Software Design Document (SWDD) Template

Background

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SWDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SWDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage—i.e., the detailed design stage—more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the *IEEE Recommended Practice for Software Design Descriptions*. The *IEEE Recommended Practice for Software Design Descriptions* have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to IEEE Std 1016 for the full *IEEE Recommended Practice for Software Design Descriptions*.

Helarica
HUST-PC Website for IT equipments
Software Design Document

Name (s):
Lab Section:
Workstatio
n:

Date: (mm/dd/yyyy)

Mục lục

1. INTRODUCTION

1.1 Purpose

The purpose of the HUST-PC system is mainly to support the management and sales of computer stores. This system provides functions for product search, store information management, product management, employee management, order processing, invoicing, and other process-related functions. business in the computer store.

Accordingly, the final product must be quick, efficient, and extremely easy to use. It must offer useful features without overwhelming the user with options. The user interface must be intuitive and have little or no learning curve.

1.2 Scope

The software identified in this document is a website dedicated to the sale of assembled computers and information technology equipment, named HUST-PC. The main purpose of the software is to provide an online sales platform so that customers can customize their computer configuration and purchase information technology products according to their needs.

HUST-PC Software is geared towards the following benefits, objectives, and purposes:

- Benefit to Customers:

  - Custom configuration: Customers have the ability to customize the computer configuration according to personal requirements and preferences.

  - Diversified choice: The software provides a wide range of computer products and information technology equipment for customers to have a diverse choice.

  - Details: Customers can view detailed information about each product, including specifications, photos, and customer reviews.

- Benefit to Business:

  - Increase sales: HUST-PC software allows customers to easily buy goods online, creating favorable conditions to increase sales.

  - Branding: By providing quality service and a good shopping experience, HUST-PC aims to build a prestigious brand in the field of fast-assembled computers and information technology equipment.

  - Expand market: Software helps businesses expand their market to reach customers online, not limited by geography.

HUST-PC software is closely linked to the company's strategic goal, to build a leading brand in the field of selling quick-assembled computers and information technology equipment. It assists in increasing sales, expanding markets, and enhancing customer satisfaction by providing quality service and the best shopping experience.

1.3 Overview

This SDD provides a low-level description of the design aspects of the HUSTPC system. It includes the system context, architectural design, database design, user interface design, implementation plan, and maintenance and support procedures. The document is organized into several sections to provide a solid understanding of the inner workings of the HUSTPC system.

While the software requirement specification (SRS) document is written for a more general audience, this document is intended for individuals directly involved in the development of HUSTPC. This includes software developers, project consultants, and team managers. This document need not be read sequentially; users are encouraged to jump to any section they find relevant. Below is a brief overview of each part of the document:

• Part 1 (Introduction) :This section offers a summary of the HUSTPC project, including the purpose, project scope and some general system details.

• Part 2 (System Overview):This section gives a general description of the functionality, context and design of the HUSTPC project.

• Part 3 (Architectural Design) :This section describes the HUSTPC system class by class, including class hierarchies and performance/design constraints.

• Part 4 (Data Design): Readers interested in how HUSTPC organizes and handles data should consult this section, which covers data structures and flow patterns utilized by the system.

• Part 5 (Component Design): This section will provide a closer look at what each component does in a more systematic way

• Part 6 (Human Interface Design) :This section covers all of the details related to the structure of the graphical user interface (GUI), including some preliminary mockups of the HUSTPC application. Readers can view this section for a tentative glimpse of what the final product will look like.

• Part 7 (Requirement matrix) : Readers interested in the software testing process should consult this section, which offers a list of test cases, expected responses, and other pertinent information.

• Part 8 (Appendices): This section includes any additional information which may be helpful to readers.

1.4 Reference Material

Software Design Documentation - IEEE-1016-Template
https://github.com/Xett/Software-Design-Documentation/blob/master/IEEE-1016%20(SDD)/IEEE-1016-Template.md

Software Design Documentation by Dinesh Thakur

https://ecomputernotes.com/software-engineering/softwaredesigndocumentation

Software Design Document, Testing, and Deployment and Configuration Management Unified University Inventory System (UUIS)

https://arxiv.org/ftp/arxiv/papers/1005/1005.0595.pdf

1.5 Definitions and Acronyms

This section is optional.

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SWDD. These definitions should be items used in the SWDD that are most likely not known to the audience.

2. SYSTEM OVERVIEW

HUSTPC is an online store that specializes in selling IT equipment to individuals and businesses. It aims to provide a convenient and user-friendly platform for customers to browse, select, and purchase a wide range of IT products, including computers, laptops, peripherals, and accessories. The store operates exclusively online, eliminating the need for physical retail locations.

2.1. Features:

The HUSTPC system is composed of two primary components: a client-side application that receives user input and performs interaction with users, and a server-side application which updates and synchronizes data across devices. The system features can be broken up into two groups as well: core features, which are essential to the function of the application, and additional features, which are only meant to add extra functionality.

*Core features:

The following list includes all of the features currently designated for inclusion in the final release of HUSTPC:

1.   Browse the web:

- Function that allows customers to access the website and browse through product pages.

- Customers can search for products, sort them according to different criteria, and view detailed information about the products.

2.   Build PC:

- Feature that allows customers to customize the components of a computer.

   - Customers can search for, view details of, and select components such as the central processing unit (CPU), motherboard, graphics card (GPU), memory (RAM), hard drive (HDD/SSD), and many other components to suit their needs.

3.   Customize products:

   - Feature that allows customers to customize the configuration of products.

   - Customers can view detailed information about the product, select options, and customize the configuration to meet their needs.

   - For customizable products, customers can change elements such as color, size, options, and other details.

- The system should visually display the options and help customers preview the customization changes on the product.

4.   Place order:

- Feature that allows customers to add products to the shopping cart, proceed with payment, and place an order.

- Customers can review the shopping cart, edit the quantity of products, or remove products from the cart before proceeding with payment.

- After the customer clicks the "Place Order" button, the system will display a successful order confirmation page.

5.   Manage products:

- Feature for store managers to manage product information.

- Store owners can add, modify, and delete product information, including descriptions, images, prices, remaining quantities, and other attributes.

6.   Manage orders:

- This feature allows employees and store owners to manage orders.

- Employees and store managers can view a list of orders, detailed information for each order, and the order status (paid, unpaid, in transit, completed, etc.).

*Additional features:

The features below are not guaranteed to be present in the final release of SplitPay, but will be added as time permits. Due to their tentative nature, they will not be covered in this document.

1.   Email Notification

- Automatically delivers notifications via e-mail and/or text message

2.   Language Translation

- The system shall support many languages, translate and display the language that the user desire

3.   Feedback/ Rating System

- This feature allows the cusstomer to leave review about the product

4.   Help Center

- The customer can contact the staff member for support via the Help menu, which displays a list of topics covering the different components of HUSTPC

2.2. Design:

HUSTPC's online store features a clean and intuitive user interface, designed to enhance the shopping experience. The layout is organized logically, with product categories and filters to facilitate easy navigation. The system employs responsive web design, ensuring seamless compatibility across different devices and screen sizes.

To optimize performance and scalability, HUSTPC's system is built on a robust and reliable infrastructure, utilizing cloud-based servers and scalable database solutions. The system leverages modern web technologies and frameworks to deliver fast response times and efficient data retrieval.

Furthermore, the system integrates analytics and reporting tools, allowing HUSTPC to gather insights into customer behavior, popular products, and sales trends. These analytics can inform business decisions, such as marketing campaigns, product promotions, and inventory management strategies.

2.3. Software context

The HUSTPC system will operate within the context of the internet and will be accessible to users through web browsers. It will interact with external systems such as payment gateways, shipping providers, and inventory management systems. The website will also incorporate user-specific functionalities, including user registration, login, and personalized account management

Future development plans will be based on the features (if any) that do not make it in the initial release of the application. If all of these features are included, there are several experimental features that will potentially be incorporated. These features are not covered in this document.

3. System Architecture

SYSTEM ARCHITECTURE REPORT

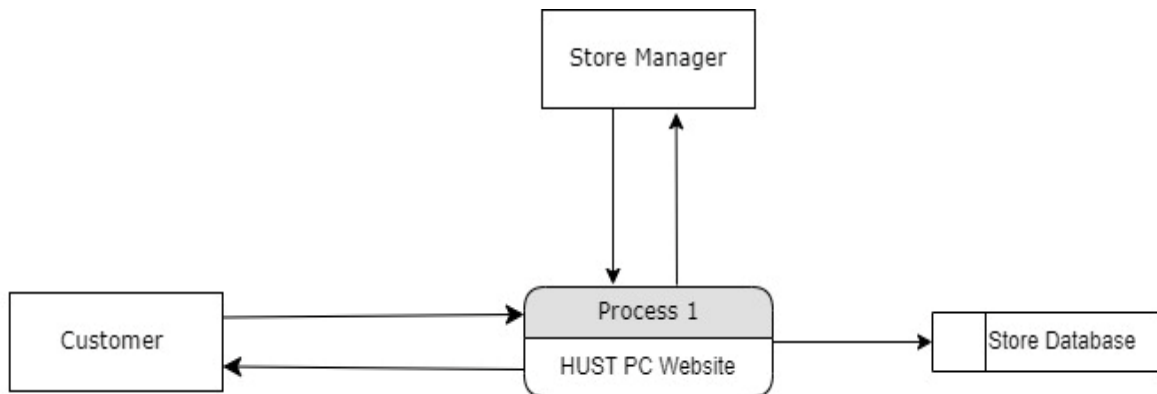## 3.1. Architectural Design

Our system follows a modular program structure, where responsibilities are partitioned and assigned to subsystems to achieve complete functionality. The high-level subsystems identified in our architectural design are as following diagram:
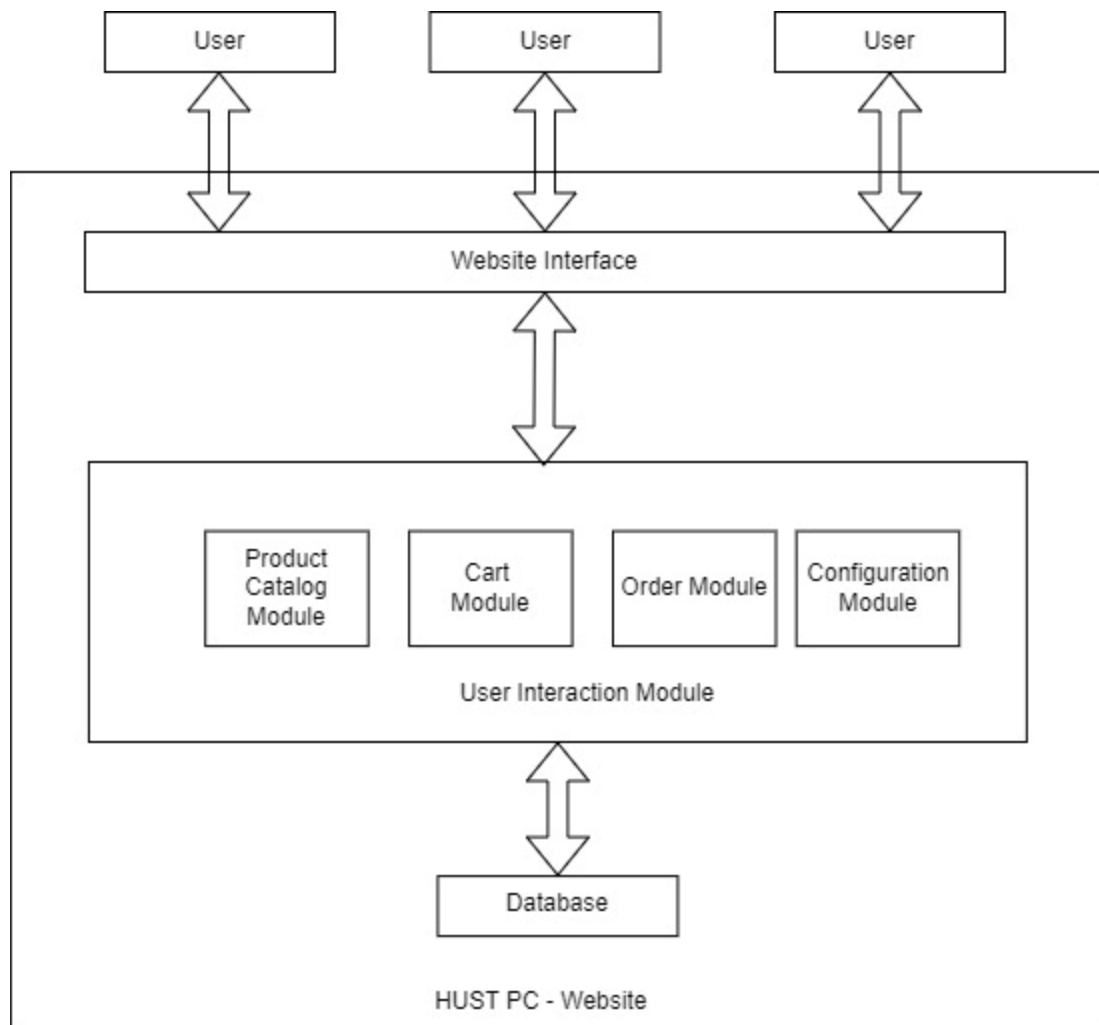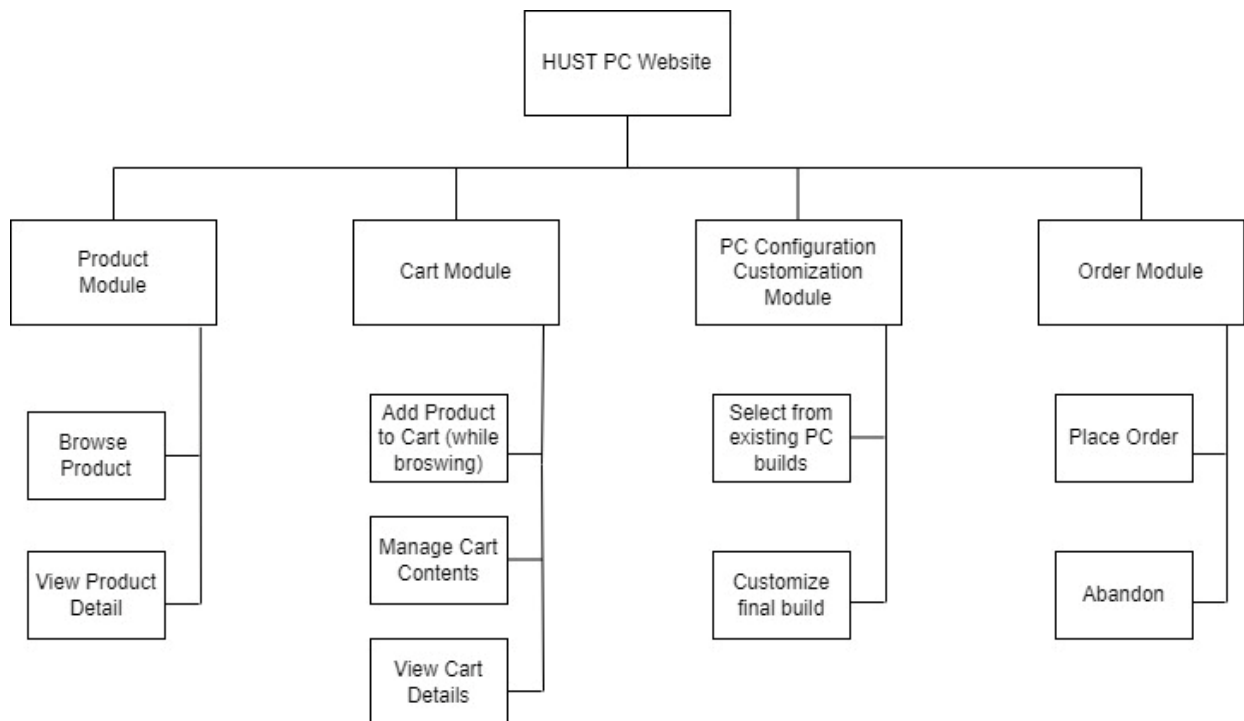
3.2. Decomposition Description

Level 0 data flow diagram is as follows:

More detailed functional decomposition diagram is as follows:

3.3. Design Rationale

The selected architecture was chosen based on several factors and considerations:

Modularity: By decomposing the system into subsystems, namely "Product Module", "Cart Module", "PC Configuration Module", "Order Module", we achieve a modular structure that allows for better organization and maintenance of code. Each module focuses on a specific set of functions, enhancing code reusability and facilitating future enhancements or modifications.

Scalability: The modular design enables scalability, as each subsystem can be independently scaled to accommodate increased user demand or functionality expansions. This provides flexibility in managing system growth.

Separation of Concerns: The decomposition allows for separation of concerns, ensuring that each module has a clear set of responsibilities. This promotes code maintainability, readability, and reduces complexity.

Collaboration and Interoperability: The subsystems collaborate by exchanging data and information to achieve the desired functionality. The modular structure facilitates the seamless integration and communication between subsystems, enabling the system to work cohesively.

Trade-offs: While this architecture provides flexibility and maintainability, there might be a slight increase in system complexity due to the need for inter-module communication. However, the benefits outweigh this trade-off, as it allows for a scalable and robust system.

Other architectures were considered, including a monolithic architecture where all functionalities are contained within a single module. However, this approach lacked modularity and scalability, making it challenging to maintain and adapt to future requirements. The chosen modular architecture provided a more flexible and scalable solution, aligning with our system's goals.

Overall, the selected architecture offers a well-organized, scalable, and maintainable solution that meets the requirements of our system.

Please note that the actual implementation of the modules may vary and require further design and development to meet specific functional and non-functional requirements.

Note: The decomposition and diagrams mentioned in sections 2 and 3 should be provided based on the specific requirements and complexity of your system. The given sections serve as placeholders and should be expanded with the relevant information for your system architecture.

4.   DATA DESIGN

1. Transforming Information Domain into Data Structure

In this section, we will describe how the information domain of the system is transformed into data structures, as well as how the major data or system entities are stored, processed, and organized. We will also list the databases or data storage items utilized by the website.

The information domain is divided into these main classes: *User; Role, Order, Order_Detail; Cart; CartItems; Product; Feedback.* Each class represents a specific entity within our system.

The attributes of each class are represented in fields or columns in the database table. Section 4.2 will specify their data types, sizes and constraints.

2. Data Storage
The data are stored in a relational database management system (RDBMS) using MySQL. The server is implemented using PHP.

3. Processing and Organization
The following image shows the database diagram of the HUSTPC system, which illustrates the relation among entities of the system.
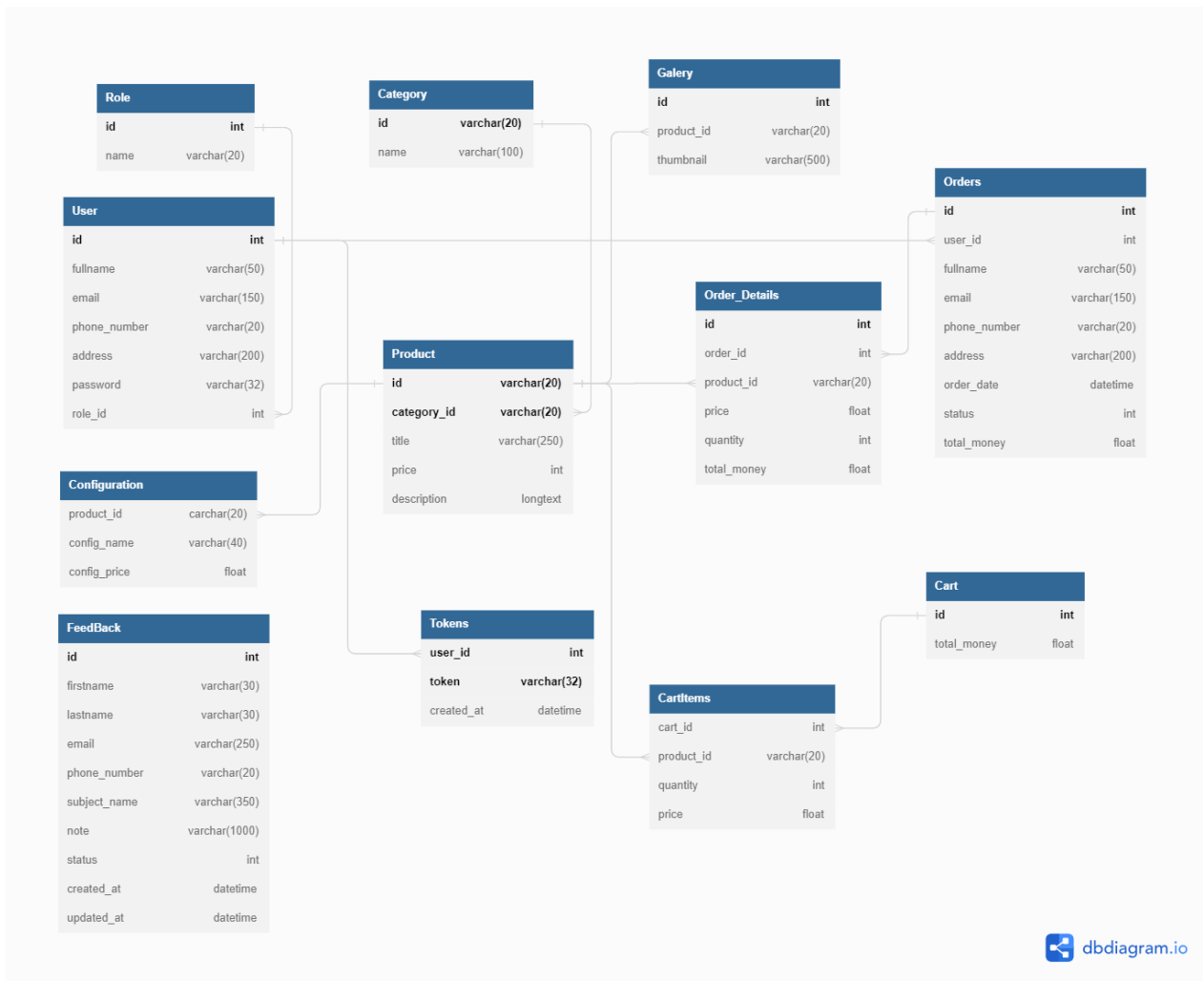
Database Diagram of HUSTPC

4. Data Dictionary

In this section, we will list and specify each table included in the HUSTPC website database diagram, in terms of field name, type, size, constraint, format and provide some additional notes where necessary.

1. **Table "Cart":**

Field Name	Type	Size	Constraints	Format	Description
id	Int		Primary key, Auto-incremented	Positive integer number	This attribute represents the unique identifier for each Cart object

total_money	Float			Positive float number	The total amount of money that need to be checkout

2.   **Table "CartItems":**

3.   **Table "Category":**

Field Name	Type	Size	Constraints	Format	Description
id	Varchar(20)	20 characters	Primary key	Text	This attribute represents the unique identifier for each Category object
name	Varchar(100)	100 characters		Text	The name of each product category

4.   **Table "Configuration"**

Field Name	Type	Size	Constraints	Format	Description
product_id	Varchar(20)	20 characters	Foreign key referencing the "Product" table	Text	The identifier of the referenced product in the table Product.
config_name	Varchar(200)	200 characters		Text	The name of the configuration option
config_price	Float			Positive float number	The price of each configuration option

5.   Table "Feedback"

Field Name	Type	Size	Constraints	Format	Description
id	Int		Primary key, Auto-incremented	Positive Integer Number	This attribute represents the unique identifier for each Feedback object
first_name	varchar(30)	30 characters		Text	The first name of the customer
last_name	Varchar(30)	30 characters		Text	The last name of the customer
email	Varchar (150)	250 characters		Text	The customer 's email address
phone_number	Char(10)	10 characters		Digit sequence	The customer 's phone number
message	Varchar (1000)	1000 characters		Text	The message that the customer input and send to the server
status	Int			0 or 1	Check whether the feedback is accepted or not

Field Name	Type	Size	Constraints	Format	Description
created_at	Datetime			Date-Month-Year	The date at which the feedback is created
updated_at	Datetime			Date-Month-Year	The date at which the feedback is updated

6. Table "Gallery"

Field Name	Type	Size	Constraints	Format	Description
id	Int		Primary key, Auto-incremented	Positive Integer Number	This attribute represents the unique identifier for each displaying object
product_id	Varchar(20)	20 characters		Text	The identifier of the referenced product in the table Product

7. Table "Orders"

Field Name	Type	Size	Constraints	Format	Description
id	Int		Primary key, Auto-incremented	Positive Integer Number	This attribute represents the unique identifier for each Order object

user_id	Int		Foreign key referencing table "User"	Positive Integer Number	The identifier of the referenced user ID in the table User
fullname	Varchar(50)	50 characters		Text	The full name of the customer
email	Varchar(150)	150 characters		Text	The customer's email address
phone_number	Char(10)	10 characters		Digit Sequence	The customer's phone number
address	Varchar(200)	200 characters		Street-Town-District-Province	The address at which the product will be shipped to
order_date	Datetime			Date-Month-Year	The date at which the order is created
status	Int			0 or 1	Check whether the order is accepted or not
total_money	Float			Positive Float Number	The total money of the order that need to be checkout

8.   Table "Order_Details"

Field Name	Type	Size	Constraints	Format	Description
order_id	Int		Foreign key referencing table "Orders"	Positive Integer Number	The identifier of the referenced order in the table Order
product_id	Varchar(20)	20 characters	Foreign key referencing table "Product"	Text	The identifier of the referenced product in the table Product
price	Float			Positive Float Number	The price of the product in the order
quantity	Int			Positive Integer Number	The quantity of product selected in the order
total_money	Float			Positive Float Number	The total amount of money that need to be checkout

9.   Table "Role"

| Field Name	Type	Size	Constraints	Format	Description
id	Int		Primary key, Auto-incremented	Positive Integer Number	This attribute represents the unique identifier for each Role object

name	Varchar(20)	20 characters		Text	Values for the role name: "Customer", "Employee","Manager"

10. Table "User"

| Field Name	Type	Size	Constraints	Format	Description
id	Int		Primary key, Auto-incremented	Positive Integer Number	This attribute represents the unique identifier for each Cart object
fullname	Varchar(50)	50 characters		Text	The full name of the customer
email	Varchar(150)	150 characters		Text	The user 's email address
phone_number	Char(10)	10 characters		Digit Sequence	The user 's phone number
address	Varchar(200)	200 characters		Street-Town-District-Province	The address of the user
role_id	Int		Foreign key referencing table "Role"		The identifier of the referenced role in the table Role

5. COMPONENT DESIGN
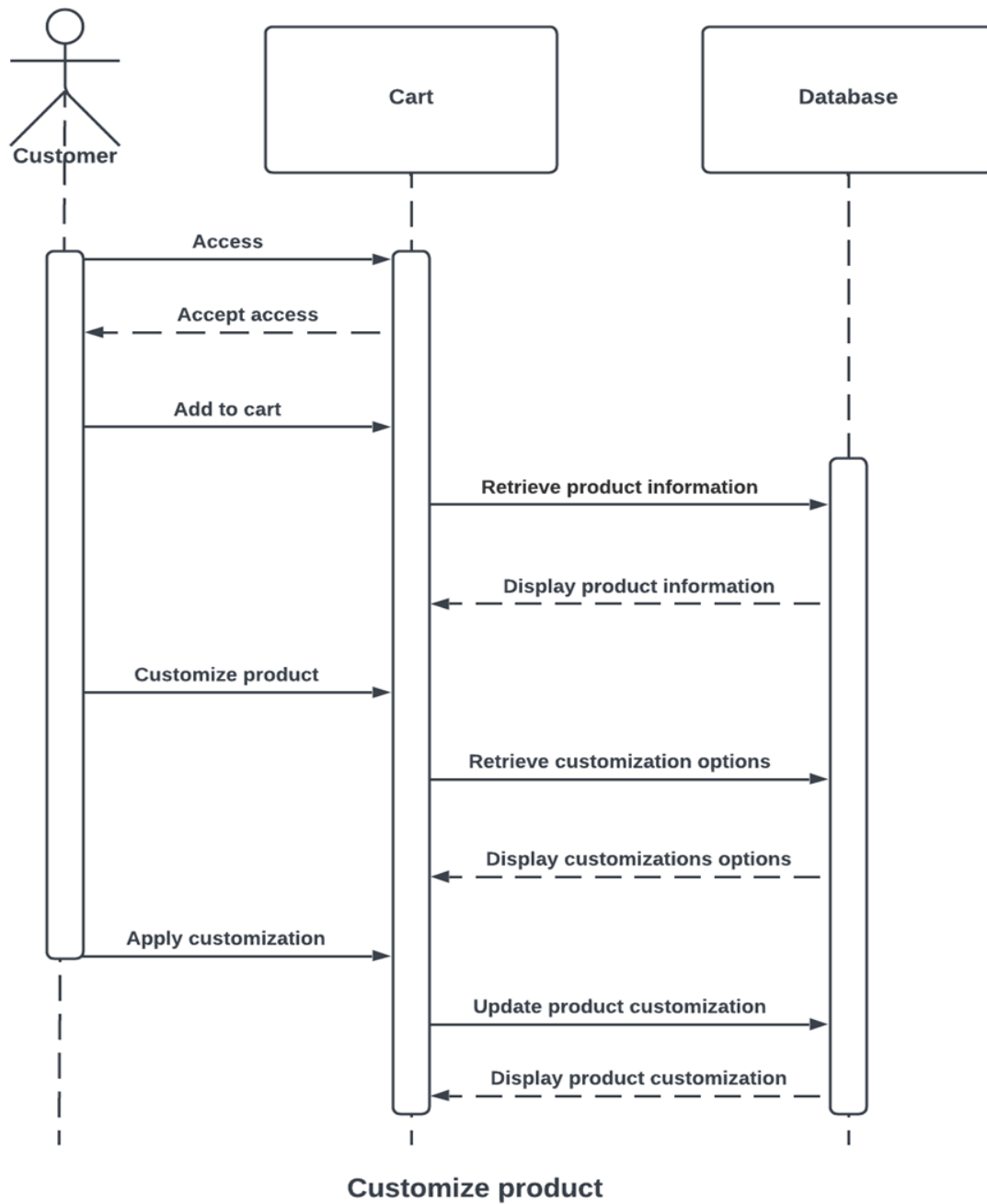
5.1. Browse the web Component
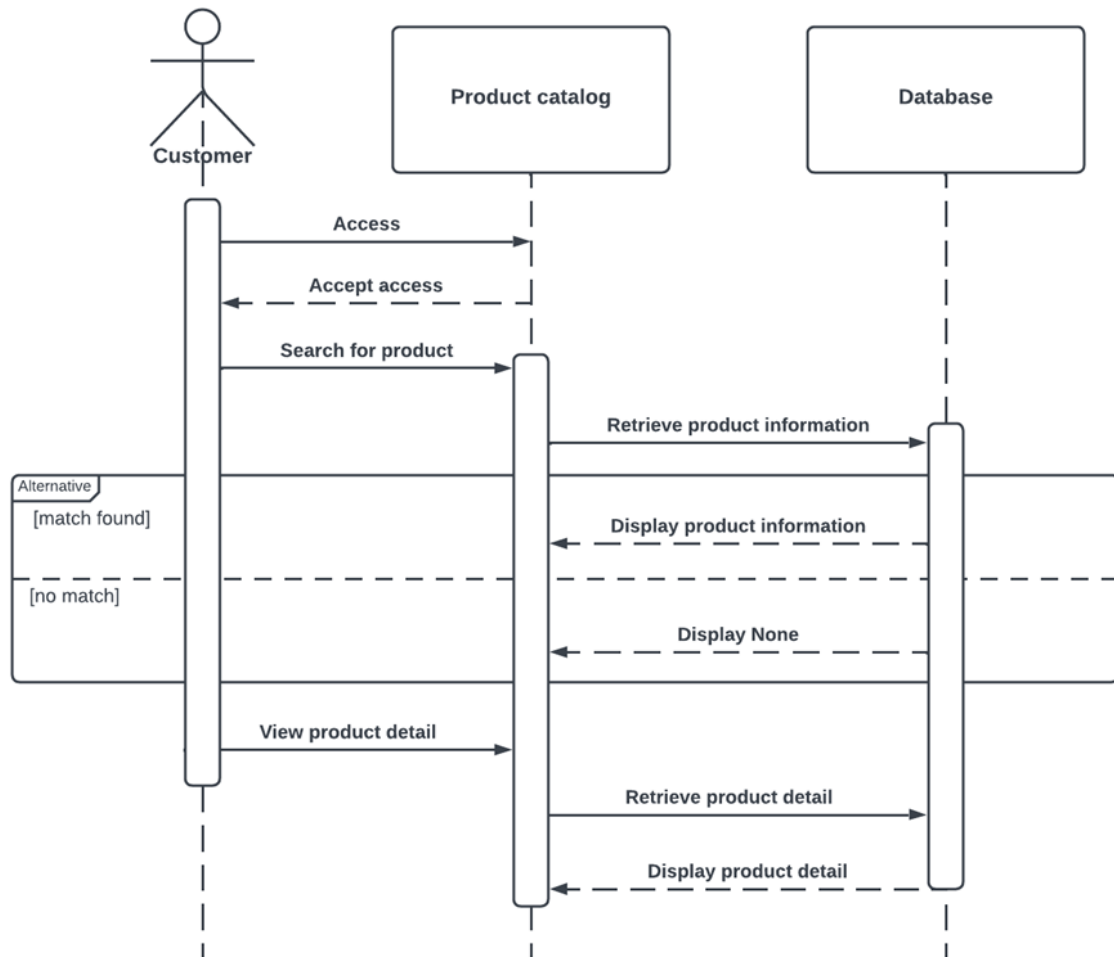
Browse the web
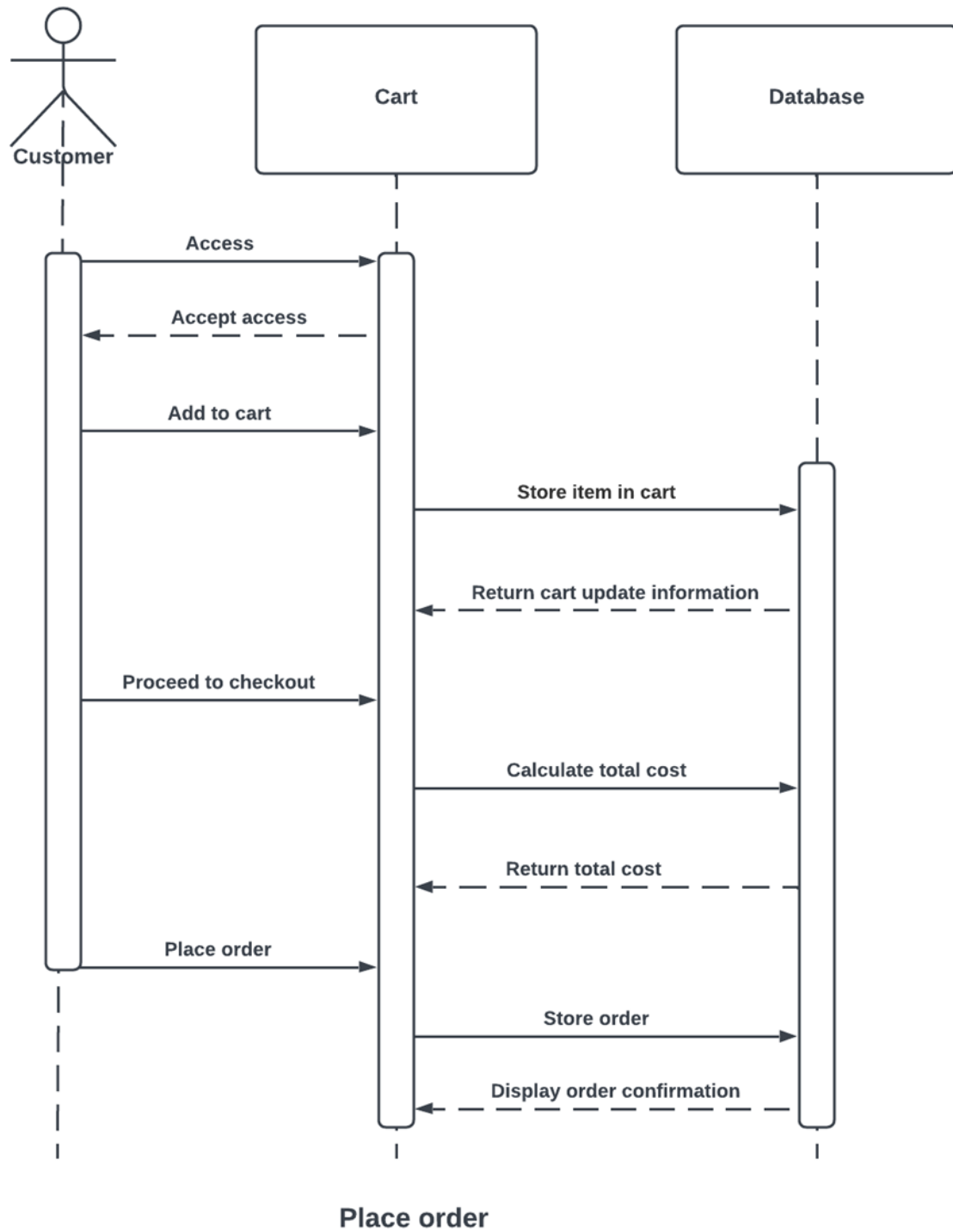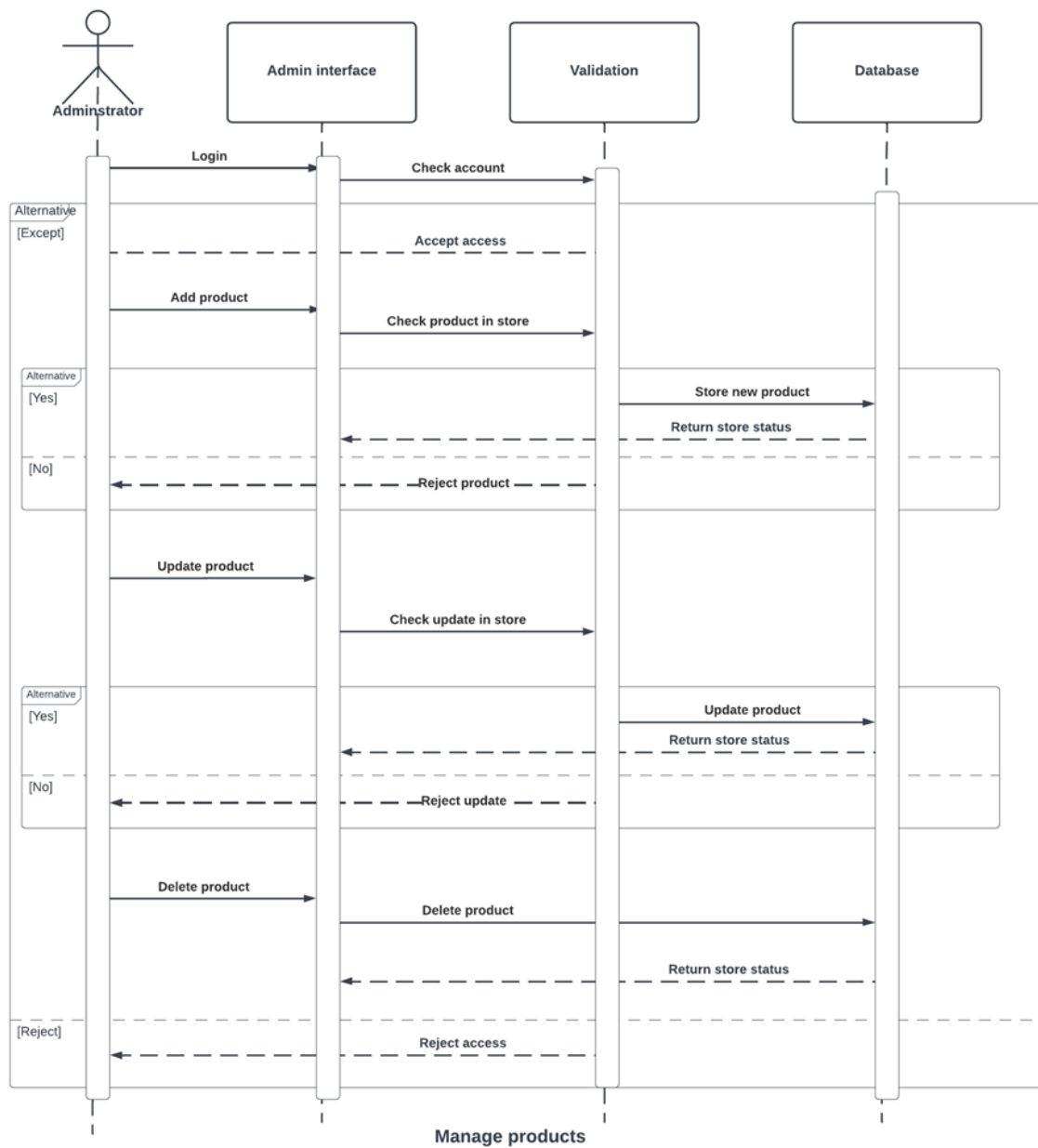
5.2. Build PC Component:

Build PC
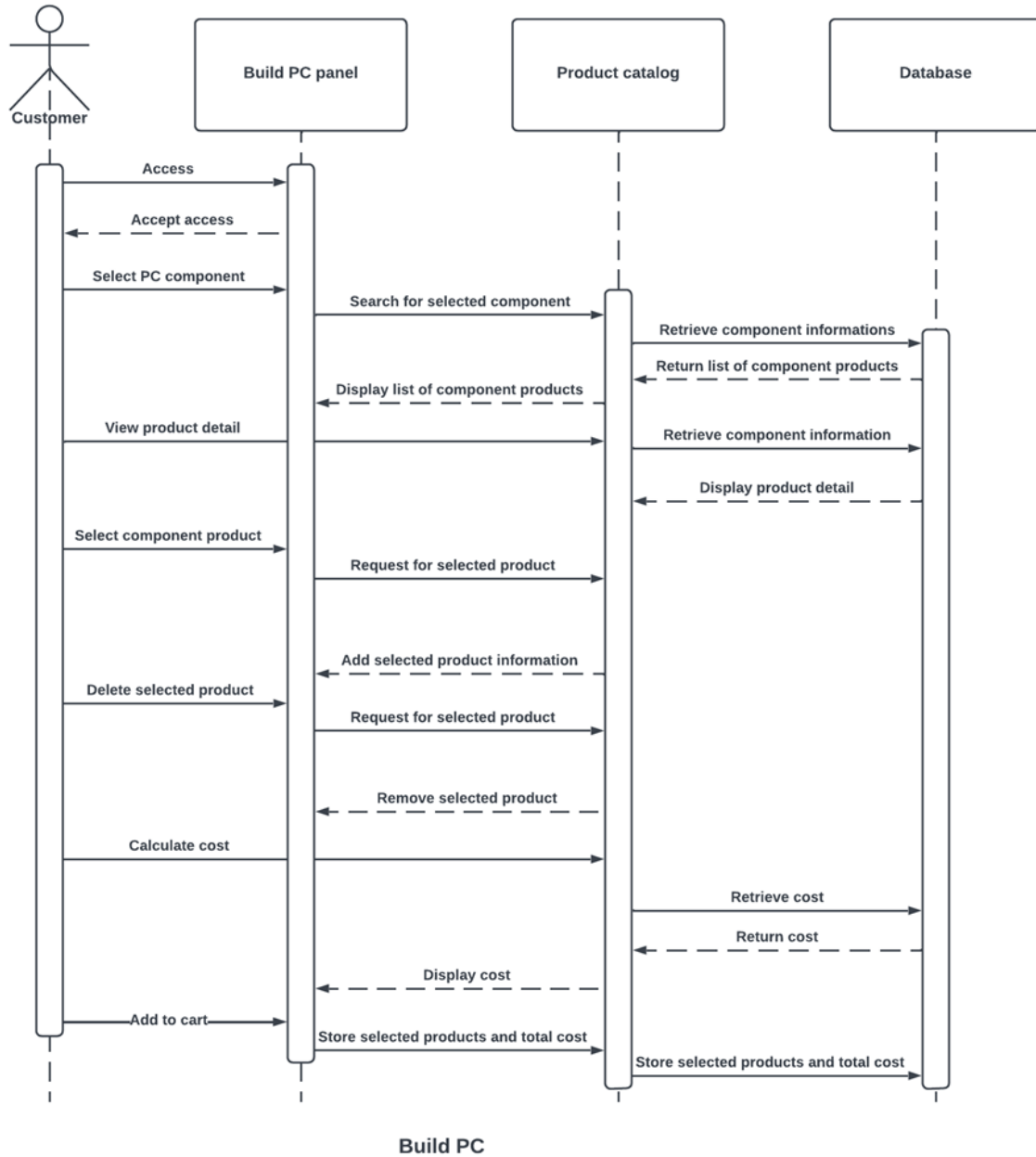
5.3. Customize Products Component:

Customize product

4. Place Order Component:

Place order

5. Manage Products Component:

Manage products

6. Manage Orders Component:

Staff / Administrator

Staff interface

Validation

Database

Login

Check account

Alternative

[Except]

Accept access

View orders

Retrieves oders information

Return orders information

Accept orders

Update orders

Return orders status

Reject orders

Update orders

Return orders status

Generate order report

Retrieve data for order report

Return orders status

[Reject]

Reject access

Manage orders

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

Homepage and Navigation:

Upon visiting the Helarica website, customers are greeted with a visually appealing homepage showcasing featured products, promotions, and categories.

The navigation menu allows users to browse different product categories, such as RAM, ROM, Screen, Accessories,...

Users can choose to navigate to the "Build PC" page and can choose every single part of PCs in order to create a PC as they want.

Users can also search for specific products using a search bar prominently displayed on the homepage.

Product Listings and Filtering:

When users select a category or perform a search, they are presented with a list of products that match their selection.

When users click on a part name needed for PCs in the Build PC page, there will be a list of products which users can choose those parts as they want.

The products are displayed with relevant details such as product name, price, and an image.

Users can further refine their product search by applying filters such as price range, brand, configuration, etc., to narrow down the options.

Product Details and Build PC page:

Clicking on a specific product from the listing, or in the homepage showcasing takes the user to a dedicated product page.

The product page provides comprehensive information about the item, including detailed descriptions, specifications, available variants (configurations, prices, number of products, colors,...)

The Build PC page provides a list of names of parts which are needed for a PC, users can add products by clicking on the select button next to it and choose number, color, configurations as users need, they also can choose "clear all" if they like this.

Shopping Cart and Checkout:

Users can add desired products to their shopping cart by clicking on an "Add to Cart" button present on each product page, product listing or Build PC page. The shopping cart page displays a summary of the items added, their quantities, and the total cost.

Users can modify the quantity of items, remove products, or apply discount codes if applicable.

To proceed with the purchase, users can click on the "Process to check out" button, which leads them to the Checkout page where users are provided a form which requires them to add information about them and billing/shipping addresses.

Payment and Order Confirmation:

Helarica offers various payment options, such as credit/debit card, PayPal, or other online payment gateways.

After completing the payment process, users receive an order confirmation page displaying the order details, estimated delivery date, and a confirmation number.

Users may also receive an email confirmation with the same information for their records.

6.2   Screen Images

Display screenshots showing the interface from the user's perspective. These can be hand drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

6.3   Screen Objects and Actions
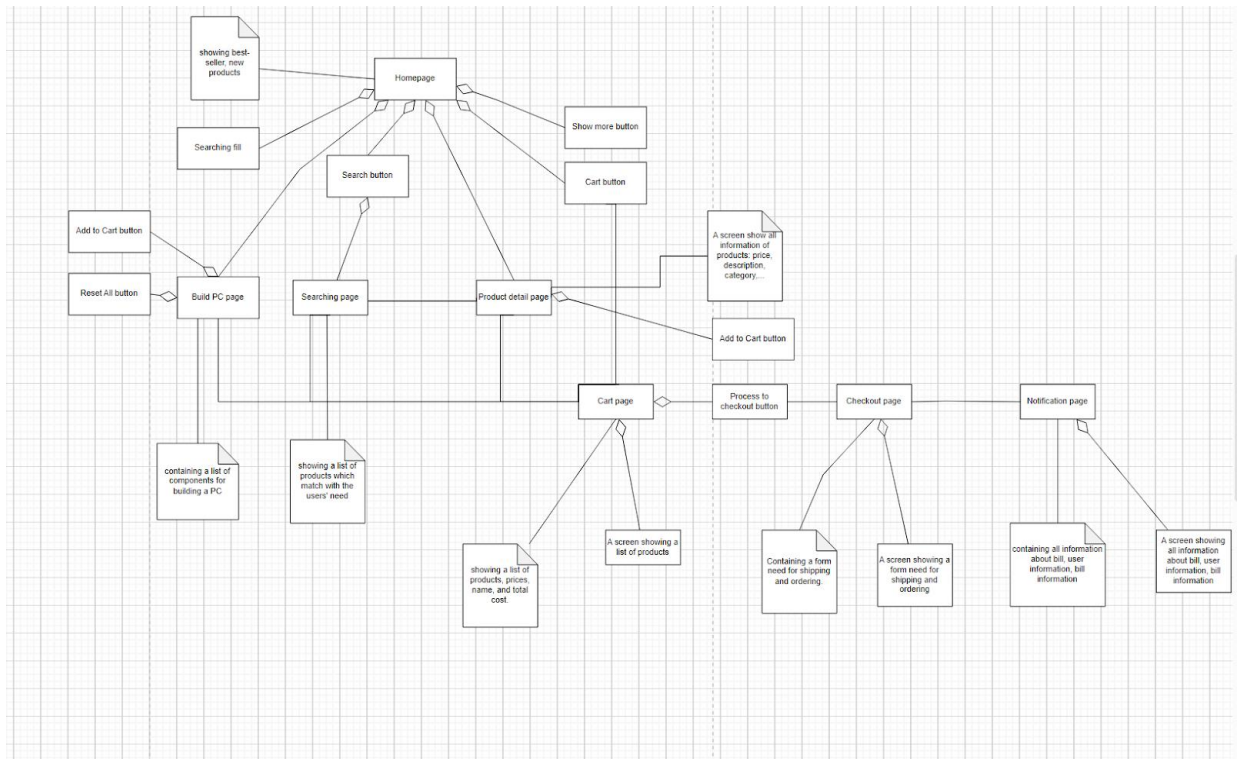
"Home" page:

The home page has menu with submenus on the top of the page:

- All categories

  - CPU

  - RAM

  - CPU Cooler

  - Motherboard

  - Video Card

  - Case

  - Power Supplies

  - Storages

  - Operating System

  - Accessories

- Home

- Products

- Best Sellers

- Blogs

- About Us

- Contact Us

User clicks on the necessary item in the menu in order to perform the operation. For "all categories", "Best sellers", "Products" users could find specific types of products, or some demand to add to cart. "About Us", "Contact Us" give more info about the brand, and Blogs give information about tech.

In addition user can perform next actions:

See all available best seller products, new products or hot discount using scroll bar, click on one of those products could navigate to the detail products;

Watch more products by clicking on what shows more buttons.

Perform basic search for current page putting string in appropriate text box and clicking button "Search" and can choose categories to narrow down the suitable options;

Back to the heading, or see cart, quick search for products by flow button always in the left of screen.

"Product detail" page:

Submenu contains parts as the home page having the same function and operation (see above).

In addition user can perform next actions:

See all details, name, images and prices of a product, and choose the number, configuration of the product and put it in a cart by clicking on the "Add to Cart" button.

"Build PC" page:

Submenu contains parts as the home page having the same function and operation (see above).

In addition user can perform next actions:

Choose all components which are available on the store to build a PC as users want.

Could use the "Reset All" button to remove all parts of the PC or "Add to Cart" button when they want to order it.

"Cart" page:

Submenu contains parts as the home page having the same function and operation (see above).

In addition user can perform next actions:

Watch all products including: name, image, categories, number of products, prices, and total prices and choose "Process to Checkout" to navigate to the "checkout" page.

"Checkout" page:

Submenu contains parts as the home page having the same function and operation (see above).

In addition user can perform next actions:

Fill in all the information for shipping and ordering parts.

Look at the detail bills in the left of form

"Notification" page:

Submenu contains parts as the home page having the same function and operation (see above).

In addition user can perform next actions:

Could see all information about detailed bills, personal information, information about shipping and ordering.

"Contact us" page:

Submenu contains parts as the home page having the same function and operation (see above).

In addition user can perform next actions:

Scroll down to see brand information and address and relevant information.

A scheme presented, in figure below, shows the main web pages (pink rectangles) and main actions (blue rectangles) that can be performed on each page.

7. REQUIREMENTS MATRIX

The following table will present all the system components relating to their functional requirements that are mentioned in the Software Requirement Specification document, along with the "Status" column which shows the progress of each requirement.

Requirement ID	Requirement Description	Priority	Dependency	Status
REQ-001	The website shall provide a product catalog with search and filter functionality.	High	-	Complete
REQ-002	The website shall provide a shopping cart feature for users to add and manage products.	High	-	Complete
REQ-003	The website shall allow users to place orders and provide shipping information. | High	High	-	Complete
REQ-004	The website shall send email notifications to users upon successful order placement	Medium	REQ-003	Draft
REQ-005	The website shall provide a user-friendly interface for browsing and selecting products.	High	-	Complete
REQ-006	The website shall display product details, including price, description, and reviews.	High	-	Complete
REQ-007	The website shall provide a secure login and authentication mechanism for users.	HIgh	-	Draft

REQ-008	The website shall generate order confirmation and invoice for each successful transaction.	Medium	REQ-003	Draft
REQ-009	The website shall allow users to track the status of their orders. |	Medium	REQ-003	In-progress
REQ-010	The website shall provide customer support through a contact form and live chat	Medium	-	Draft
REQ-011	The website shall support multiple languages for international users.	Low	-	Draft
REQ-012	The website shall provide a feedback/rating system for users to rate and review products.	Low	REQ-006	Draft

8. APPENDICES

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.