SOFTWARE DESIGN TITLE

"Design of a Clothing Brand Web System for Online Retail Operations"

Luminario, Venice Lou Gabrielle M. - Leader
Calica, Ljay L.
Gabuyo, Ivan Love D.
Inverzo, Kyle Andrei D.
Aquino, Jester J.
Tan, Charles Dominic S.
Elpedes, Glen Jorge A.

University of Caloocan City
Caloocan City

12/06/2025

Approval Sheet

This design project entitled **"**Design of a Clothing Brand Web System for Online Retail Operations**"** prepared by **Luminario, Venice Lou Gabrielle M., Calica, Ljay L., Tan, Charles Dominic S., Gabuyo, Ivan Love D., Inverzo, Kyle D., Elpedes, Glen Jorge A., Aquino, Jester J.** of the Computer Engineering Department, was examined and evaluated by the members of the Student Design Evaluation Panel and is hereby recommended for approval.

Engr. Maria Rizette H. Sayo
Adviser

Student Design Evaluation Panel:

ENGR/DR. FIRSTNAME LASTNAME
Panel Member

ENGR/DR. FIRSTNAME LASTNAME
Panel Member

ENGR/DR. FIRSTNAME LASTNAME
Lead Panel

ENGR. MARIA CONCEPCION MIRABUENO
Program Chair

UNIVERSITY OF CALOOCAN CITY
Caloocan City

SOFTWARE DESIGN PROJECT INFORMATION

2nd Semester, SY 2025-2026

Student/Team **Team Glenn10**	Luminario, Venice Lou Gabrielle M. - Leader Calica, Ljay L. Gabuyo, Ivan Love D. Inverzo, Kyle Andrei D. Aquino, Jester J. Tan, Charles Dominic S. Elpedes, Glen Jorge A.
Project Title	Web Based System for Online Retail Operations of Doing Good
Project Concentration Area	Web Systems, E-Commerce Technologies, Database-Driven Applications
Design Objectives	The general objective is to provide a comprehensive design and development plan for the clothing brand web system. This includes the features and system functions that will be implemented in the project. Specifically, it aims to: 1. Develop a Web Based system that; • That showcase the clothing (Hoodies, Polo, Pant etc..) of Doing Good • Shows the categories of clothe based gender • Has a database management for products, users, orders, and inventory 2. Test and evaluate the system's accuracy and performance
Constraints	
Performance Constraints (Response Time)	The web system must respond to user interactions (e.g., page load, search queries) within 3 seconds.
Security Constraints (Data Protection Level)	The system must comply with data protection regulations such as GDPR or CCPA, ensuring secure storage and processing of user and payment

	data.
Scalability Constraints (Number of Concurrent Users)	The system should be able to support up to 1,000 concurrent users without performance degradation.
Budget Constraints (Development and Maintenance Costs)	Total development costs should not exceed ₱3,000 and annual maintenance should stay below ₱5,000. The money available for publication will range from ₱500 to ₱3,000.
Technology Constraints (Technology Stack)	The system must be developed using specific technologies, such as React for the front end and Node.js for the back end.
Other constraints: These constraints do not affect each design; therefore, these were not included in selecting the best design.	
Constraint	A constraint is simply a rule or limit that controls what you can or cannot do.
Standards	
Standard A - WCAG 2.1 Level AA	This standard makes sure websites are accessible to people with disabilities. In the project, it was used to improve text contrast, add text to product images, and ensure the site can be used with a keyboard.
Standard B - OWASP Top 10	This standard lists the most common security risks for web systems. In the project, it helped guide secure login, protect user data, and prevent issues like unsafe input or weak passwords.

Abstract

The design and development of a web system for a clothing brand that will facilitate online retail operations is presented in this project. The system's main goals are to create an intuitive user interface for administrators and customers alike, efficiently arrange product catalogs, and create a seamless inventory and order management workflow. A database system to manage products, users, orders, and stock levels, as well as front-end modules for user interaction and back-end modules for data processing, are all part of the project. Performance testing and functional assessments were carried out to confirm accuracy, responsiveness, and dependability in order to guarantee system quality. To improve user experience and safeguard sensitive data, accessibility and security standards, such as WCAG 2.1 Level AA and the OWASP Top 10, were implemented. The final design provides a scalable, safe, and effective web-based retail solution that is suited to a contemporary clothing brand's requirements.

Keywords: Web System Design, Online Retail, Clothing Brand, User Interface, Product Catalog Management, Inventory and Order Management, Front-End Development, Back-End Development, Database Management, System Performance, Accessibility Standards, WCAG 2.1 Level AA, Security Standards, OWASP Top 10, Scalable Web Solution.

List of Tables

List of figures

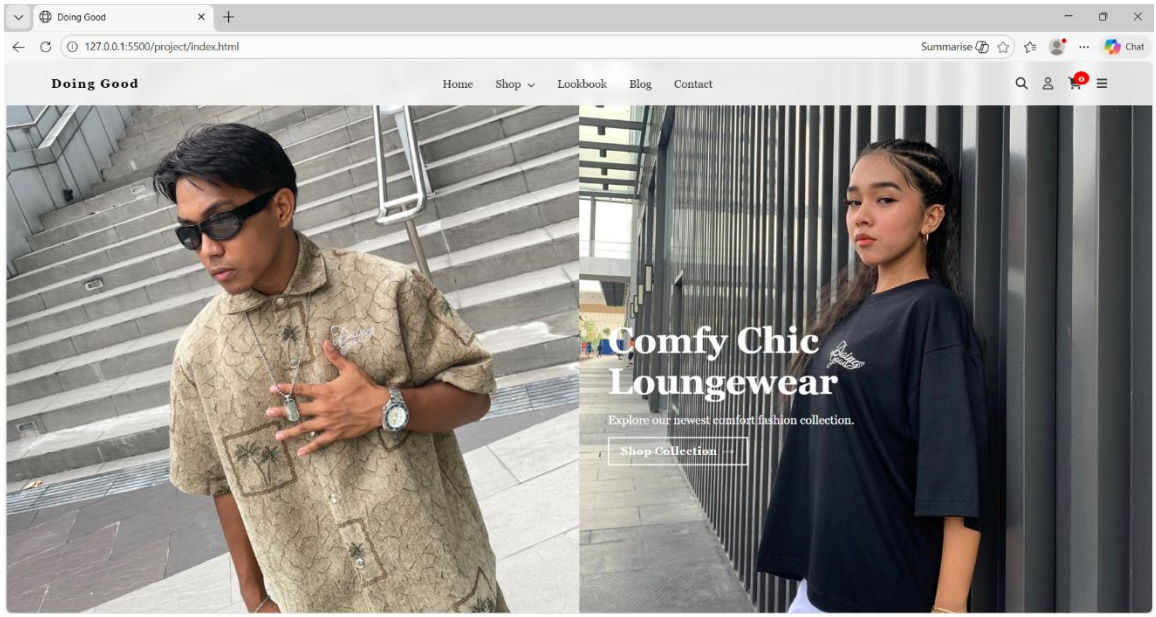

Figure 1.1 The Engineering Design Process (Teach Engineering, 2025)

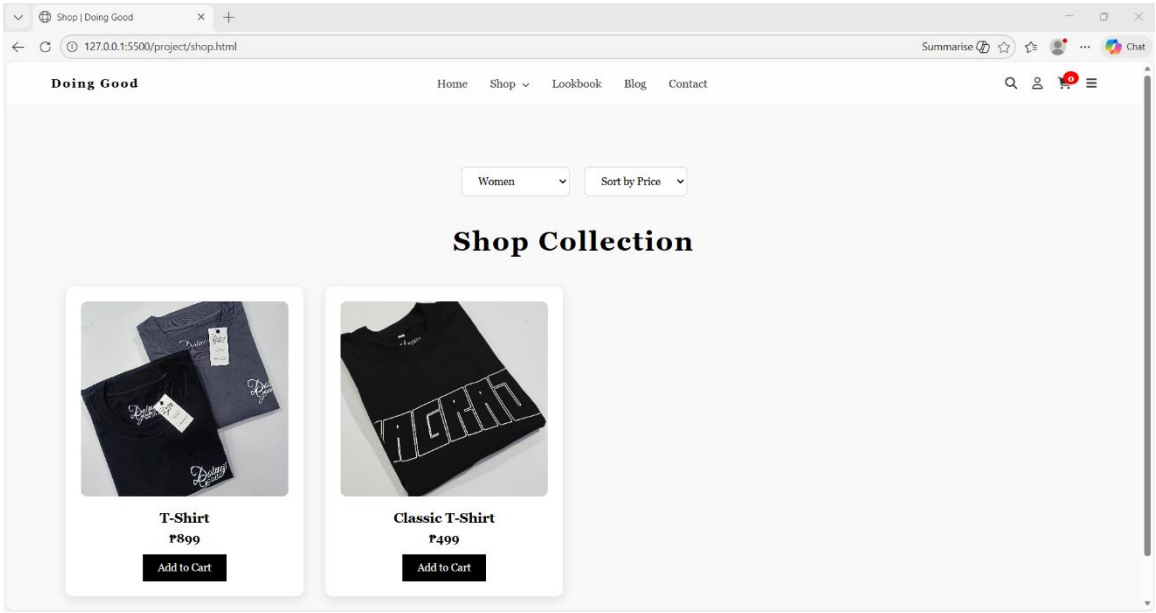

Figure 1.2 The Engineering Design Process (Teach Engineering, 2025)

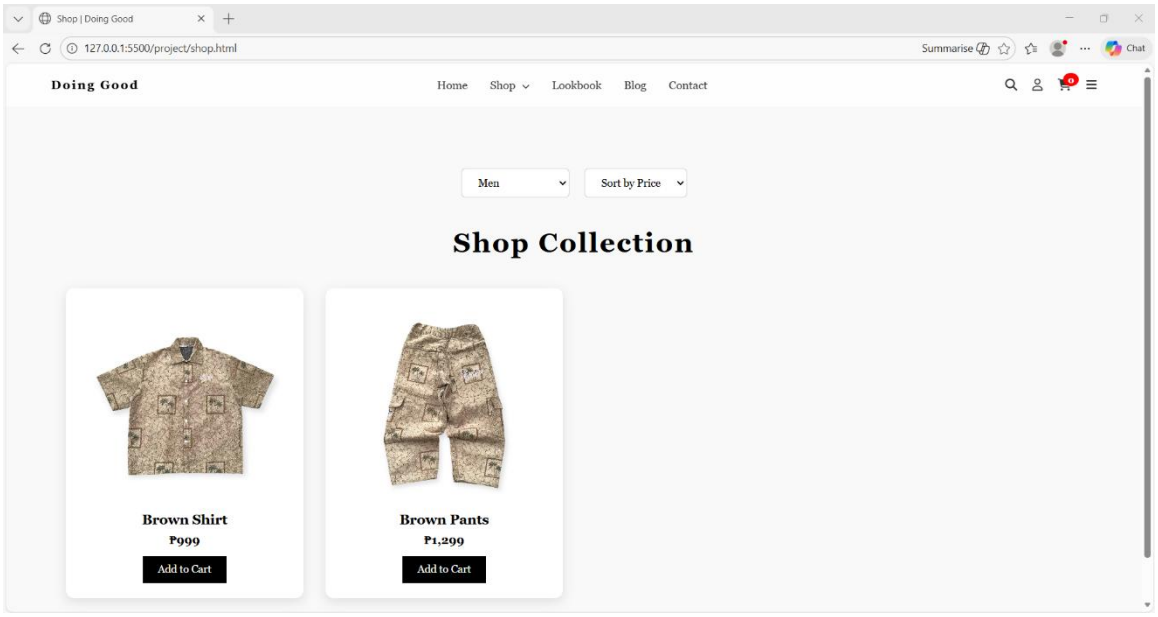

Figure 1.3 The Engineering Design Process (Teach Engineering, 2025)

List of abbreviation

Definition of terms

TABLE OF CONTENTS

CHAPTER 1: THE PROJECT AND ITS BACKGROUND

Online shopping is now vital for clothing brands, but small and medium firms often struggle with poor order management, manual inventory tracking, and messy product catalogs. These problems cause delays, wrong stock info, unhappy customers, and security risks. The project's goal is to build a strong web system that fixes these issues and helps apparel businesses run smoothly online.

1.1 The Problem

Resources to explore:

- Define the Problem in Engineering Design https://www.sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-problem-statement

- How to Write an Effective Problem Statement Example https://libguides.wmich.edu/iee1020/problem

- How do you frame a problem statement in engineering? https://static.nsta.org/ecybermission-files/helpdocs/ED%20Defining%20the%20Problem.pdf

1.2 The Client

The client for this project is **Mr. Dexter Luis Vega**, the owner of an established clothing brand that is currently operating without its own online retail system. To support the growth and digital expansion of his business, our team will be developing a web-based platform that aligns with the brand's operational needs and customer expectations.

Through the interview and data-gathering process, our team will identify the specific requirements of Mr. Vega's clothing brand. The planned system will provide structured product displays, real-time inventory management, secure customer transactions, and an organized order processing workflow. By creating this platform, the project will contribute to enhancing the brand's online presence, accessibility, and overall retail efficiency.

Table 1-1. Client and Engineering Requirements / Considerations

Client Requirements / Considerations	Engineering Requirements / Considerations
The system can allow customers to browse and view clothing products easily.	Design a responsive, user-friendly UI using React; follow WCAG 2.1 Level AA for accessibility.
The system can securely store user accounts and handle online transactions safely.	Implement secure authentication, encrypted data storage, and OWASP Top 10 security practices.

Client Requirements / Considerations	Engineering Requirements / Considerations
The system can track inventory and update stock levels automatically when orders are placed.	Create a database with automated inventory management logic using backend APIs in Node.js.

1.3 The Project

This project focuses on creating a web system for **Mr. Dexter Luis Vega's clothing brand** to help the business operate online. The system will allow customers to browse products and make purchases, while administrators can manage products, inventory, and orders efficiently. The system will have a **front-end** for users to interact with, a **back end** to process orders and update data, and a **database** to store product, user, and sales information. It will also follow **accessibility** and **security standards** to ensure it is safe and easy to use. By developing this system, our team will help the brand **expand online, improve operations, and prepare for future growth**.

1.4 Project Objectives

The primary objective of this project is to design and develop a comprehensive web system to facilitate the online retail operations of Mr. Dexter Luis Vega's clothing brand. The system will integrate functionalities for efficient management of product catalogs, inventory control, order processing, and user interaction.

Specifically, the project aims to:

1. Develop the software modules encompassing both front-end and back-end functionalities, including the following:

- dynamic product display
- secure transaction handling
- database management
- real-time inventory synchronization

2. Test and evaluate

- the system's accuracy
- performance
- security
- compliance with relevant standards to guarantee robustness
- reliability
- data integrity

1.5 Scope and Delimitations

This project focuses on designing and developing a web system for Mr. Dexter Luis Vega's clothing brand to support online sales and business operations. The system will include features for managing product listings, tracking inventory, processing customer orders, and handling user accounts for both customers and administrators. The system will follow important standards to ensure it is accessible and secure.
The web system will be built using React for the user interface and Node.js for the server side, along with a database to store product and customer data. The system will be tested in controlled environments to check its accuracy and performance under normal use. This project covers only the web application and does not include developing a mobile app or connecting with external delivery or payment services beyond basic online transactions. It will support up to 1,000 users at the same time but will not include advanced performance tuning for larger traffic. Security will protect against common threats, but highly advanced cyberattacks are beyond the project's scope. Marketing, sourcing of products, and physical inventory handling outside the system are not part of this project. The user interface will focus on core functions without extensive design customization or branding details.

1.6 Design Constraints

Constraint A - Performance (Response Time)

- Performance refers to how fast the system responds to user actions such as loading pages, searching products, or submitting orders.

- The metric used is **average response time**, which measures how long the system takes to react. It is obtained by timing system operations during testing and calculating the average duration.

- A lower response time means the system performs better and provides a smoother user experience.
  Therefore, the design with the lowest response time is the best design.

Constraint B - Security (Data Protection Level)

- Security refers to the system's ability to protect user information, especially customer accounts and transactions.

- The metric used is the **number of security vulnerabilities found**, such as weak passwords or unsafe inputs.

- It is measured by checking the system against OWASP Top 10 risks during testing.
  A lower number of vulnerabilities means stronger protection and safer system use.
  Therefore, the design with the fewest security vulnerabilities is the preferred design.

Constraint C - Scalability (Concurrent Users)

- Scalability refers to the system's ability to support many users using the website at the same time without slowing down. The metric used is **maximum supported concurrent users** before performance drops.

- It is measured by load-testing the system under increasing user levels until response time exceeds 3 seconds.

- A higher supported user count means better scalability and system strength.
  Therefore, the design that supports the most concurrent users is the winning design.

Constraint D - Usability (User Interface Accessibility)

- Usability refers to how easy the system is to use for customers and administrators. The metric used is the **usability score**, based on WCAG 2.1 Level AA accessibility guidelines.

- It is measured through evaluation checklists, user testing, and accessibility tools.
  A higher usability score means the system is more accessible and easier to navigate.
  Therefore, the design with the highest usability score is the best choice.

Constraint E - Cost (Development and Maintenance Budget)

- Cost refers to the financial limits available for building and maintaining the web system. The metric used is the **total project cost**, which includes development tools, hosting, and yearly maintenance.

- It is computed by adding all expenses and comparing them with the assumed limits of **₱3,000 for development** and **₱5,000 for annual maintenance** based on our research.

- A lower total cost—while still meeting system requirements—is more desirable.
  Therefore, the design with the lowest cost that stays within budget is the most suitable design.

Public Health

Welfare

Social

Global

Cultural

1.7 Engineering Standards

The engineering standards used in this project ensure that the clothing brand web system is accessible, secure, reliable, and aligned with accepted practices for modern web applications. These standards support the system's design goals and help maintain quality throughout development.

WCAG 2.1 Level AA (Web Accessibility Standard)

This standard ensures that the website is accessible to all users, including those with disabilities. It supports the project's goal of creating an easy-to-use interface by guiding text contrast, keyboard navigation, readable layouts, and alt-text for product images. This directly strengthens the system's usability requirement.

OWASP Top 10 (Web Security Standard)

This standard lists the most common web security risks. It connects to the project's constraint on data protection by helping identify and prevent issues such as weak authentication, unsafe input handling, and data exposure. This ensures that user accounts, transactions, and stored data remain secure.

ISO/IEC 25010 (Software Quality Standard)

This standard guides how software quality is evaluated. It supports the project's testing requirements by defining expectations for performance, reliability, usability, and security. This connects to your objective of ensuring that "all functions operate correctly" and the system responds efficiently.

ISO/IEC 27001 (Information Security Standard)

This standard provides principles for protecting sensitive information. It supports the project's need for secure data storage and processing, especially for customer information, orders, and administrator accounts. It directly connects to your security constraints and database requirements.

ISO/IEC 14543-3 (System Architecture Standard)

Although originally intended for electronic systems, this standard is used here to support a clear and organized system architecture. It helps guide how system components communicate and work together, which connects to your objective of designing the overall structure of the system (front-end, back-end, and database).

1.8 Engineering Design Process

The image shows two people wearing stylish and comfortable clothes. Just like how clothing is planned and created, the Engineering Design Process also follows steps to make a good final product. It starts with finding a problem, thinking of ideas, testing them, and improving the design. This process helps make sure the result is useful, creative, and fits what people need.

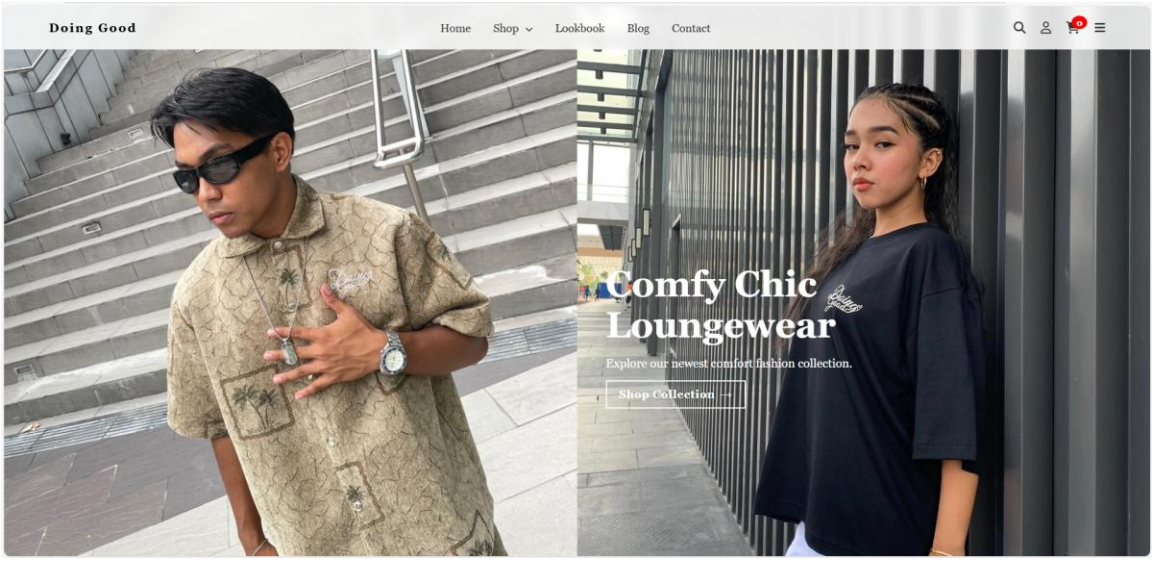

Figure 1.1 The Engineering Design Process (Teach Engineering, 2025)

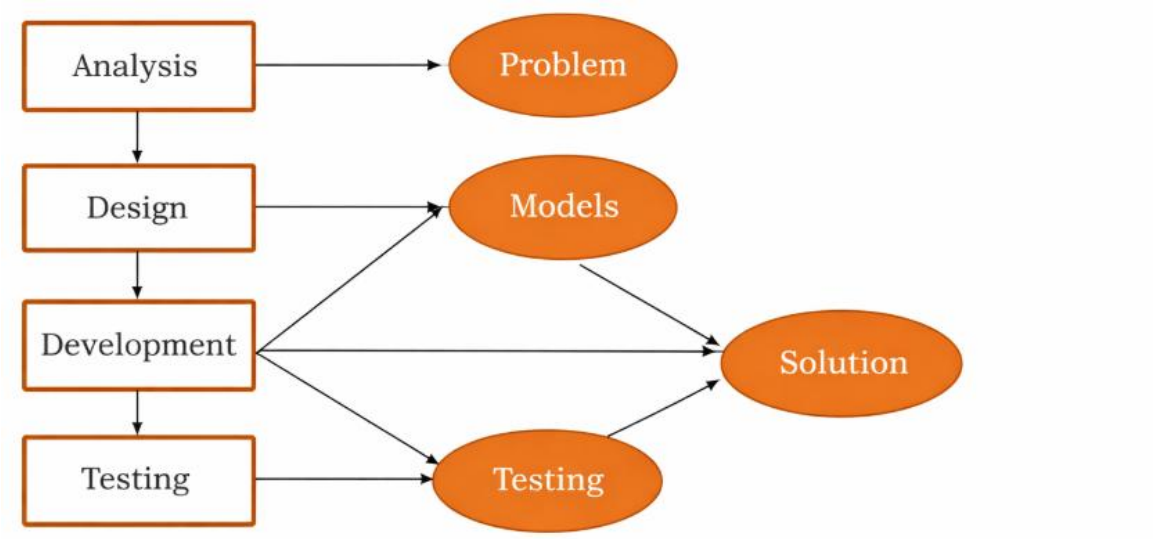

Figure 1.2 The Engineering Design Process (TeachEngineering, 2025)

The picture depicts two individuals dressed comfortably and stylishly. The Engineering Design Process follows steps to produce a high-quality final product, just like clothing is planned and made. These steps are clearly depicted in the diagram. Analysis is the first step, where we comprehend the issue or requirement. After that, we proceed to the Design stage, where concepts and models are developed. These concepts are then transformed into actual products during the development stage. Lastly, testing verifies that everything functions properly. These procedures aid in enhancing the design and guarantee that the finished product is practical, imaginative, and meets people's needs.

1.8.1 Ask: Identifying the Need and Constraints

This step entails figuring out what the client needs and comprehending the main issue. A fully functional clothing brand web system that facilitates online retail operations is required for this project. Constraints like performance limitations, security needs, financial limitations, and accessibility requirements are identified by the team. By posing crucial queries, such as "What should the system do?" Who is going to use it? What limits must it follow? —the group creates a strong basis for the design.

1.8.2 Research the Problem

The team collects pertinent data and examines current systems after determining the need. This entails examining industry standards like WCAG 2.1 and OWASP Top 10, evaluating existing e-commerce platforms, and conducting client interviews to gain a deeper understanding of operational requirements. Technologies like React, Node.js, and database structures are also covered in research. This stage guarantees that the project is based on well-informed choices rather than conjecture.

1.8.3 Imagine: Develop Possible Solution

The group imagines and brainstorms different solutions in this step. This could include different database designs, inventory tracking strategies, and user interface layouts for the web system. Here, the team is encouraged to be creative and can freely explore ideas without worrying about constraints right away. To determine the most promising system structure, a number of design options are drafted.

1.8.4 Plan: Select a Promising Solution

1.8.5 Create: Build a Prototype

A working model or preliminary version of the web system is created during this stage. Simple versions of the front-end and back-end interactions, the basic UI framework, or the initial database setup are examples of prototypes. The objective is to develop a working prototype that shows how the finished system will function, not to achieve perfection. Early on in the development process, this prototype aids in identifying possible problems.

1.8.6 Test and Evaluate the Prototype

Once the prototype is created, it undergoes performance, usability, and security testing. The team examines whether the system meets the client's requirements and satisfies constraints such as response time, accessibility, and data protection. User feedback and testing tools help uncover errors, weaknesses, and areas for improvement. Evaluation ensures that the design functions reliably before moving to full development.

1.8.7 Improve: Redesign as Needed

Based on test results, the team revises and enhances the design. Improvements may involve fixing bugs, optimizing performance, strengthening security features, or reorganizing the interface for better user experience. This step reflects the iterative nature of engineering: the design is refined repeatedly until it meets the required standards and delivers a high-quality final pro

CHAPTER 2: SOFTWARE DESIGN

This chapter presents the software design of the proposed Clothing Brand Web System for Online Retail Operations. It outlines the design solution, system architecture, data management approach, applied engineering principles, and the design alternatives evaluated prior to selecting the final implementation. This chapter aims to clearly describe the system structure and demonstrate how the design meets the project objectives and constraints defined in Chapter 1.

2.1 Description of the Design Solution

2.1.1 General Description

The proposed design solution is a web-based e-commerce system that allows customers to browse products, create accounts, place orders, and track purchases, while enabling administrators to securely manage products, inventory, and customer orders through a centralized dashboard.

The system follows a three-tier architecture:

1. **Presentation Layer (Front-end)** – built using React, responsible for user interaction and interface rendering.

2. **Application Layer (Back-end)** – developed using Node.js and Express, responsible for business logic, authentication, order processing, and inventory updates.

3. **Data Layer (Database)** – stores product information, user accounts, orders, and inventory records.

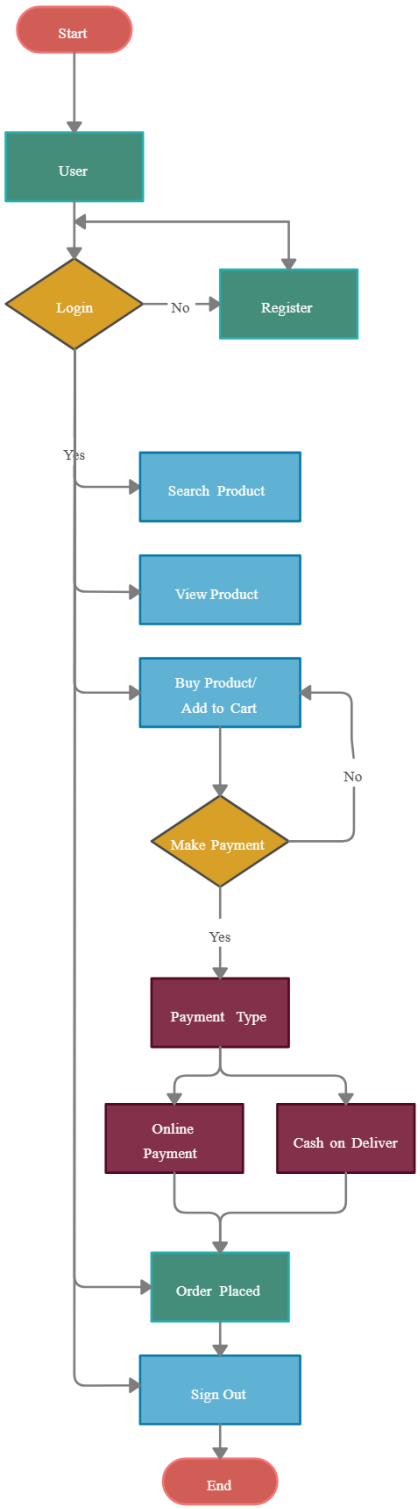

Figure 2.1 Illustrative Diagram of the System

Users access the system through a web browser, where requests are handled by the React-based front end and forwarded to the Node.js API server. The server processes each request, retrieves or updates data from the database, and returns the appropriate response to the front end for display.j Administrators follow the same process but are granted access to additional system management modules.

This architecture promotes separation of concerns, simplifies system maintenance, supports scalability, and enhances overall security.

2.1.2 Engineering Principles Involved

The system design applies several fundamental software and systems engineering principles to ensure reliability, maintainability, and scalability:

1. Modularity

- The system is structured into independent modules, including authentication, product management, order processing, and inventory control. This modular approach simplifies debugging, testing, and future enhancements.

2. Abstraction

- Complex operations such as payment processing, database interactions, and security validation are encapsulated within application programming interfaces (APIs) and service layers, allowing developers to work with simplified and consistent interfaces.

3. Separation of Concerns

- The system architecture clearly separates responsibilities among the front end (presentation), back end (business logic), and database (data storage). This improves maintainability and reduces overall system complexity.

4. Scalability

- The client–server architecture enables the system to accommodate increasing user demand by supporting server upgrades and the integration of load-balancing mechanisms when necessary.

5. Reliability and Fault Tolerance

- Input validation, structured error handling, and logging mechanisms are implemented to ensure stable operation and facilitate rapid identification and resolution of system issues.

6. Security by Design

- Security considerations are incorporated from the initial design phase, including encrypted

password storage, role-based access control, and secured API endpoints to protect sensitive data and system resources.

2.1.3 Prior Art Analysis

Several existing e-commerce systems influenced the design of this project, particularly the official Nike website, which serves as a benchmark for modern online retail platforms. The Nike website demonstrates effective product presentation, streamlined navigation, secure user account management, and efficient order processing workflows.

Narrative comparison:

The Nike website provides a highly responsive and visually structured product catalog that enhances user experience and supports quick product discovery. It also features advanced filtering options, real-time inventory visibility, and an intuitive checkout process that minimizes user effort.

In addition, the platform integrates secure authentication mechanisms and order tracking functionalities, allowing customers to monitor their purchases seamlessly. However, such enterprise-level systems typically require significant development resources and infrastructure, making them less practical for small to medium-sized businesses.

In contrast, the proposed system adopts similar functional concepts and usability principles while maintaining a simplified architecture and lower implementation cost, making it more suitable for the client's operational scale and budget constraints.

Matrix Format is preferred, in addition to narratives. Explain the table.

Table 2.1 Prior Art Analysis Matrix

Design	Features				
	Product Catalog Organization	**Advance Search & Filtering**	**User Account Management**	**Order Tracking**	**Brand Customization**
Nike Website	X	X	X	X	X
Generic E-commerce Template	X		X	X	Limited
Marketplace Platform	X	X	X	X	
Proposed System	X	X	X	X	X

2.2 General System Architecture

This section explains how the architectural components are realized and integrated within the system, without revisiting their prior definitions.

2.2.1 Hardware Elements

The proposed system does not require specialized or dedicated hardware components. It is a purely software-based web application that operates using existing client devices and standard server infrastructure. Users access the system through commonly available devices such as desktop computers, laptops, tablets, or smartphones with web browsers. Server-side operations are handled using standard web hosting environments without the need for custom hardware configurations.

2.2.2 Software Elements

A. Embedded Software
No embedded software is required for this project, as the system is purely web-based.

B. Application Software
The system is implemented as a web application using the following technologies:

- Frontend: React, HTML, CSS, JavaScript
- Backend: Node.js, Express.js
- Database: Relational database (e.g., MySQL or PostgreSQL)
- Security Libraries: JWT for authentication, bcrypt for password hashing

Both customer and administrator user interfaces are browser-based and require no additional software installation.

C. Key Algorithms Used

- User authentication and authorization algorithms
- Inventory synchronization logic
- Order validation and processing algorithms

2.2.2 System Algorithm

The system follows a structured request–response algorithm:

1. User submits a request through the frontend interface.
2. The frontend validates input and sends the request to the backend API.
3. The backend authenticates the user and processes business logic.
4. The database is queried or updated as required.
5. The backend sends a response to the frontend.
6. The frontend displays the result to the user.

2.2.3 Data, Datasets, and Processing

A. Datasets

The system uses internally generated datasets including user accounts, product details, order records, and inventory data. Data is collected through user registration forms, administrative inputs, and transaction records.

B. Data Processing Scheme

Raw data from user inputs is validated and sanitized before storage. Inventory data is automatically updated after each confirmed order. Processed data is stored in structured database tables and retrieved for reporting and display purposes.

C. Other Data Utilized

Mock data is used during system testing to simulate product listings, user activity, and order transactions.

2.3 Design Alternatives

The various design options taken into consideration for the Clothing Brand Web System for Online Retail Operations are shown in this section. Each option offers a workable way to put the system into place while meeting the project's goals and limitations in terms of cost, performance, security, scalability, and usability. To make sure that the final design choice is technically sound, economical, and suitable for the client's business's operating size, it is essential to evaluate several options.

2.3.1 Rationale for Design Alternatives

The chosen design options concentrate on several web system architectures that are frequently employed in e-commerce platforms. These options were selected because they have a direct impact on system performance, security, scalability, development complexity, and maintenance costs—all of which are crucial limitations mentioned in Chapter 1.

The design team was able to assess how various structural approaches affect system behavior under real-world circumstances like concurrent user access, secure transaction handling, and budget constraints by contrasting a monolithic architecture, a microservices-based architecture, and a serverless/cloud-based architecture. For a small to medium-sized clothing firm, these options are important because they direct the decision-making process toward a design that strikes a balance between technical capacity and practical viability.

2.3.1 Design Alternative 1:

A. **Engineering Principles of Alternative**

Tight component integration, centralized system architecture, and process simplicity are the tenets of this design option. A single backend program is used to implement all system functions, including order processing, inventory control, product administration, and user authentication. Internal communication between components minimizes deployment overhead and architectural complexity.

This method is appropriate for small-scale systems with tight budgets because it places an emphasis on simplicity of development, simple debugging, and low infrastructure requirements.
Note: This repeats for all 3 designs. It must not mention the principles mentioned above but instead those specific to the design alternative.

B. Architecture of Design Alternative

The system is set up as a single web application that includes:

- A React-based front end for communicating with administrators and customers
- All business logic is handled by an Express and Node.js backend.
- Users, goods, orders, and inventory information are all stored in a single database.

The same server program handles all client requests; it talks with the database directly and provides answers to the front end.

C. Constraints

- **Constraint A (Performance):**
  Efficient for low to moderate traffic, but response time may degrade as user load increases.

- **Constraint B (Security):**
  Security controls are centralized, simplifying enforcement but increasing risk if the application is compromised.

- **Constraint C (Scalability):**
  Limited scalability due to the single-application structure.

- **Constraint D (Usability):**
  Supports consistent UI behavior and accessibility implementation.

- **Constraint E (Cost):**
  Lowest development and maintenance cost among the alternatives.

iv. Evaluation Results (if model)

2.3.2 Design Alternative B

i. Engineering Principles of Alternative

This alternative is based on **distributed systems**, **service independence**, and **loose coupling**. Each major function of the system—authentication, product catalog, inventory management, and order processing—is implemented as a separate service with its own logic and interfaces.

This approach improves scalability and fault isolation by allowing services to operate and scale independently.

ii. Architecture of Design Alternative

The system is divided into multiple backend services, each exposed through RESTful APIs:

- A front-end React application communicates with multiple backend services
- Each service runs on Node.js and handles a specific function
- Services interact with a shared or service-specific database layer

This architecture allows individual services to be updated or scaled without affecting the entire system.

iii. Evaluation of Constraints

- **Constraint A (Performance):** Improved performance under heavy load but with added network communication overhead.
- **Constraint B (Security):** Better isolation of sensitive services, though increased complexity in securing inter-service communication.
- **Constraint C (Scalability):** High scalability due to independent service deployment.

2.3.3 Design Alternative C

i. Engineering Principles of Alternative

This alternative applies **event-driven computing**, **cloud resource abstraction**, and **automatic scalability**. Backend logic is implemented using cloud-based serverless functions that execute in response to user requests, eliminating the need for dedicated server management.
The system relies heavily on managed cloud services for computing, authentication, and database storage.

ii. Architecture of Design Alternative

- A React front end communicates with cloud-hosted APIs

- Backend logic is executed through serverless functions
- A managed cloud database stores system data
- The cloud provider automatically handles scaling, availability, and infrastructure maintenance

iii. Constraints

- **Constraint A (Performance): Good performance after initialization, but may experience cold-start delays.**
- **Constraint B (Security): Strong built-in security features from cloud providers.**
- **Constraint C (Scalability): Excellent scalability, automatically adjusting to user demand.**

2.4 Standards Involved in the Design

This section presents the standards followed by the design, including their references. Matrix may be used to show how standards are used in each specific design.

Table 2.4 Summary of Standards Involved in the Alternatives

Standard	Brief Description	DESIGNS		
		DESIGN A	DESIGN B	DESIGN C
WCAG 2.1 Level AA (Web Accessibility Standard)	Web accessibility standard ensuring usability for users with disabilities.	Used to ensure readable text, proper contrast, and alt-text for product images.	Used to support keyboard navigation and accessible checkout process.	N/A
OWASP Top 10 (Web Security Standard)	Web security standard identifying common web application vulnerabilities.	Used to secure user authentication and form inputs.	Used to protect payment and transaction processes.	Used to prevent data exposure and injection attacks.
ISO/IEC 25010 (Software Quality Standard)	Software quality standard for evaluating system performance and reliability.	Used to verify functional correctness of core features.	Used to assess system performance and responsiveness.	Used to evaluate usability and security during testing.
ISO/IEC 27001 (Information Security Standard)	Information security management standard.	Used to secure customer accounts and personal data.	Used to protect order and payment records.	Used to control administrator access and database security.
ISO/IEC 14543-3 (System Architecture Standard)	Standard supporting structured system architecture and component interaction. system components.	Used as reference for front-end and back-end integration.	Used to guide communication between system modules.	N/A

Explain of the Table

The table summarizes the standards considered in the design of the online clothing brand system and illustrates how each standard is applied across different design alternatives. WCAG 2.1 Level AA is used to enhance accessibility and usability for all users. OWASP Top 10 focuses on identifying and mitigating common security threats to protect user data and transactions. ISO/IEC 25010 ensures the system meets quality expectations such as reliability, performance, and usability. ISO/IEC 27001 is applied to safeguard sensitive information, while ISO/IEC 14543-3 supports the organization and interaction of system components.

Summary

In summary, applying these standards ensures that the online clothing brand system is accessible, secure, reliable, and well structured. The standards guide the design choices across different alternatives, helping the system meet user requirements, security constraints, and overall project objectives.

CHAPTER 3: DESIGN TRADEOFFS

3.1 Summary of Constraints

Explain table xx below in this paragraph.

Table xx Summary of Design Constraints

Designs	Constraints				
	Constraint A (Metric)	Constraint B (Metric)	Constraint C (Metric)	Constraint D (Metric)	Constraint E (Metric)
Design A					
Design B					
Design C					

Synthesize for the next section.

3.2 Trade-offs

Table xx Preference and Importance of Constraints

Constraints	Preference	Importance (raw)	% Importance

Explain the use of Pareto Multi-Criteria Decision Making (MCDM).

$$Minimization = 9 \times \left(\frac{Max\ Value - Raw\ Value}{Max\ Value - Min\ Value}\right) + 1$$ Equation No. xx

$$Maximization = 9 \times \left(\frac{Raw\ Value - Min\ Value}{Max\ Value - Min\ Value}\right) + 1$$ Equation No. xx

3.2.1 Tradeoff 1: Constraint A (Metric)

3.2.1.1 Design 1: Normalization of Constraint A (Metric)
<Introduce>

Table xx Evaluation of Three Design Alternatives based on Constraint A

Design	Constraint (Metric)

<Analyze>

3.2.1.2 Design 2: Normalization of Constraint A (Metric)

Table xx Evaluation of Three Design Alternatives based on Constraint B

Design	Constraint (Metric)

3.2.1.3 Design 3: Normalization of Constraint A (Metric)

Table xx Evaluation of Three Design Alternatives based on Constraint C

Design	Constraint (Metric)

3.2.2 Tradeoff 2: Constraint B (Metric)

3.2.2.1 Design 1: Normalization of Constraint B (Metric)

3.2.2.2 Design 2: Normalization of Constraint B (Metric)
3.2.2.3 Design 3: Normalization of Constraint B (Metric)

3.2.3 Tradeoff 3: Constraint C (Metric)

3.2.3.1 Design 1: Normalization of Constraint C (Metric)
3.2.3.2 Design 2: Normalization of Constraint C (Metric)
3.2.3.3 Design 3: Normalization of Constraint C (Metric)

3.2.4 Tradeoff 4: Constraint D (Metric)

3.2.4.1 Design 1: Normalization of Constraint D (Metric)
3.2.4.2 Design 2: Normalization of Constraint D (Metric)
3.2.4.3 Design 3: Normalization of Constraint D (Metric)

3.2.5 Tradeoff 5: Constraint E (Metric)

3.2.5.1 Design 1: Normalization of Constraint E (Metric)
3.2.5.2 Design 2: Normalization of Constraint E (Metric)
3.2.5.3 Design 3: Normalization of Constraint E (Metric)

3.3 Summary of the Normalized Values of the Three Designs

Designs	Constraints				
	Constraint A (metric)	Constraint B (metric)	Constraint C (metric)	Constraint D (metric)	Constraint E (metric)
Design A					
Design B					
Design C					

3.4 Designers Raw Ranking for the Three Designs

Table xx Designers Raw Ranking for the Three Designs

Decision Criteria	Criterion's Importance		Ability to Satisfy Criterion		
	Scale (0-10)	Percentage (%)	Design A	Design B	Design C

3.5 Sensitivity Analysis
3.6 Influence of the Design Tradeoffs in the Final Design

CHAPTER 4: FINAL DESIGN

4.1 Final Design
4.1.1 Software Application
4.1.2 Hardware Design
4.2 Test Procedures and Evaluation
4.2.1 Test Procedures
4.2.2 Test Evaluation
4.3 Test and Evaluation Results
4.3.1 Test Results
4.3.2 Evaluation Results
4.4 Conclusion
4.5 Impact of the Design
4.5.1 Societal
Target UN SDG.
4.5.2 Ethical
In compliance with known ethical codes/standards
4.5.3 Legal
National / Intl Laws
4.6 Sustainability Plan

CHAPTER 5: BUSINESS PLAN AND MODEL

5.1 Business Plan
5.1.1 Executive Summary
5.1.2 General Company Description
5.1.3 Products and Services Offered
5.1.4 Marketing Plan
5.1.5 Marketing Strategy
5.2 Business Model
5.3 Intellectual Property (IP) Reports

REFERENCES

Note: This must be done using APA format. Check the guide for more details: https://www.scribbr.com/apa-style/apa-seventh-edition-changes/

Covey, S. R. (2013). *The 7 habits of highly effective people: Powerful lessons in personal change*. Simon & Schuster.

APPENDICES

Include standards preview, certification from experts/clients, code snippets, patent reports, and other long and detailed documents here. Format is as follows below:

APPENDIX A: TITLE OF THE SECTION

<figure>

Note: No figure number. Standards must be followed with a paragraph explaining its contents and purpose.