, whose leadership, guidance, and support were essential in keeping the project on track and ensuring its success.

Our gratitude goes to the project team members: Vu Thi Bích Phuong, Nguyen Gia Khiem, Dau Hoang Khoi, Ho Dac Nhan Tam, Duong Tuan Kiet who worked tirelessly to bring this project to fruition. Their dedication, professionalism, and teamwork were instrumental in achieving the project goals.

We would also like to acknowledge the feedback and contributions of the stakeholders and end-users, who provided valuable insights and recommendations, helping us to improve the quality of the store.

Finally, we would like to thank all others who have directly or indirectly contributed to this project's success.

Thank you all for your contributions, support, and unwavering commitment to this project.

Best Regards,

Project team.

# **Definition and Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| UC | Use Case |
| SC | Screen Description |
| ERD | Entity Relationship Diagram |
| API | Application Program Interface |
| Guest | The users who don’t need to login but can using the web application |
| Customer | The users who need to login to using more function |
| Admin | The users who need to login to manages the web application |

Table 1 – Definition and Acronyms

1. **Project Introduction**
   1. **Overview**
   2. **Project Information**

* Project name: STUDENT DEAL
* Subject code: SWP391
* Software type: Web Application
  1. **Project Team**

|  |  |  |
| --- | --- | --- |
| **Full name** | **Email** | **Role** |
| Lâm Hữu Khánh Phương | phuonglhk@fe.edu.vn | Mentor |

Table 2 – Mentor

|  |  |  |
| --- | --- | --- |
| **Full name** | **Email** | **Role** |
| Đậu Hoàng Khôi |  | Leader |
| Vũ Thị Bích Phương |  | Member |
| Nguyễn Gia Khiêm |  | Member |
| Hồ Đắc Nhân Tâm |  | Member |
| Dương Tuấn Kiệt |  | Member |

Table 3 – Team members

* 1. **Product Background**
* Target voucher audience: Student Deal mainly target vouchers students. Its platform aims to provide special discounts and offers tailored to the needs and interests of students. The target voucher audience includes high school students, college students and university students from various educational institutions.
* Brand Partnerships: Student Deal establishes partnerships with brands and retailers to offer exclusive student deals. These partnerships are vital to the platform's success as they offer a wide variety of discounts and offers across different product categories such as fashion, technology, travel, entertainment, and more. so again.
* User registration and verification: To access offers and discounts, students need to create an account on the Student Deal platform. The registration process typically involves verifying the user's student status, which can be done through an authentication process using their institution's email address or related verification methods. other.
* Aggregating and distributing discounts: Student Deal collects information about available offers and discounts from its brand partners and aggregates them on its platform. document.
* Communication and Interaction: Student Deal maintains communication channels with its users, including email newsletters, push notifications, and social media platforms.
* Mobile and web presence: Student Deal offers both mobile apps and web platforms to cater to different user preferences and ensure accessibility.
* Measurement and analytics: Student Deal uses data analytics to measure the effectiveness of promotions, track user engagement, and better understand user preferences and behavior . This data-driven approach helps the company optimize services and improve overall performance.
* By understanding these business requirements, Student Deal can continue to improve its platform and provide valuable services to its student users while maintaining strong partnerships with other businesses. brands and retailers
  1. **Business Opportunity**
* User Acquisition and Growth: One of the main goals of Student Deal is to acquire a large and active student user base. The company aims to attract new users through marketing campaigns, partnerships with educational institutions, and word-of-mouth referrals. The goal is to achieve sustained user growth and increase the platform's reach among students.
* Brand partnerships and revenue generation: Student Deal relies on brand partnerships to provide exclusive deals and discounts to its user base. The main objective is to establish and maintain strong relationships with a wide variety of brands and retailers. By providing a valuable marketing channel to these partners, Student Deal aims to generate revenue through sponsored partnerships, advertising and promotions.
* User engagement and retention: Student Deal aims to provide a compelling user experience that encourages high engagement and repeat use. The goal is to keep students actively using the platform by offering personalized recommendations, timely notifications of new discounts, and a seamless browsing and redemption process. By focusing on user engagement and retention, Student Deal aims to build long-term loyalty and increase customer lifetime value.
* Expansion into new markets: While the Student Deal may be present in certain regions or countries, the goals may be to expand into new markets. This could involve launching the platform in new countries, forming partnerships with international brands, and tailoring the offering to meet the specific needs of different student groups. . The goal is to tap the global student market and drive international growth.
* Data analysis and optimization: Student Deal aims to leverage data analytics to better understand user behavior, preferences and market trends. The goal is to optimize the platform's services by identifying popular categories, improving the personalization algorithm, and refining the user experience. Data analytics also helps measure the effectiveness of marketing campaigns and identify opportunities for improvement.
* Reputation and Brand Awareness: Building a strong brand reputation and increasing brand awareness among students and brands is the main goal of Student Deal. The company aims to position itself as a trusted and trustworthy platform for student discounts and offers. This involves maintaining a positive brand image, delivering on promises and actively interacting with the student community through various marketing and communication channels.
* Innovation and difference: To stay competitive in the market, Student Deal aims to continuously innovate and differentiate itself from competitors. This may involve introducing new features, exploring emerging technologies, and staying up to date with changing student interests and trends. The goal is to deliver a unique value proposition that differentiates the Student Deal and appeals to both users and brand partners.
* By aligning its business goals with these key areas, Student Deal can strive to achieve growth, profitability and a strong position in the market as the leading platform for Student deals and discounts.
  1. **Software Product Vision**
* User Acquisition and Growth: One of the main goals of Student Deal is to acquire a large and active student user base. The company aims to attract new users through marketing campaigns, partnerships with educational institutions, and word-of-mouth referrals. The goal is to achieve sustained user growth and increase the platform's reach among students.
* Brand partnerships and revenue generation: Student Deal relies on brand partnerships to provide exclusive deals and discounts to its user base. The main objective is to establish and maintain strong relationships with a wide variety of brands and retailers. By providing a valuable marketing channel to these partners, Student Deal aims to generate revenue through sponsored partnerships, advertising and promotions.
* User engagement and retention: Student Deal aims to provide a compelling user experience that encourages high engagement and repeat use. The goal is to keep students actively using the platform by offering personalized recommendations, timely notifications of new discounts, and a seamless browsing and redemption process. By focusing on user engagement and retention, Student Deal aims to build long-term loyalty and increase customer lifetime value.
* Expansion into new markets: While the Student Deal may be present in certain regions or countries, the goals may be to expand into new markets. This could involve launching the platform in new countries, forming partnerships with international brands, and tailoring the offering to meet the specific needs of different student groups. . The goal is to tap the global student market and drive international growth.
* Data analysis and optimization: Student Deal aims to leverage data analytics to better understand user behavior, preferences and market trends. The goal is to optimize the platform's services by identifying popular categories, improving the personalization algorithm, and refining the user experience. Data analytics also helps measure the effectiveness of marketing campaigns and identify opportunities for improvement
* Reputation and Brand Awareness: Building a strong brand reputation and increasing brand awareness among students and brands is the main goal of Student Deal. The company aims to position itself as a trusted and trustworthy platform for student discounts and offers. This involves maintaining a positive brand image, delivering on promises and actively interacting with the student community through various marketing and communication channels.
* Innovation and difference: To stay competitive in the market, Student Deal aims to continuously innovate and differentiate itself from competitors. This may involve introducing new features, exploring emerging technologies, and staying up to date with changing student interests and trends. The goal is to deliver a unique value proposition that differentiates the Student Deal and appeals to both users and brand partners.
* By aligning its business goals with these key areas, Student Deal can strive to achieve growth, profitability and a strong position in the market as the leading platform for Student deals and discounts.

1. **Software Requirements Specification**
   1. **Overall Description**
   2. **User Requirements Overview**

* The system will maintain an up-to-date inventory of products and their stock levels. The system will update the stock levels when a purchase is made and will notify customers if a product is out of stock or on backorder. The system will also provide customers with a tracking number and estimated delivery date for each order.
* Student Deal will maintain the privacy and security of all user data and will comply with all relevant laws and regulations regarding data protection and user privacy. The system will also have measures in place to protect against common security threats, such as cross-site scripting and SQL injection attacks.
  + 1. **Guest**
* As a guest, users can view the products available in the store without logging in or registering. They can search for products, view product details, and add items to their cart. However, guests cannot make purchases or save their cart for future reference. If they wish to purchase items, they will need to create an account or fill the information when they checkout.
  + 1. **Customer**
* As a customer, users can create an account, log in, and save their personal information and addresses for future purchases. They can view their order history and track the status of their current orders. Customers can also leave feedback on products they have purchased, which will be visible to other customers.
* Customers can make purchases using a credit or debit card and can also save multiple get voucher methods for future use. The system will provide a secure checkout process, with all transactions processed using industry-standard encryption and security measures.
  + 1. **Admin**
* As an admin, users will have access to the dashboard where they can manage product inventory, view sales reports, and manage customer accounts. Admins can add, update, and delete products and categories, and can also view and respond to customer feedback. The system will provide a user-friendly interface for admins to manage the store, with easy-to-use tools and reporting features.
  1. **Business Rules**
* Student Deal operates 24/7 and provides products continuously. Prices are displayed in points and all transactions are securely processed using industry standard encryption and security protocols. The system will use the get voucher gateway to process get voucher and store information.
* The system will use RESTful APIs to integrate with third party services, such as Google Maps API for address validation and Gmail API for email communication and verification. The system will also implement pagination and filtering of product search results to improve performance and user experience.
  1. **User Requirements**

1. 1. **System Actor**

* The system will maintain the privacy and security of all user data and will ensure that user roles and permissions are always enforced. The system will also provide a user-friendly interface that is intuitive and easy to use, regardless of the user's level of technical expertise.
* The system will have three types of users: Guest, Customer, and Admin. Each type of user will have specific permissions and functionality within the system.
  + 1. **Guest**
* A guest is a user who has not yet created an account or logged in to an existing account. A guest user will have limited functionality within the system and will only be able to access the public pages of the web store. Guest users will not be able to add items to their cart or view their order history. However, they will be able to browse the products and view their details.
  + 1. **Customer**
* A customer is a user who has created an account and logged in to the system. A customer will have full access to the functionality of the web store, including the ability to add items to their cart, view their order history, and manage their account information. A customer will also be able to view their order status, update their address, and reset their password.
  + 1. **Admin**
* An admin is a user who has been granted special permissions within the system. An admin will have access to the admin dashboard, which will allow them to perform CRUD operations on products, categories, and orders. Admin users will also be able to manage customer accounts, view reports, and update the site settings. Only authorized admins will have access to the admin dashboard, and their actions will be logged for security purposes.
  1. **Use Case Diagram**
     1. **Guest and Customer**

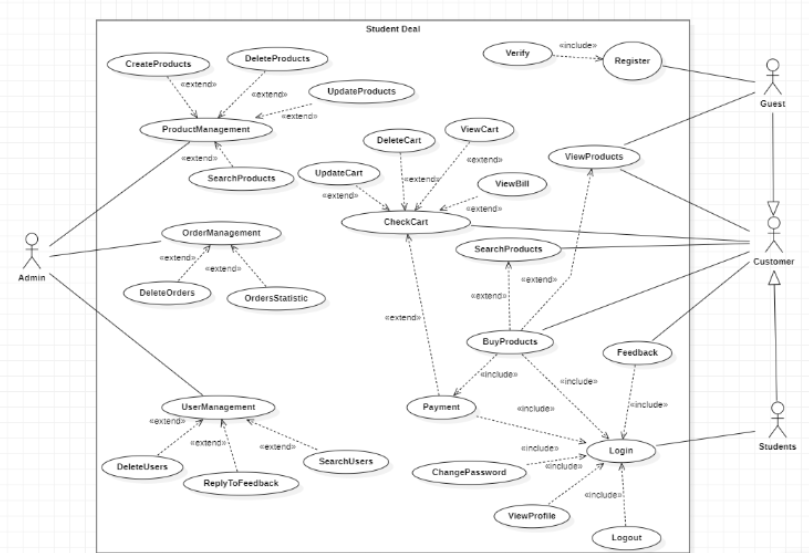
Z`

Figure 1 – Guest and Customer Use Case Diagram

Diagram

Description automatically generated

Figure 2 – Admin Use Case Diagram

* 1. **Use Case List**
* The following table lists all the use cases for the STUDENT DEAL website, along with their actors and descriptions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **Use Case Name** | **Primary Actor** | **Secondary Actors** |
| UC-01 | Register | Guest | N/A |
| UC-02 | Login | Customer | Admin |
| UC-03 | Login by Google Account | Customer | N/A |
| UC-04 | Logout | Customer | Admin |
| UC-05 | Reset Password | Customer | N/A |
| UC-06 | Search | Customer | Guest |
| UC-07 | Search by Filter | Customer | Guest |
| UC-08 | Show Products | Customer | Guest |
| UC-09 | Pagination | Customer | Guest |
| UC-10 | View Cart | Customer | Guest |
| UC-11 | Add to Cart | Customer | Guest |
| UC-12 | Edit Cart | Customer | Guest |
| UC-13 | Delete Item from Cart | Customer | Guest |
| UC-14 | Checkout | Customer | N/A |
| UC-15 | Show Customer Information | Customer | N/A |
| UC-16 | Edit Customer Information | Customer | N/A |
| UC-17 | Show Orders History | Customer | N/A |
| UC-18 | Leave Feedback | Customer | N/A |
| UC-19 | Create Product | Admin | N/A |
| UC-20 | Show List Products | Admin | N/A |
| UC-21 | Update Product | Admin | N/A |
| UC-22 | Delete Product | Admin | N/A |
| UC-23 | Create Brand | Admin | N/A |
| UC-24 | Show List Brands | Admin | N/A |
| UC-25 | Update Brand | Admin | N/A |
| UC-26 | Delete Brand | Admin | N/A |
| UC-27 | Create Category | Admin | N/A |
| UC-28 | Show List Categories | Admin | N/A |
| UC-29 | Update Category | Admin | N/A |
| UC-30 | Delete Category | Admin | N/A |
| UC-31 | Create Account | Admin | N/A |
| UC-32 | Show List Account | Admin | N/A |
| UC-33 | Update Account | Admin | N/A |
| UC-34 | Delete Account | Admin | N/A |
| UC-35 | Show List Feedback | Admin | N/A |
| UC-36 | Delete Feedback | Admin | N/A |
| UC-37 | Show List Orders | Admin | N/A |
| UC-38 | Update Orders | Admin | N/A |
| UC-39 | Dashboard | Admin | N/A |

Table 4 – Use Case List

* 1. **Use Case Specification**
     1. **Guest**
        1. **Register**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-01: Register | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Guest | Secondary Actor: | N/A |
| Trigger: | The user wants to create a new account on the Student Deal system. | | |
| Description: | This use case describes the process of registering a new account on the Student Deal system. | | |
| Preconditions: | - The user is a guest.  - The Student Deal system must be operational. | | |
| Postconditions: | - The user's account is created and stored in the system.  - The user is granted access to the system and logged in automatically.  - The user is redirected to the homepage.  - Name only contains alphabetical characters.  - Phone number only contains numeric characters.  - Phone number length limit must be 10 characters. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user navigates to the registration page. | |
| 2 | The system presents the registration page. | |
| 3 | The user fills in the required information such as name, email, password, phone number, address. | |
| 4 | The user clicks on the “SIGN UP” button.  [Ex 4a] | |
| 5 | The system validates the user's input and creates a new account in the system. | |
| 6 | The system sends a verification email to the user's email address. | |
| 7 | The user clicks on the verification link in the email. | |
| 8 | The system verifies the user's email address and activates the user's account. | |
| 9 | The system logs the user in automatically. | |
| 10 | The system redirects the user to the homepage. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 4a | The system shows the message “You can't leave this empty” or “You must set your birthday!!!” when you click “SIGN UP” without filling or not choose the date. | |
| 5a | The system shows the message “[your email] is existed!!!” when you input an existed email. | |
| 6a | The system shows the message “Confirm must be matched password” when you input password does not match. | |
| 7a | The system take user back to “empty” register page. | |
| Priority: | High | | |
| Frequency of Use: | Infrequently, as users typically only need to register once. | | |
| Business Rule: | - The password length must be from 6 – 30 characters.  - The user must provide valid and complete information during registration. The user's email address must be verified before the account can be activated. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + 1. **Customer**
       1. **Login**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-02: Login | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Admin |
| Trigger: | The user wants to access the website and perform an action that requires authentication. | | |
| Description: | This use case describes the process of logging into the Student Deal system. | | |
| Preconditions: | - The user must have a registered account.  - The user must have access to the internet.  - The Student Deal system must be operational. | | |
| Postconditions: | - The user is authenticated and granted access to the system.  - The user's personal information and account data are retrieved from the system.  - The user is redirected to the homepage. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user navigates to the login page. | |
| 2 | The system presents the login page. | |
| 3 | The user enters the email and password.  [Alt 3a] | |
| 4 | The user clicks on the “login” button. | |
| 5 | The system verifies the entered email and password. [Ex 5a] | |
| 6 | The system authenticates the user and grants access to the system. | |
| 7 | The system retrieves the user's personal information and account data. | |
| 8 | The system redirects the user to the homepage. | |
| Alternative Flow: | **Step** | **Action** | |
| 3a | The user clicks on “Login by Google”. | |
| 4a | The system will pop up the Gmail window. | |
| 5a | The user input valid Gmail account to be verified by Gmail system. | |
| 6a | The Gmail returns the login successful status to our system. | |
| 7a | Back to Main Flow step 6. | |
| Exceptions: | **Step** | **Cause** | |
| 5a | The system shows the message “Invalid username or password”. | |
| 6a | The system take user back to “empty” login page. | |
| Priority: | High | | |
| Frequency of Use: | Every time the user wants to access the Student Deal system. | | |
| Business Rule: | - The user's email and password must match the ones stored in the system's database.  - The password is more than 6 characters and under 30 characters.  - Email must follow the right format. | | |
| Other Information: | N/A | | |
| Assumptions: | The user has already created an account with the Student Deal system. | | |

* + - 1. **Login by Google Account**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-03: Login by Google Account | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Admin |
| Trigger: | The user wants to access the website and perform an action that requires authentication. | | |
| Description: | This use case describes the process of logging into the Student Deal system. | | |
| Preconditions: | - The user must have access to the internet.  - The Student Deal system must be operational. | | |
| Postconditions: | - The user is authenticated and granted access to the system.  - The user's personal information and account data are retrieved from the system.  - The user is redirected to the homepage. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user navigates to the login page. | |
| 2 | The system presents the login page. | |
| 3 | The user clicks on the “LOGIN” button. | |
| 4 | The user clicks on “Login with Google”. | |
| 5 | The system will pop up the Gmail window. | |
| 6 | The user input valid Gmail account to be verified by Gmail system. | |
| 7 | The Gmail returns the login successful status to our system. | |
| 8 | The system authenticates the user and grants access to the system. | |
| 9 | The system retrieves the user's personal information and account data. | |
| 10 | The system redirects the user to the homepage. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Every time the user wants to access the Student Deal system. | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Logout**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-04: Logout | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Admin |
| Trigger: | The Customer chooses to log out of their account. | | |
| Description: | This use case allows a customer to log out of their account, ending their current session on the website. | | |
| Preconditions: | The Customer must be logged in to their account. | | |
| Postconditions: | The Customer is successfully logged out and their session is ended. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The Customer clicks on the "Log out" link. | |
| 2 | The system ends the Customer's session and logs them out. | |
| 3 | The system redirects the Customer to the home page. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rule: | Customers must be logged in to access certain features of the website, such as viewing their account information, cart, and orders history. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Reset Password**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-05: Reset Password | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | N/A |
| Trigger: | The user has forgotten their password and wants to reset it. | | |
| Description: | This use case describes the process of resetting a user's password on the Student Deal system. | | |
| Preconditions: | - The user is customer.  - The Student Deal system must be operational. | | |
| Postconditions: | - The user's password is updated in the system.  - The user is logged in automatically and redirected to the homepage. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user clicks on the "Forgot password?" link. | |
| 2 | The system presents the password recovery page. | |
| 3 | The user enters their email address then clicks “Send”. [Ex 3a] | |
| 4 | The system sends an email to the user's email address with a verify code with 4 digits. | |
| 5 | The user input 4 digits to verify. [Ex 5b] | |
| 6 | The system presents the password reset page. | |
| 7 | The user clicks on the "Save" button. [Ex 7c] | |
| 8 | The system validates the new password and updates the user's password in the system. | |
| 9 | The user enters a new password and confirms it. [Ex 9d] | |
| 10 | The system logs the user in automatically. | |
| 11 | The system redirects the user to the homepage. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 3a | The system shows the message “Sorry, this email is not sign up” when you input not an existed email. | |
| 5b | The system shows the message “Sorry, code is not wrong, please recheck and try again!” when you input wrong code | |
| 7c | The system shows the message “You can't leave this empty” when you click “Save” without filling. | |
| 9d | The system shows the message “Password is required input from 6 to 30 characters” when you input under 6 characters or more than 30 characters. | |
| 10d | The system shows the message “Confirm must be matched password” when you input password does not match. | |
| 11d | The system take user back to “empty” Password Recovery page. | |
| Priority: | High | | |
| Frequency of Use: | Infrequently, as users typically remember their passwords. | | |
| Business Rule: | The user's email and password must match the ones stored in the system's database.  - The password is more than 6 characters and under 30 characters.  - The user must provide a valid email address to initiate the password reset process. The new password and confirmation must match. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Search**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-06: Search | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user enters a search query in the search bar. | | |
| Description: | This use case describes the process of searching for products on the e-commerce website. | | |
| Preconditions: | - The user has internet access.  - The user is on the Student Deal.  - The user has entered a search query in the search bar. | | |
| Postconditions: | The search results are displayed on the screen. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user enters a search query in the search bar. | |
| 2 | The system validates the search query. | |
| 3 | The system retrieves the products that match the search query. | |
| 4 | The system displays the search results on the screen. | |
| 5 | The user can view the details of the products or add them to their cart. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rule: | - The search function should be fast and accurate.  - The search results should only display products that are currently available in the inventory. | | |
| Other Information: | - The search function can also include advanced filters such as brand, category, points, etc.  - The search results can be sorted by relevance, price, popularity, etc. | | |
| Assumptions: | The user is familiar with basic search functions and can enter a search query in the search bar. | | |

* + - 1. **Search by Filter**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-07: Search by Filter | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user wants to search for products based on specific filters. | | |
| Description: | This use case describes the process of searching for products on the Student Deal system using specific filters. | | |
| Preconditions: | - The user is either a guest or a customer.  - The Student Deal system must be operational. | | |
| Postconditions: | The system displays a list of products that match the user's search criteria. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user clicks on the search bar. | |
| 2 | The system presents the search page. | |
| 3 | The user enters a search term in the search bar | |
| 4 | The user selects one or more filters (e.g. category, brand, points, etc.). | |
| 5 | The user clicks on the "search" button. | |
| 6 | The system searches for products that match the user's search criteria. | |
| 7 | The system displays a list of products that match the user's search criteria. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Frequently, as users often search for products on the Student Deal system. | | |
| Business Rule: | At least one filter must be selected to perform a search. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Shows Product**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-08: Show Products | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user wants to view the available products. | | |
| Description: | This use case describes the process of displaying all the products available on the Student Deal system. | | |
| Preconditions: | - The user is either a guest or a customer.  - The Student Deal system must be operational. | | |
| Postconditions: | The system displays a list of all available products. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user clicks on the logo Student Deal on the homepage. [Alt 1a] | |
| 2 | The system displays a list of all available products. | |
| Alternative Flow: | **Step** | **Action** | |
| 1a | The user clicks on “Categories” on navigation bar. | |
| 2a | The system displays a list of all available brands. | |
| 3a | The user clicks on one brand on navigation bar. | |
| 4a | The system displays a list of all available products of brand. | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Frequently, as users often want to view the available products on the Student Deal system. | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Pagination**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-09: Pagination | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user wants to navigate through a list of products. | | |
| Description: | This use case describes the process of navigating through a list of products displayed on the Student Deal system using pagination. | | |
| Preconditions: | - The user is either a guest or a customer.  - The Student Deal system must be operational.  - There are more products to display beyond the current page. | | |
| Postconditions: | The system displays the next or previous page of products. | | |
| Normal Flow: | Step | Action | |
| 1 | The user clicks on the number of pages at the bottom of the products list. | |
| 2 | The system loads and displays the next or previous page of products. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Frequently, as users often want to navigate through a long list of products. | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **View Cart**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-10: View Cart | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user wants to review the items that they have added to their cart. | | |
| Description: | This use case describes how a user can view their cart contents. | | |
| Preconditions: | - The user has logged in to their account.  - The user has added at least one product to their cart. | | |
| Postconditions: | - The user can view the products in their cart.  - The user can continue shopping, update their cart, or proceed to checkout. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user clicks on the "Cart" button in the navigation bar. [Alt 1a] | |
| 2 | The system displays the user's cart with number of products. | |
| Alternative Flow: | **Step** | **Action** | |
| 1a | The user clicks on “VIEW CART” button. | |
| 2a | The system displays the user's cart with number of products, all detail of the product and a text box for customer to take a note. | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Multiple times per visit | | |
| Business Rule: | - The system shall display the user's cart contents accurately and in real-time.  - The system shall allow the user to edit their cart before proceeding to checkout.  - The system shall save the user's cart contents when they exit the browser and restore it when they return to the website. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Add to Cart**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-11: Add to Cart | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user wants to add a product to their cart. | | |
| Description: | This use case describes the process of adding a product to the customer's cart on the Student Deal system. | | |
| Preconditions: | The Student Deal system must be operational. | | |
| Postconditions: | The product is added to the user's cart. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user navigates to the product they wish to add to their cart. | |
| 2 | The user clicks the "Add to Cart" button on the product page. | |
| 3 | The system adds the product to the customer's cart and displays a message indicating the product has been added. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Frequently, as users often add products to their cart when shopping on the Student Deal system. | | |
| Business Rule: | The product when customer add to cart must be in stocking. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Edit Cart**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-12: Edit Cart | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user wants to edit the contents of their cart. | | |
| Description: | This use case describes the process of editing the contents of the user's cart on the Student Deal system. | | |
| Preconditions: | - The user has at least one product in their cart.  - The Student Deal system must be operational. | | |
| Postconditions: | - The user's cart is updated with the edited contents. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user navigates to their cart. | |
| 2 | The user clicks the "VIEW CART" button. | |
| 3 | The system displays the contents of the user's cart. | |
| 4 | The user can update the quantity or remove a product from their cart. | |
| 5 | The user can take a note for store. | |
| 6 | The system updates the user's cart and displays a message indicating the cart has been updated. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasionally, as customers may need to edit their cart before checkout. | | |
| Business Rule: | The quantity of products on cart must be positive. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Delete Item from Cart**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-13: Delete Item from Cart | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | Guest |
| Trigger: | The user wants to remove an item from their cart. | | |
| Description: | This use case describes the process of removing an item from the user's cart on the Student Deal system. | | |
| Preconditions: | The user has at least one item in their cart.  The Student Deal system must be operational. | | |
| Postconditions: | The item is removed from the user's cart. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The user navigates to their cart. | |
| 2 | The user clicks the "X" button under the product they wish to remove. | |
| 3 | The system shows the pop-up windows then user choosing “Yes” or “No” [Alt 3a, 3b] | |
| Alternative Flow: | **Step** | **Action** | |
| 3a | The user clicks on “Yes” on the pop-up windows. | |
| 4a | The system removes the item from the user's cart and displays a message indicating the item has been removed. | |
| 3b | The user clicks on “No” on the pop-up windows. | |
| 4b | The system take user back to customer’s cart screen. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasionally, as users may need to remove items from their cart before checkout. | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Checkout**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-14: Checkout | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | N/A |
| Trigger: | Customer has added one or more products to their cart and wants to initiate the checkout process. | | |
| Description: | This use case allows a customer to enter their shipping and get voucher information and complete the purchase of the products in their cart. | | |
| Preconditions: | - Customer must be logged in.  - Customer must have one or more products in their cart. | | |
| Postconditions: | - The system creates a new order with the products in the cart.  - The system sends an order confirmation email to the customer. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | Customer selects the "GET VOUCHER" button from the cart page. | |
| 2 | System displays a checkout page with fields for shipping and get voucher information. | |
| 3 | Customer enters their shipping and get voucher information. | |
| 4 | Customer confirms the order details and clicks the "Place Order" button. | |
| 5 | System validates the information and charges the get voucher method. | |
| 6 | System creates a new order with the products in the cart and marks them as "sold". | |
| 7 | System sends an order confirmation email to the customer with the order details. | |
| 8 | System redirects the customer to a page showing the order details. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Very high | | |
| Business Rule: | - The system must charge the customer's get voucher method before marking the products as sold.  - The system must validate the customer's shipping and get voucher information before processing the order. | | |
| Other Information: | - The checkout page should display a summary of the order details, including the products, shipping and get voucher information, and order total.  - The order confirmation email should include the order number, order date, list of products, shipping and billing address, and order total. | | |
| Assumptions: | N/A | | |

* + - 1. **Show Customer Information**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-15: Show Customer Information | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | N/A |
| Trigger: | The customer wants to view their account information. | | |
| Description: | This use case describes the process of displaying the customer's account information on the Student Deal system. | | |
| Preconditions: | - The customer is logged in to their account.  - The Student Deal system must be operational. | | |
| Postconditions: | - The customer's account information is displayed. | | |
| Normal Flow: | Step | Action | |
| 1 | The customer clicks the "My Account" link. | |
| 2 | The system displays the customer's account information, including name, email, birthday, sex, phone, address, rank. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasionally, as customers may want to view or update their account information. | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Edit Customer Information**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-16: Edit Customer Information | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | N/A |
| Trigger: | The customer wants to update their account information. | | |
| Description: | This use case describes the process of allowing the customer to edit their account information on the Student Deal system. | | |
| Preconditions: | - The customer is logged in to their account.  - The Student Deal system must be operational. | | |
| Postconditions: | - The customer's account information is updated. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The customer clicks the "My Account" link. | |
| 2 | The system displays the customer's account information. | |
| 3 | The customer clicks the "Update Self-information" link. | |
| 4 | The system displays the form for editing that information. | |
| 5 | The customer makes their changes and clicks the "Confirm" button. | |
| 6 | The system updates the customer's account information. [Ex 6a] | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 6a | The system shows the message “You can't leave this empty” or “invalid phone number!!” when you click “CONFIRM” without filling. | |
| 7a | The system shows the message “Confirm must be matched password” when you input password does not match. | |
| 8a | The system take user back to “empty” update account page. | |
| Priority: | Medium | | |
| Frequency of Use: | Occasionally, as customers may want to update their account information. | | |
| Business Rule: | - The password length must be from 6 – 30 characters.  - Name only contains alphabetical characters.  - Phone number only contains numeric characters.  - Phone number length limit must be 10 characters. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Show Orders History**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-17: Show Orders History | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | N/A |
| Trigger: | The customer wants to view their order history. | | |
| Description: | This use case describes the process of displaying the customer's order history on the Student Deal system. | | |
| Preconditions: | - The customer is logged in to their account.  - The Student Deal system must be operational. | | |
| Postconditions: | - The customer's order history is displayed. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The customer clicks the "My Account" link. | |
| 2 | The system displays the customer's account information. | |
| 3 | The customer clicks the "My Orders" button. | |
| 4 | The system displays a list of the customer's previous orders, including order number, date, status, and total cost. | |
| 5 | The customer can click on a specific order to view its details, including products purchased, quantities, and cost. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasionally, as customers may want to review their past purchases. | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Leave Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-18: Show Orders History | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Customer | Secondary Actor: | N/A |
| Trigger: | The customer wants to leave feedback about a product or service. | | |
| Description: | This use case describes the process of allowing the customer to leave feedback on a specific product or the overall service provided by Student Deal. | | |
| Preconditions: | - The customer is logged in to their account.  - The customer has purchased a product or used a service on the Student Deal. | | |
| Postconditions: | - The feedback is submitted and saved to the system.  - The system notifies the admin of the new feedback. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The customer navigates to the product they want to leave feedback on. | |
| 2 | The system displays a form for the customer to enter their feedback, including a rating and text field. | |
| 3 | The customer fills in the form and submits it. [Ex 3a] | |
| 4 | The system saves the feedback to the customer's account and notifies the admin of the new feedback. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 3a | The system shows the message “You can't leave this empty” when you click “CONFIRM” without filling. | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional, as customers may want to provide feedback. | | |
| Business Rule: | - Feedback must be appropriate and relevant to the product or service provided by Student Deal.  - Feedback must not contain any inappropriate content. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + 1. **Admin**
       1. **Manage Product**
          1. **Create Product**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-19: Create Product | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to create a new product. | | |
| Description: | This use case describes the steps involved in creating a new product in the STUDENT DEAL. | | |
| Preconditions: | The admin must be logged in and authorized to create a new product. | | |
| Postconditions: | A new product is added to the product database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Product" option from the dashboard. | |
| 2 | The admin clicks the "Add a new Product" option from the dashboard. | |
| 3 | The system displays a form for the admin to fill in the details of the new product. | |
| 4 | The admin fills in the product details such as name, description, quantity, size, price, description, update image and choose category, brand. [Ex 4a] | |
| 5 | The admin clicks “Add” button. | |
| 6 | The system validates the form data and creates the new product in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 5a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 6a | The system take user back to “empty” create product pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | The product must have a unique name and must be associated with an existing category and brand in the database. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Show List Products**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-20: Show List Product | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to view the list of all products. | | |
| Description: | This use case describes the steps involved in viewing the list of all products in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of products. | | |
| Postconditions: | The system displays the list of all products in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Product" option from the dashboard. | |
| 2 | The system displays the details of the list of products. [Ex 2a] | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 2a | If there are no products in the database, the system displays a message indicating that there are no products available. | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Update Product**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-21: Update Product | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to update an existing product. | | |
| Description: | This use case describes the steps involved in update a product in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of products. | | |
| Postconditions: | The selected product is updated in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Product" option from the dashboard. | |
| 2 | The system displays a list of all products in the database. | |
| 3 | The admin Click the “Update Product” icon of they want to update from the list. | |
| 4 | The system displays the details of the selected product in an editable form. | |
| 5 | The admin updates the product details they want to change. | |
| 6 | The admin submits the updated form. [Ex 6a] | |
| 7 | The system validates the form data and updates the product in the database. | |
| 8 | The system displays a success message to the Admin. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 6a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 7a | The system take user back to “empty” update product pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Delete Product**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-22: Delete Product | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to delete an existing product. | | |
| Description: | This use case describes the steps involved in delete a product in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of products. | | |
| Postconditions: | The selected product is deleted. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Product" option from the dashboard. | |
| 2 | The system displays a list of all products in the database. | |
| 3 | The admin Click the “Delete Product” icon of they want to delete from the list. | |
| 4 | The system displays the pop-up windows to confirm. | |
| 5 | The system updates the status of product in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | Delete product is mean update the status of product | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Manage Brand**
         1. **Create Brand**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-23: Create Brand | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to create a new brand. | | |
| Description: | This use case describes the steps involved in creating a new brand in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to create a new brand. | | |
| Postconditions: | The new brand is added to the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Brand" option from the dashboard. | |
| 2 | The admin clicks the "Add a new Brand" option from the dashboard. | |
| 3 | The system displays a form for the admin to fill in the details of the new brand. | |
| 4 | The admin fills in the brand details such as name, description. [Ex 4a] | |
| 5 | The admin clicks “Add” button. | |
| 6 | The system validates the form data and creates the new brand in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 5a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 6a | The system take user back to “empty” create a brand pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Show List Brands**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-24: Show List Brand | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to view the list of all brands. | | |
| Description: | This use case describes the steps involved in viewing the list of all brands in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of brands. | | |
| Postconditions: | The system displays the list of all brands in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Brand" option from the dashboard. | |
| 2 | The system displays the details of the list of brands. [Ex 2a] | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 2a | If there are no brands in the database, the system displays a message indicating that there are no brands available. | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Update Brand**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-25: Update Brand | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to update an existing brand. | | |
| Description: | This use case describes the steps involved in update a brand in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of brands. | | |
| Postconditions: | The selected brand is updated in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Brand" option from the dashboard. | |
| 2 | The system displays a list of all brands in the database. | |
| 3 | The admin clicks the “Update Brand” icon of they want to update from the list. | |
| 4 | The system displays the details of the selected brand in an editable form. | |
| 5 | The admin updates the brand details they want to change. | |
| 6 | The admin submits the updated form. [Ex 6a] | |
| 7 | The system validates the form data and updates the brand in the database. | |
| 8 | The system displays a success message to the admin. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 6a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 7a | The system take user back to “empty” update brand pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Delete Brand**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-26: Delete Brand | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to delete an existing brand. | | |
| Description: | This use case describes the steps involved in delete a brand in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of brands. | | |
| Postconditions: | The selected brand is deleted. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Brand" option from the dashboard. | |
| 2 | The system displays a list of all brands in the database. | |
| 3 | The admin Click the “Delete Brand” icon of they want to delete from the list. | |
| 4 | The system displays the pop-up windows to confirm. | |
| 5 | The system updates the status of brand in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | Delete brand is mean update the status of brand | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Manage Category**
         1. **Create Category**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-27: Create Category | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to create a new category. | | |
| Description: | This use case describes the steps involved in creating a new category in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to create a category. | | |
| Postconditions: | The new category is added to the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Category" option from the dashboard. | |
| 2 | The admin clicks the "Add a new Category" option from the dashboard. | |
| 3 | The system displays a form for the admin to fill in the details of the new categories. | |
| 4 | The admin fills in the category details such as name, description. [Ex 4a] | |
| 5 | The admin clicks “Add” button. | |
| 6 | The system validates the form data and creates the new category in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 5a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 6a | The system take user back to “empty” create category pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Show List Categories**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-28: Show List Categories | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to view the list of all categories. | | |
| Description: | This use case describes the steps involved in viewing the list of all categories in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of categories. | | |
| Postconditions: | The system displays the list of all categories in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Category" option from the dashboard. | |
| 2 | The system displays the details of the list of categories. [Ex 2a] | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 2a | If there are no categories in the database, the system displays a message indicating that there are no categories available. | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Update Category**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-29: Update Category | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to update an existing category. | | |
| Description: | This use case describes the steps involved in update a category in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of categories. | | |
| Postconditions: | The selected category is updated in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Category" option from the dashboard. | |
| 2 | The system displays a list of all categories in the database. | |
| 3 | The admin Click the “Update Category” icon of they want to update from the list. | |
| 4 | The system displays the details of the selected category in an editable form. | |
| 5 | The admin updates the category details they want to change. | |
| 6 | The admin submits the updated form. [Ex 6a] | |
| 7 | The system validates the form data and update the category in the database. | |
| 8 | The system displays a success message to the admin. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 6a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 7a | The system take user back to “empty” update category pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Delete Category**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-30: Delete Category | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to delete an existing brand. | | |
| Description: | This use case describes the steps involved in delete a category in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of categories. | | |
| Postconditions: | The selected category is deleted. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Category" option from the dashboard. | |
| 2 | The system displays a list of all categories in the database. | |
| 3 | The admin Click the “Delete Category” icon of they want to delete from the list. | |
| 4 | The system displays the pop-up windows to confirm. | |
| 6 | The system updates the status of category in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | Delete category is mean update the status of category | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Manage Account**
         1. **Create Account**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-31: Create Account | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to create a new account. | | |
| Description: | This use case describes the steps involved in creating a new account in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to create a new account. | | |
| Postconditions: | A new account is added to the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Account" option from the dashboard. | |
| 2 | The admin clicks the "Add a new Account" option from the dashboard. | |
| 3 | The system displays a form for the admin to fill in the details of the new account. | |
| 4 | The admin fills in the account details such as name, phone number, email, password, role. [Ex 4a] | |
| 5 | The admin clicks “Add” button. | |
| 6 | The system validates the form data and creates the new account in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 5a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 6a | The system take user back to “empty” create account pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Show List Accounts**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-32: Show List Account | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to view the list of all accounts. | | |
| Description: | This use case describes the steps involved in viewing the list of all accounts in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of accounts. | | |
| Postconditions: | The system displays the list of all accounts in the database. | | |
| Normal Flow: | Step | Action | |
| 1 | The admin selects the "List Accounts" option from the dashboard. | |
| 2 | The system displays the details of the list of categories. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rule: | The database must have at least one account of admin. | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Update Account**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-33: Update Account | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to update an existing account. | | |
| Description: | This use case describes the steps involved in update an account in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of accounts. | | |
| Postconditions: | The selected account is updated in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Account" option from the dashboard. | |
| 2 | The system displays a list of all accounts in the database. | |
| 3 | The admin Click the “Update Account” icon of they want to update from the list. | |
| 4 | The system displays the details of the selected account in an editable form. | |
| 5 | The admin updates the account details they want to change. | |
| 6 | The admin submits the updated form. [Ex 6a] | |
| 7 | The system validates the form data in the database. | |
| 8 | The system displays a success message to the Admin. | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 6a | The system shows the message “You can't leave this empty” when not filling the form. | |
| 7a | The system take user back to “empty” update account pop-up window. | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Delete Account**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-34: Delete Account | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to delete an existing account. | | |
| Description: | This use case describes the steps involved in delete an account in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of accounts. | | |
| Postconditions: | The selected account is deleted. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Account" option from the dashboard. | |
| 2 | The system displays a list of all accounts in the database. | |
| 3 | The admin Click the “Delete Account” icon of they want to delete from the list. | |
| 4 | The system displays the pop-up windows to confirm. | |
| 5 | The admin updates status of the account they want to delete. | |
| 6 | The system updates the status of account in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | Delete account is mean update the status of account | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Manage Feedback**
         1. **Show List Feedbacks**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-35: Show List Feedback | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to view the list of customer feedback. | | |
| Description: | This use case describes the steps involved in displaying the list of feedbacks given by customers in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the feedbacks list. | | |
| Postconditions: | The list of feedbacks is displayed to the admin. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Feedback" option from the dashboard. | |
| 2 | The system displays a list of all feedback given by customers. [Ex 2a] | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 2a | If there is no feedback available, the system displays a message that there is no feedback to display. | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Delete Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-36: Delete Feedback | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to delete existing feedback. | | |
| Description: | This use case describes the steps involved in delete feedback in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of feedbacks. | | |
| Postconditions: | The selected feedback is deleted. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Feedback" option from the dashboard. | |
| 2 | The system displays a list of all feedbacks in the database. | |
| 3 | The admin clicks the “Delete Feedback” icon of they want to delete from the list. | |
| 4 | The system displays the pop-up windows to confirm. | |
| 5 | The admin updates status of the feedback they want to delete. | |
| 6 | The system updates the status of feedback in the database. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | Delete account is mean update the status of account | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - 1. **Manage Orders**
         1. **Show List Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-37: Show List Orders | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to view the list of customer order. | | |
| Description: | This use case describes the steps involved in displaying the list of orders given by customers in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the orders list. | | |
| Postconditions: | The list of orders is displayed to the admin. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Order" option from the dashboard. | |
| 2 | The system displays a list of all orders given by customers. [Ex 2a] | |
| Alternative Flow: | N/A | | |
| Exceptions: | **Step** | **Cause** | |
| 2a | If there is no order available, the system displays a message that there is no order to display. | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Update Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-38: Update Orders | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to update an existing order. | | |
| Description: | This use case describes the steps involved in update an order in the Student Deal. | | |
| Preconditions: | The admin must be logged in and authorized to view the list of orders. | | |
| Postconditions: | The selected order is updated in the database. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin selects the "List Order" option from the dashboard. | |
| 2 | The system displays a list of all orders in the database. | |
| 3 | The admin Click the “Update Order” icon of they want to update from the list. | |
| 4 | The system displays the details of the selected order in an editable form. | |
| 5 | The admin updates the order status they want to change. | |
| 6 | The admin submits the updated form. [Ex 6a] | |
| 7 | The system validates the form data in the database. | |
| 8 | The system displays a success message to the Admin. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Moderate | | |
| Business Rule: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

* + - * 1. **Dashboard**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name | UC-39: Dashboard | | |
| Created By: | KhiemNG | Date Created: | 12/07/2023 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | The admin wants to access the dashboard of the Student Deal. | | |
| Description: | This use case describes the steps involved in accessing the dashboard of the Student Deal. The dashboard provides a summary of the store's performance, including sales, revenue, inventory, and customer data. | | |
| Preconditions: | The admin must be logged in and authorized to access the dashboard. | | |
| Postconditions: | The dashboard is displayed with the relevant data. | | |
| Normal Flow: | **Step** | **Action** | |
| 1 | The admin logs into the Student Deal. | |
| 2 | The system validates the admin’s login credentials and redirects them to the dashboard page. | |
| 3 | The system displays a summary of the store's performance, including sales, revenue, inventory, and customer data. | |
| 4 | The admin can view and analyze the data presented on the dashboard. | |
| 5 | The admin can access other functions of the store, such as creating or editing products, from the dashboard. | |
| Alternative Flow: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Daily | | |
| Business Rule: | The dashboard must be updated regularly to provide accurate and relevant data to the Admin. | | |
| Other Information: | The dashboard should be designed with a user-friendly interface to enable easy navigation and analysis of data. | | |
| Assumptions: | The admin has the necessary technical skills to analyze and interpret the data presented on the dashboard. | | |

* 1. **Functional Requirements**

3. 1. **System Functional Overview**
      1. **Screen Flow**

**Graphical user interface

Description automatically generated with low confidence**

Figure 3 – Screen Flow

* + 1. **Screen Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Feature** | **Screen** | **Description** |
| SD-01 | View Main Page | Main Page | Conclude the search bar, Cart button, Customer information button |
| SD-02 | Get voucher List of Product | Main Page | List of product load from database |
| SD-03 | Get voucher List of Product Detail | Search Page | List of product detail |
| SD-04 | Get voucher Customer Cart | Cart Page | List of products on cart |
| SD-05 | Get voucher Customer Cart Detail | Cart Page | Conclude plus button, minus button and delete button |
| SD-06 | Login | Login Page | Conclude button login Google |
| SD-07 | View Customer Information | Customer Information Page | List the information of customer |
| SD-08 | View Order | Checkout Page | List of the product of customer pay points |
| SD-09 | Cancel Order | Customer Information Page | Conclude Cancel Order button |
| SD-10 | Get voucher List Account | List Account Page | List of accounts on database |
| SD-11 | Get voucher List Product | List Product Page | List of products on database |
| SD-12 | Get voucher List Brand | List Brand Page | List of brands on database |
| SD-13 | Get voucher List Category | List Category Page | List of categories on database |
| SD-14 | Get voucher List Feedback | List Feedback Page | List of feedbacks on database |
| SD-15 | Get voucher List Orders | List Orders Page | List of orders on database |
| SD-21 | Show Dashboard | Dashboard Page | Conclude date time dropdown, The chart |

Figure 4 – ERD Diagram

* 1. **Non-Functional Requirements**

1. 1. **External Interface**
      1. **Hardware Interface**

* The system shall be accessible from any device with an internet connection, including desktop computers, laptops, tablets, and smartphones.
* The system shall be compatible with major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari.
  + 1. **Software Interface**
* The system shall integrate with a secure get voucher gateway to process online transactions.
* The system shall integrate with third-party authentication providers, such as Google, for user authentication and login.
  + 1. **User Interface**
* The authenticated users UI must be fully shown in web application.
* The user interface shall be designed to be user-friendly and intuitive, with clear navigation and easy-to-use search and filtering functions.
  + 1. **Communication Interface**
* The system can call API through data transferred by HTTP.
* The system shall use encryption to ensure secure communication between the user's device and the server.
* The system shall use email notifications to communicate with users, such as order confirmation and password reset instructions.
  1. **Quality Attribute**
     1. **Performance**
* The system shall load pages and search results quickly, with a response time of no more than 3 seconds.
  + 1. **Usability**
* All text, label and message should be uniformly written in English.
* The web application should be friendly and easy for users to using.
* The web application for admin should require no more than 2 days of training of use.
  + 1. **Security**
* Student Deal is divided into 3 roles: Guest, Customer and Admin.
* All input data are validated before saving to the database.
* Each role only access to a group of functions.
* The system always checks authentication and authorization before making any features.
* The system shall use industry-standard encryption and secure protocols to protect user data and prevent unauthorized access.
* The system shall implement user authentication and access control measures to ensure that only authorized users can perform certain actions, such as editing personal information and managing orders.

1. **Software Design Description**
   1. **Overall Description**
   2. **Assumptions**

* This system is designed basing on these following assumptions:
* Window 10/11
* Microsoft Edge Browser
* SQL Server 2019
* End-User: Customer, project team
* Security requirement: prevent users from cheating as much as possible.
* Network communication: This system depends on network connections.
  1. **Overall System Architecture**



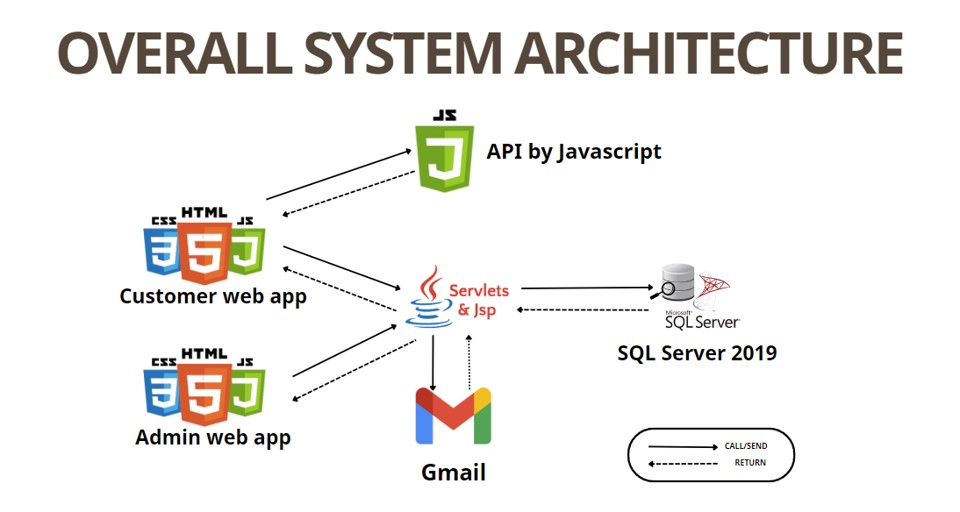
****

Figure 5 – Overall System Architecture

* 1. **Data and Database Design**

1. 1. **Database Design**

Graphical user interface, diagram

Description automatically generated

Figure 6 – Database Design

* 1. **Data File Design**
     1. **Brand Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| BrandID | Int | ID of brand | Yes | Yes | PK |
| Name | Nvarchar(50) | Name of brand | No | No | No |
| Description | Nvarchar(200) | Description of brand | No | No | No |
| CreateTime | Date | Date create brand | No | No | No |
| Status | Bit | Status of brand | No | No | No |

* + 1. **Cart Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| CustomerID | Int | ID of customer | Yes | Yes | FK |
| ProductID | Int | ID of product | No | Yes | FK |
| Quantity | Int | Quantiy of product on cart | No | No | No |
| CreateTime | Date | Date create cart | No | No | No |

* + 1. **Category Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| CateID | Int | ID of category | Yes | Yes | PK |
| Name | Nvarchar(50) | Name of category | No | No | No |
| Description | Nvarchar(MAX) | Description of category | No | No | No |
| Status | Bit | Status of category | No | No | No |

* + 1. **Customer Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| CustomerID | Int | ID of customer | Yes | Yes | PK |
| Name | Nvarchar(50) | Name of customer | No | No | No |
| Password | Varchar(50) | Password of customer | No | No | No |
| DateOfBirth | Date | Birth date of customer | No | No | No |
| Email | Varchar(50) | Email of customer | No | No | No |
| Phone | Varchar(50) | Phone number of customer | No | No | No |
| Address | Nvarchar(150) | Address of customer | No | No | No |
| Role | bit | Role of users | No | No | No |
| RankID | Int | Rank of customer | No | No | FK |
| Sex | Bit | Sex of customer | No | No | No |
| TypeOfLogin | int | Login method of customer | No | No | No |

* + 1. **Feedback Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| ProductID | Int | ID of product | Yes | Yes | FK |
| CustomerID | Int | ID of customer | Yes | Yes | FK |
| Voting | Int | Scale of feedback | No | No | No |
| Status | Bit | Status of feedback | No | No | No |
| FeedBackTime | Date | Date feedback | No | No | No |
| TextComments | Nvarchar(MAX) | Comment of feedback | No | No | No |

* + 1. **Manager Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| AdminID | Int | ID of admin | Yes | Yes | PK |
| Username | Varchar(50) | Full name of admin | No | No | No |
| Name | Varchar(50) | Name of admin | No | No | No |
| Password | Varchar(50) | Password of admin | No | No | No |

* + 1. **MembershipTier Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| RandID | Int | ID of rank | Yes | Yes | PK |
| Rank | Varchar(50) | Name of rank | No | No | No |
| Description | Nvarchar(MAX) | Description of rank | No | No | No |
| Discount | Float | Number of discount | No | No | No |
| Minimum\_Coins | Int | Point of discount | No | No | No |

* + 1. **Orders Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| OrdersID | Int | ID of order | Yes | Yes | PK |
| CustomerID | Int | ID of customer | No | No | No |
| Get vouchermentID | Int | ID of get voucherment method | No | No | No |
| ShippingID | Int | ID of shipping method | No | No | No |
| DateOrders | Date | Date order | No | No | No |
| Status | Int | Status of order | No | No | No |
| CusName | Nvarchar(MAX) | Customer name | No | No | No |
| CusPhone | Nvarchar(MAX) | Customer phone number | No | No | No |
| CusAddress | Nvarchar(MAX) | Customer address | No | No | No |

* + 1. **OrdersDetail Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| OrdersDtID | Int | ID of ordersDetail | Yes | Yes | PK |
| ProductID | Int | ID of product | No | No | No |
| OrdersID | Int | ID of order | No | No | No |
| Quantity | Int | Quantity of order | No | No | No |
| Discount | Float | Amount of discount | No | No | No |
| Price | Float | Product price of order | No | No | No |
| Total | Float | Total price of order | No | No | No |
| Status | Bit | Status of ordersDetail | No | No | No |

* + 1. **Get vouchermentMethod Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| Get vouchermentID | Int | ID of get voucherment method | Yes | Yes | PK |
| NameOfMethod | Varchar(50) | Name of get voucherment method | No | No | No |

* + 1. **Product Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| ProductID | Int | ID of product | Yes | Yes | PK |
| Name | Nvarchar(MAX) | Name of product | No | No | No |
| Description | Nvarchar(MAX) | Description of product | No | No | No |
| Quantity | Int | Quantity of product | No | No | No |
| Price | Float | Price of product | No | No | No |
| Status | Bit | Status of product | No | No | No |
| Size | Int | Size of product | No | No | No |
| CreateTime | Date | Date create product | No | No | No |
| CateID | Int | Category ID of product | No | No | No |
| BrandID | Int | Brand ID of product | No | No | No |
| Avatar | Varchar(MAX) | Image 1 of product | No | No | No |
| Avatar2 | Varchar(MAX) | Image 2 of product | No | No | No |
| SizeID | Int | Size of product | No | No | No |

* + 1. **ProductSize Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| SizeID | Int | ID of size | Yes | Yes | PK |
| Size | Int | Number of sizes | No | No | No |
| Status | Bit | Status of size | No | No | No |