

## ACKNOWLEDGEMENT

With profound gratitude, I acknowledge the blessings of almighty God, the most gracious and most merciful, who has endowed me with the ability and fortitude to accomplish this dissertation.

I would like to express my deepest appreciation to **Ms. Priti Bala**, Department of Computer Science, Aligarh Muslim University (AMU), for his exceptional mentorship, invaluable feedback, and constant encouragement throughout the duration of this project. His vast knowledge and expertise have been instrumental in shaping the direction of my project work and guiding it to completion.

I am also deeply indebted to **Prof. Arman Rasool Faridi**, Chairperson of the Department of Computer Science, AMU, for his continuous encouragement and leadership. His commitment to fostering a vibrant academic environment has facilitated my growth throughout this project.

My sincere thanks go to the esteemed faculty members of the Department of Computer Science, AMU. Their teachings and guidance have been crucial to my academic development. Their passion for teaching extends beyond the classroom, and their willingness to engage in discussions and share their wealth of knowledge has made my academic journey both challenging and incredibly rewarding.

I extend my heartfelt gratitude to my colleagues at the Department of Computer Science, AMU, for their constant support and inspiration throughout this process.

Finally, I am deeply grateful to my family, whose unconditional love and support have been my backbone. Their constant encouragement has always pushed me to do my best.

**MD ARSLAN**

**GL 7134**

**23CAMSA131**

# ABSTRACT

Artisanal e-commerce businesses that manage a complete production lifecycle, from raw material procurement to direct-to-consumer sales, face a significant operational disconnect due to the lack of integrated digital platforms. This fragmentation forces them into inefficient workflows, hinders data-driven decision-making, and constrains growth. This thesis details the design, development, and evaluation of Vachetta, a novel, vertically integrated e-commerce platform architected to solve this problem.

Built on a modern technology stack including Next.js, React, and MongoDB, the project follows a systematic Iterative Development Model, evolving from a core Minimum Viable Product (MVP) to a feature-rich application. The scientific novelty of Vachetta lies in its unified dashboard that provides a holistic, real-time view of the entire business, seamlessly connecting a Business-to-Business (B2B) procurement module for managing suppliers and raw materials with a Business-to-Consumer (D2C) storefront. Key features include an interactive product customizer that delivers a bespoke luxury experience for customers and an integrated backend that streamlines the complex workflow of producing and selling customized goods.

The resulting platform serves as a successful prototype and proposes a new paradigm for digital craftsmanship. It demonstrates that a purpose-built, integrated system can significantly reduce operational friction, enhance brand value, and provide the necessary tools for niche, vertically integrated artisans to thrive in the competitive digital economy.

## Table of Content

<b>ACKNOWLEDGEMENT.....</b>	<b>1</b>
<b>ABSTRACT.....</b>	<b>2</b>
<b>Table of Content.....</b>	<b>3</b>
<b>Chapter - 1.....</b>	<b>5</b>
<b>INTRODUCTION.....</b>	<b>5</b>
1.1 Introduction.....	6
1.2 Literature Survey.....	6
1.3 Present State of the Art and its Shortcomings.....	8
1.4 Realization of the Problem.....	9
1.5 Introduction of the Problem to be Taken Up.....	9
<b>Chapter - 2.....</b>	<b>11</b>
<b>PROBLEM FORMULATION.....</b>	<b>11</b>
2.1 Problem Definition.....	12
2.2 Various Aspects of the Problem.....	12
2.3 Existing System.....	13
2.4 Scientific Novelty and Need of the Work.....	14
2.5 Proposed System.....	15
<b>Chapter - 3.....</b>	<b>17</b>
<b>REQUIREMENT ANALYSIS AND SPECIFICATION.....</b>	<b>17</b>
3.1 Functional Requirements.....	18
3.2 Hardware Configuration.....	21
3.3 Software Configuration.....	21
3.4 Non-Functional Requirements.....	23
3.5 Feasibility Study.....	24
3.6 Development Methodology: The Iterative Model.....	25
<b>Chapter - 4.....</b>	<b>28</b>
<b>SYSTEM DESIGN.....</b>	<b>28</b>
4.1 Input Design.....	29
4.2 Process Design.....	31

4.3 Database Design.....	32
4.4 Output Design.....	33
<b>Chapter - 5.....</b>	<b>35</b>
<b>CONCLUSION &amp; FUTURE WORK.....</b>	<b>35</b>
5.1 Conclusion.....	36
5.2 Advantages and Special Features of the System.....	37
5.3 Limitations.....	38
5.4 Future Extensions.....	39
<b>REFERENCES.....</b>	<b>41</b>
<b>APPENDIX:.....</b>	<b>42</b>
A1: Entity Relationship Diagram.....	42
A-2: Data Flow Diagram.....	45
A3: UML Use Case Diagram.....	48
A4: Class Diagram.....	51

# **Chapter - 1**

---

## **INTRODUCTION**

---

## 1.1 Introduction

The emergence of computers has ignited a technological transformation, revolutionizing industries and reshaping our world in unprecedented ways. In the realm of e-commerce, computers have become indispensable tools, empowering businesses to reach global audiences, streamline operations, and provide personalized shopping experiences. Online marketplaces, driven by sophisticated algorithms and vast databases, have democratized access to a diverse range of products and services, transforming how we shop and conduct transactions. This digital shift, amplified by cloud solutions and the rise of m-commerce, has fundamentally altered consumer behavior and expectations, ushering in a new era of digital commerce.

While this revolution has enabled mass-market retail on an unprecedented scale, it has also created a counter-movement among consumers who seek value beyond price and convenience. There is a growing demand for products that embody quality, durability, craftsmanship, and a unique story. This dissertation operates at the confluence of these two trends: leveraging the power of modern e-commerce technology to serve the nuanced market for high-quality, artisanal goods. Specifically, this research focuses on the unique challenges and opportunities inherent in selling luxury leather articles online, where tactile qualities and brand narrative are paramount.

The proposed project, **Vachetta**, is a bespoke e-commerce platform designed from the ground up to address this niche. It is engineered not merely as a transactional tool, but as a digital showroom that conveys the story, texture, and unique aging process (patina) that defines high-quality leather. The platform aims to merge a seamless, high-tech user experience with the high-touch sensibility of a traditional luxury brand, creating a trustworthy and engaging environment for discerning customers. This project explores the concept of "digital craftsmanship"—the practice of using technology not to replace but to amplify the value of artisanal work.

## 1.2 Literature Survey

A review of the evolution of e-commerce provides the necessary context for this project. Previous research and development in e-commerce have yielded various platforms and solutions, each with its own strengths and weaknesses that inform the need for the Vachetta platform.

- **Early E-commerce Platforms (1990s):** The emergence of the internet paved the way for early e-commerce platforms, which were rudimentary in functionality and security. These

initial systems, while groundbreaking, were little more than digital catalogs with basic ordering capabilities. They proved the concept of online retail but lacked the sophistication required for complex product representation, secure transactions, or meaningful customer interaction.

- **Rise of Content Management Systems (2000s):** Content management systems (CMS) like WordPress and Drupal gained popularity, enabling businesses to create and manage websites without extensive technical expertise. While these platforms democratized web presence, their e-commerce capabilities were often implemented through third-party plugins (e.g., WooCommerce). This approach frequently resulted in a fragmented system architecture, potential security vulnerabilities from multiple plugins, and a user experience that was not seamlessly integrated. For a luxury brand, this patchwork solution often fails to deliver the required polish and reliability.
- **Evolution of E-commerce Giants and SaaS Platforms (2010s):** E-commerce giants like Amazon and Alibaba revolutionized online shopping with vast product selections, personalized recommendations, and streamlined checkout processes. Concurrently, SaaS (Software as a Service) platforms like Shopify and BigCommerce emerged, offering businesses an all-in-one solution for a monthly fee. These platforms are powerful for standard product sales but reveal significant limitations for businesses focused on bespoke or highly customizable goods. Their rigid data models make it difficult to manage complex product variations, and true customization often requires costly, specialized apps that can slow down the site and complicate management.
- **Custom-built Enterprise Solutions:** In response to the limitations of generic platforms, large enterprises invest in custom-built e-commerce websites tailored to their specific business needs. These sites often integrate with Enterprise Resource Planning (ERP) systems to centralize data on inventory, orders, customers, and operations. The advent of Point of Sale (POS) systems further enables businesses to unify online and offline sales. While this approach offers maximum control, the cost and complexity of development and maintenance are prohibitive for small-scale artisans and independent brands.
- **Integration of AI and Machine Learning (2020s):** The integration of artificial intelligence (AI) and machine learning (ML) into e-commerce platforms introduced features like chatbots, personalized marketing, and dynamic pricing. This has raised consumer expectations for a personalized and intelligent shopping experience, a standard that niche brands must now meet to remain competitive.

This evolutionary trajectory reveals a significant gap for platforms that cater specifically to high-value, craft-based D2C (Direct-to-Consumer) brands. Such brands require not only robust transactional functionality but also sophisticated tools for brand storytelling and product personalization—an area where generic platforms fall short and custom enterprise solutions are out of reach.

### 1.3 Present State of the Art and its Shortcomings

While existing e-commerce platforms have empowered businesses to establish an online presence, several critical limitations hinder their ability to fully capitalize on the digital marketplace, especially for niche luxury goods.

- **Fragmented Management:** Disparate tools for managing inventory, orders, customer data, and content create operational inefficiencies, data silos, and increased overhead. For a business offering custom products, this fragmentation is particularly damaging. An artisan might have to use one app for product options, another for inventory tracking of raw materials, and manually reconcile custom order details, a process that is both time-consuming and prone to significant fulfillment errors.
- **Scalability Constraints:** Many platforms struggle to accommodate growth in product complexity. As an artisan wishes to offer more leather types, thread colors, or hardware options, the backend of a generic platform can become unwieldy, leading to performance bottlenecks, increased costs for additional apps, and missed opportunities to offer a richer customization experience.
- **Subpar User Experience:** Clunky interfaces, convoluted navigation, and limited personalization contribute to customer frustration, high bounce rates, and abandoned carts. In the luxury market, a poor UX directly translates to a poor brand image. Generic themes and checkout processes lack the elegance and sense of security required to convince a customer to make a high-value purchase.
- **Limited Integrations:** Compatibility issues and limited integration capabilities with essential business tools hinder seamless data flow and automation. For example, syncing custom order data with accounting software or a CRM can be a significant manual challenge.
- **Security Vulnerabilities:** Older or poorly maintained platforms, or those reliant on a multitude of third-party plugins, can expose sensitive customer data and financial



transactions to security risks. For a new luxury brand whose primary currency is trust, a single security breach can be irreparably damaging.

- **Inefficient Order Management:** The core logic of most e-commerce platforms is built around standard SKUs (Stock Keeping Units). They are inherently ill-equipped to handle the unique workflow of a made-to-order or customized product, which requires tracking of specific material choices and production stages. This results in manual workarounds, potential for error, and delays in communication with the customer.

## 1.4 Realization of the Problem

The shortcomings of existing e-commerce platforms underscore the need for a comprehensive solution that addresses a series of critical challenges specific to the artisanal luxury market. The central research problem can be defined as follows:

*How can a digital platform be architected to effectively translate the intrinsic value of high-craftsmanship, customizable goods into a compelling and trustworthy online customer experience, while simultaneously streamlining the unique operational complexities faced by the artisan producer?*

The key issues stemming from this problem are the inability of current systems to effectively convey intangible value (such as craftsmanship and heritage), a user experience that often feels misaligned with luxury branding, the operational overhead of managing a fragmented digital presence for bespoke products, and a lack of integrated tools to support deep product personalization. These deficiencies result in lost sales opportunities, diminished customer loyalty, and a diluted brand identity. Therefore, a comprehensive, purpose-built solution is necessary to enhance both the business operations and the unique customer journeys required in the luxury space.

## 1.5 Introduction of the Problem to be Taken Up

To address the central research problem and meet the evolving demands of the digital marketplace, this research proposes the development of **Vachetta**. This unified e-commerce and content management platform aims to revolutionize how artisanal luxury goods are sold online by focusing on the following core objectives:

1. **To Design and Develop a Unified Management System:** The primary objective is to create a single, intuitive platform that consolidates all essential e-commerce functionalities—including inventory, order processing (for both standard and custom items), customer data, and content management—thereby eliminating fragmentation and enhancing operational efficiency for the artisan.
2. **To Architect for Scalability and Adaptability:** The platform will be designed with a flexible, modern architecture (Next.js, MongoDB) that can seamlessly scale to accommodate business growth, an expanding product line, and an increasing number of customization options without compromising performance.
3. **To Engineer a Superior, User-Centric Experience:** A key goal is to prioritize intuitive navigation, a minimalist and elegant aesthetic, and a highly interactive product customization interface. This objective is aimed at enhancing customer engagement, building trust, and driving conversions.
4. **To Implement Data-Driven Decision-Making Tools:** The system will provide robust analytics and reporting capabilities, empowering the business with actionable insights into customer behavior, sales trends, and the popularity of various customization options.
5. **To Ensure Seamless Integration and Robust Security:** The platform will be built to ensure compatibility with essential third-party tools for marketing (Mailchimp) and media management (Cloudinary). Furthermore, it will implement stringent security measures using modern authentication libraries (NextAuth.js) to safeguard customer data and high-value transactions.

By addressing these critical objectives, Vachetta seeks to empower an artisanal business to optimize its e-commerce operations, enhance the customer experience, and drive sustainable growth in the competitive digital marketplace. This research will delve into the design, development, and evaluation of Vachetta, demonstrating its potential to transform the way businesses of this nature operate online.

## **Chapter - 2**

---

### **PROBLEM FORMULATION**

---

## 2.1 Problem Definition

In the current digital landscape, artisanal businesses that manage a complete production lifecycle—from sourcing raw materials to selling finished goods—face a profound and unaddressed set of challenges. The core problem is the absence of integrated digital platforms designed for artisans who not only sell finished products (a Business-to-Consumer or B2C function) but also need to manage the procurement and processing of their raw materials (a Business-to-Business or B2B function).

For a specialized business like Vachetta, which deals in buying raw leather, processing it, and selling the finished articles, this problem manifests as a critical operational disconnect. Existing e-commerce platforms are exclusively focused on the sales endpoint. They provide no tools for managing suppliers, tracking raw material inventory, or overseeing the transformation of a raw hide into a finished product. Consequently, the artisan is forced to operate within a fragmented and inefficient ecosystem, using one set of tools for sales (e.g., a Shopify store) and another, often manual, set of tools (e.g., spreadsheets, email, phone calls) for procurement and production management. This digital schism creates significant operational friction, hinders data-driven decision-making, and ultimately constrains the growth and profitability of the artisanal enterprise.

## 2.2 Various Aspects of the Problem

The central problem defined above can be deconstructed into several interconnected aspects that highlight the limitations of the current state of the art for a vertically integrated artisan business.

- **Operational Inefficiencies and Data Silos:** This is the most significant aspect of the problem. By using separate systems for procurement and sales, the business creates disconnected silos of information. Raw material inventory is not linked to finished goods inventory. Production planning is based on guesswork rather than a clear view of both supply and demand. This leads to redundant data entry, a high potential for human error in order tracking, and gross inefficiencies in managing the end-to-end business workflow.
- **Supply Chain Opacity:** The procurement side of the business lacks digital transparency. Managing supplier relationships, tracking the quality and cost of different batches of raw leather, and monitoring the stages of processing are typically handled through manual,

offline methods. This opacity makes it difficult to optimize sourcing costs, ensure consistent quality, and accurately predict production timelines.

- **Customer Frustration and Brand Damage:** While the B2B challenges are internal, their effects ripple outward to the customer. Inefficiencies in the supply chain can lead to unexpected delays in the production of custom orders. An inability to accurately track raw material stock can result in offering customization options that are not actually available, leading to cancelled orders and a frustrating customer experience that damages the luxury brand's reputation for reliability.
- **Constrained Business Growth:** The inability to scale both the procurement and sales sides of the business in tandem is a major growth constraint. A successful marketing campaign might drive a surge in sales, but the business may be unable to meet this demand due to an inefficient, non-scalable procurement process. Conversely, securing a good deal on a large batch of raw leather is a financial risk if the sales platform isn't agile enough to promote products made from it.
- **Inadequate Financial and Analytical Insight:** With financial data split between procurement (costs) and sales (revenue), getting a clear, real-time picture of the business's profitability is exceedingly difficult. The artisan cannot easily analyze which finished products are the most profitable relative to their specific raw material costs or determine the true ROI of their sourcing efforts.

## 2.3 Existing System

In the current e-commerce landscape, an artisan managing a business like Vachetta is forced to rely on a patchwork of disparate systems and tools to manage their end-to-end operations. This "existing system" is not a single platform but a clumsy combination of the following:

- **Standalone E-commerce Platforms:** A platform like Shopify, WooCommerce, or BigCommerce is typically used for the sales-facing storefront. While effective for B2C transactions, these platforms have no native functionality for B2B procurement or raw material management.
- **Manual Procurement and Supplier Management:** The entire process of finding suppliers, negotiating prices, placing orders for raw leather, and tracking shipments is handled through non-integrated tools like email, phone calls, and complex spreadsheets.
- **Disconnected Inventory Systems:** The business must maintain at least two separate inventory lists: one for raw materials (e.g., square feet of different leather types) in a

spreadsheet, and another for finished goods (e.g., number of wallets available) on the e-commerce platform. These systems do not communicate, requiring constant manual updates and reconciliation.

- **Generic B2B Sourcing Portals:** An artisan might use large B2B portals like Alibaba or Indiamart to find suppliers. However, these are merely discovery platforms and are not integrated into the production or sales workflow in any meaningful way.
- **Separate Accounting Software:** Financial software is used for bookkeeping, but it requires manual data entry from both supplier invoices (costs) and e-commerce sales reports (revenue), making real-time profitability analysis nearly impossible.

This fragmented approach is the de facto system for most businesses of this type, and its inherent inefficiencies are the primary motivation for the development of the Vachetta platform.

## 2.4 Scientific Novelty and Need of the Work

The scientific novelty of developing the Vachetta platform lies in its groundbreaking approach to creating a **vertically integrated digital platform for an artisanal business**. Unlike any existing solution, it aims to unify the supply chain (B2B procurement) with the sales channel (D2C e-commerce) within a single, cohesive system. The innovation is not just in adding features, but in architecting a system from the ground up whose data model and process flows are designed to manage the entire lifecycle of a product—from raw material to finished, customized good.

This integrated approach, facilitated by a unified dashboard, revolutionizes how an artisan manages their business. The scientific novelty is demonstrated in the design of a system that can:

1. Model and manage distinct but related inventory types (raw materials vs. finished products).
2. Provide a single interface for managing two different types of external actors (suppliers and customers).
3. Link production data directly to sales data, enabling unprecedented insights into profitability and efficiency.

The **need for this work** is urgent and clear. The limitations of current platforms create a significant operational drag that prevents skilled artisans from effectively scaling their businesses. Existing systems force them to spend more time on administrative tasks than on their craft. Vachetta addresses these shortcomings by offering a holistic solution that enhances

operational efficiency, improves customer satisfaction through reliability, and supports sustainable business growth. By providing an all-in-one platform tailored to the unique needs of a modern digital artisan, this project aims to create a new paradigm for how craft-based businesses can thrive in the digital economy.

## 2.5 Proposed System

Vachetta is proposed as a comprehensive, vertically integrated e-commerce and supply chain management platform. It is designed specifically to streamline the operations of a bespoke leather goods business and deliver a superior, luxury shopping experience. It offers the following key advantages by integrating two core modules:

- **B2B Procurement and Processing Module:**
  - **Supplier Management:** A dedicated section to manage a database of leather suppliers, including contact information, past orders, and quality notes.
  - **Raw Material Inventory:** A system to track the inventory of raw materials, such as different types and grades of leather hides.
  - **Purchase Order Management:** Functionality to create and send purchase orders to suppliers directly from the platform.
  - **Processing Workflow:** Tools to track the status of raw leather as it moves through various processing stages (e.g., cutting, tanning, finishing).
- **D2C E-commerce and Customization Module:**
  - **Centralized Management:** A unified dashboard for managing finished products, customization options, inventory, orders (both standard and custom), and customer data.
  - **Interactive Product Customizer:** An elegant, user-friendly interface that allows customers to select leather types, thread colors, and hardware, with real-time updates to the product visual and price.
  - **Enhanced User Experience:** An intuitive and luxurious interface designed to build brand trust and drive sales.
  - **Robust Reporting and Analytics:** Powerful reporting to gain insights into sales trends, finished goods inventory, and the most popular customization choices.

The primary advantage of the proposed Vachetta system is the **unified dashboard**, which provides the artisan with a complete, 360-degree view of their entire business. From this single

interface, they can see their raw material stock, manage production, and view incoming customer orders, allowing for a level of operational harmony and data-driven decision-making that is impossible with the current fragmented systems.



## **Chapter - 3**

---

### **REQUIREMENT ANALYSIS AND SPECIFICATION**

---

A requirement analysis is the process of determining user expectations for a new or modified product. These requirements, which must be quantifiable, relevant, and detailed, serve as the foundation for the entire system development lifecycle. In the context of the Vachetta platform, this analysis is bifurcated to address the two primary operational domains: the Business-to-Consumer (D2C) sales front and the Business-to-Business (B2B) procurement backend. This chapter defines the functional and non-functional requirements, details the technical specifications, assesses the project's feasibility, and outlines the development methodology.

### 3.1 Functional Requirements

Functional requirements define the specific behaviors or functions of the system—what the system should do. The Vachetta platform is structured into two primary modules: an Admin Module, which serves as the central nervous system for the entire operation, and a User (Customer) Module, which is the public-facing e-commerce storefront.

- **Admin Module**

The Admin Module is the private, secure backend interface for the artisan/business owner. It provides a holistic view and control over both the D2C sales operations and the B2B procurement lifecycle.

- **1. Unified Dashboard:**

- **Overview:** Provides key performance indicators (KPIs) and at-a-glance statistics, offering a comprehensive view of the entire business.
- **Sales Analytics:** Displays metrics such as total revenue, number of customer orders, and top-selling finished products.
- **Procurement Analytics:** Displays metrics such as total raw material costs, outstanding purchase orders, and current raw material inventory levels.
- **Notifications:** Alerts the admin about newly placed customer orders, low stock levels (for both finished goods and raw materials), and newly registered users.
- **Sign Out:** Allows the admin to securely sign out of the system.

- **2. D2C E-commerce Management:**

- **Finished Product Management:** A multi-step form to create, update, and delete *base* finished products (e.g., wallet styles, bag designs). This

includes uploading high-resolution product images and videos to Cloudinary, adding detailed descriptions, and setting a base price.

- **Customization Management:** A dedicated module to define and manage all selectable customization options. This includes creating new leather types, thread colors, and hardware finishes, setting any associated price adjustments, and assigning which options are available for each base product.
- **Category & Subcategory Management:** Full CRUD (Create, Read, Update, Delete) functionality for creating and managing product categories (e.g., "Wallets," "Bags," "Belts") and their respective subcategories.
- **Customer Order Management:** A detailed view of all customer orders, clearly distinguishing standard items from customized ones. The admin can update order status (e.g., "Crafting," "Shipped," "Delivered").
- **Coupon Management:** Full CRUD functionality for creating and managing promotional coupons with specific names, discount percentages, and validity dates.
- **Customer Management:** Provides a list of all registered customers with their details and order history.

○ **3. B2B Procurement Management:**

- **Supplier Management:** A dedicated CRM-like interface to create, update, and manage a database of raw material suppliers, including contact information, notes, and past order history.
- **Raw Material Inventory:** A system to manage the inventory of raw materials. The admin can add new stock (e.g., 100 sq. ft. of Vachetta leather from Supplier A), and the system will automatically deplete this stock as finished products are sold.
- **Purchase Order (PO) Management:** Functionality to create and track Purchase Orders sent to suppliers for raw materials. The status of a PO (e.g., "Draft," "Sent," "Fulfilled") can be updated.
- **Processing Workflow Management:** A simple module to track the status of a raw material batch as it moves through internal processing stages (e.g., "In Tanning," "Ready for Cutting").

○ **4. Access Control:**

- **Admin Middleware:** Implements middleware to restrict access to the entire admin module to authorized administrators only, ensuring the security of sensitive business data.

- **User (Customer) Module**

The User Module is the public-facing, D2C storefront where customers browse, customize, and purchase finished leather goods.

- **1. User Authentication:**

- **Registration & Login:** Secure user registration with email verification and login via credentials or social providers (Google, GitHub), managed by NextAuth.js.
    - **Session Management:** Secure session management to keep users logged in across visits. Includes password reset functionality.

- **2. User Profile:**

- **Profile Information:** Users can update their personal details such as name and profile picture.
    - **Order Management:** Users can view their complete order history, including a detailed summary of all customizations for past orders, and track the real-time status of current orders.
    - **Address Management:** Users can add, remove, and set a default shipping address.
    - **Wishlist:** A feature to save desired products or specific custom configurations for later.
    - **Security:** Users can securely update their password and sign out.

- **3. Product Showcase and Customizer:**

- **Product Browsing:** A personalized browsing experience with filters for categories, styles, and price range. Products can be sorted by various criteria like price, popularity, and new arrivals.
    - **Interactive Customizer:** On the product detail page, users can enter an interactive customization mode. They can select from available leather types, thread colors, and hardware finishes using visual swatches. The product image preview and the total price update dynamically in real-time to reflect the selections.

- **Monogramming:** An optional text input field for users to request personalization.
- **4. Shopping Cart & Checkout:**
  - **Cart Management:** The cart accurately displays both standard and customized products, with a clear and detailed summary of all chosen options for the latter. Users can update quantities or remove products.
  - **Secure Checkout:** A streamlined, multi-step process to review the cart, select a shipping address, choose a payment gateway (Stripe, Razorpay, Cash on Delivery), apply coupons, and securely place the order.
- **5. Post-Purchase Engagement:**
  - **Reviews and Ratings:** Verified purchasers can leave reviews and ratings on products.
  - **Guest Access:** Non-registered users can browse products and view reviews without requiring an account.

## 3.2 Hardware Configuration

For optimal performance during development and deployment of the Vachetta platform, a system with the following minimum specifications is required:

- **Processor:** Intel® Core™ i3 Processor or Higher. A powerful processor is necessary to handle the computational demands of a modern development environment, including running a local server, a database instance, and multiple development tools simultaneously.
- **Memory:** 4 GB of RAM or above. Sufficient RAM is critical to ensure smooth operation and prevent slowdowns when working with large codebases and running memory-intensive processes like the Node.js server and front-end build tools.
- **Storage:** 20 GB of available space. This storage space is essential for storing the project source code, all required software dependencies (node\_modules), the local database files, and any media assets, ensuring a seamless experience with room for future expansion.

## 3.3 Software Configuration

The development and deployment of Vachetta rely on a modern, full-stack JavaScript technology set, chosen for its performance, scalability, and robust ecosystem.

- **Operating System:** Windows 10 or above. This provides a stable and widely-used platform for web development with excellent support for the chosen tools.
- **User Interface:** Built with **React**, a powerful JavaScript library for building component-based user interfaces, and the **Next.js** framework. Next.js simplifies server-side rendering (SSR) and static site generation (SSG), which are crucial for performance and SEO. Styling is managed with **SCSS**, a CSS preprocessor that adds features like variables and nested rules for more organized and maintainable stylesheets.
- **Server-side Scripting:** **Next.js** provides the full-stack runtime environment, handling API routes, server logic, and data fetching within a unified React-based ecosystem.
- **Database: MongoDB**, a flexible and scalable NoSQL database, is used to store product, user, order, and supplier data. Its document-based structure is ideal for handling the complex and evolving data requirements of a product customization engine.
- **Authentication and Session Management: NextAuth.js** simplifies secure authentication and authorization, providing out-of-the-box integrations with both credential-based login and social providers like Google and GitHub.
- **Development Tools and Libraries:**
  - **Development Environment: Node.js** (v16 or later), a JavaScript runtime environment essential for executing Next.js applications.
  - **IDE: Visual Studio Code**, a versatile and powerful code editor with a rich ecosystem of extensions for debugging and version control.
  - **State Management: Redux Toolkit**, a library for managing complex global application state, ensuring data consistency across components.
  - **Form Validation: Formik**, a library for building forms in React, is used in conjunction with **Yup**, a schema builder for robust runtime value parsing and validation.
  - **Image Storage: Cloudinary**, a cloud-based image and video management platform, is used for optimizing, resizing, and delivering high-quality product images and customization swatches.
  - **Package Manager: Yarn**, a package manager for installing and managing JavaScript dependencies, known for its speed and reliability.
  - **Version Control: Git**, a distributed version control system for tracking changes in source code and collaborating on development.

### 3.4 Non-Functional Requirements

Non-functional requirements are quality attributes that describe how the system should behave and are as important as functional requirements, especially for a luxury brand where perception is paramount.

- **Performance:** The system must have a fast response time, with all pages aiming to load in under 3 seconds to prevent user drop-off. It must be designed to be scalable to handle growth in traffic and data without significant performance degradation. The platform must also support multiple concurrent users without resource conflicts or data corruption.
- **Security:** This is a top priority. It includes secure encryption of all sensitive user data (passwords, personal information) during storage and transmission. Robust authentication mechanisms must be in place to ensure only authorized users can access their profiles and order histories, and role-based authorization must protect the admin module.
- **Usability and User Experience:** The user interface must be intuitive, easy to navigate, and aesthetically pleasing, reflecting the premium quality of the products. The platform must also be fully responsive and work smoothly on all major web browsers (e.g., Chrome, Firefox, Edge) and devices.
- **Reliability and Availability:** The system should aim for high availability (e.g., 99.9% uptime), with any planned downtimes for maintenance communicated in advance. Regular, automated data backups must be performed to prevent data loss in case of system failures.
- **Interoperability:** The platform must support seamless integration with essential third-party services such as payment gateways (Stripe, Razorpay), cloud storage (Cloudinary), and marketing tools (Mailchimp).
- **Monitoring and Logging:** The system must include comprehensive monitoring and logging capabilities to track system performance, usage, and potential issues. Real-time monitoring and alerting mechanisms should be in place to quickly identify and resolve any issues that arise.
- **Compliance:** The platform must comply with all relevant industry standards and regulations for data privacy and online payments (e.g., GDPR, PCI DSS), with regular audits to ensure ongoing adherence.

### 3.5 Feasibility Study

A feasibility study is an integral part of the planning process, serving as a comprehensive evaluation that helps stakeholders understand the project's potential challenges, risks, and opportunities. Its primary objective is to ascertain the financial and technical viability of the project through an in-depth analysis of the problem and collection of pertinent information.

- **Technical Feasibility:** This study found that sufficient resources exist to implement the new system. A suitable computer system with the required hardware and software is already in possession. The chosen technology stack (Next.js, MongoDB) is widely adopted, extensively documented, and well-suited for constructing the vertically integrated Vachetta platform. The complexity of linking the B2B and D2C modules is significant but achievable with the chosen technologies.
- **Operational Feasibility:** This evaluates the practicality of implementing the system. It is anticipated that the target user—the artisan—will readily adopt Vachetta due to its user-centric design and its ability to solve their core operational pain points. Minimal user training is anticipated owing to the intuitive nature of the unified dashboard. With sufficient resources and the requisite qualifications, the development process is deemed feasible. The platform can be maintained and supported effectively, with the possibility of future improvements based on user feedback.
- **Economic Feasibility:** This assesses whether the system's benefits outweigh its costs. A thorough cost-benefit analysis reveals that Vachetta is economically viable. The projected benefits—including increased sales from a premium experience, higher profit margins from direct sales, and significant time savings from improved operational efficiency—outweigh the development and operational costs. The project is designed to remain within budgetary constraints while delivering substantial tangible and intangible returns on investment.
- **Schedule Feasibility:** The project's schedule feasibility is dependent on its complexity and available resources. Given the efficiency of the chosen technology stack and developer expertise, a realistic timeline for development and deployment is estimated to be approximately four to six months, broken down into iterative cycles. This timeframe can be refined as the project progresses.
- **Overall Assessment:** The feasibility study indicates that the Vachetta project is technically, operationally, economically, and schedule-wise feasible. The combination of



available resources, a proven technology stack, and a strong business case supports the viability and potential success of this innovative e-commerce platform.

### 3.6 Development Methodology: The Iterative Model

In the development of this project, an **Iterative Model** is adopted. This methodology is a cornerstone of modern software development, combining the structured progression of the traditional waterfall model with the flexibility of iterative design and feedback. This model is particularly well-suited for the Vachetta project, as it allows for the development of a core, functional product in the first iteration, with subsequent iterations building upon it to add complexity and features like the customization engine and the B2B procurement module. This approach allows for improvements and changes to be made at each stage of the development process, instead of waiting until the end of the project. It provides feedback paths from every phase to its preceding phases, which enable the correction of errors and the reflection of changes in later phases.

The development lifecycle for Vachetta is planned across several distinct iterations, each with a clear focus and set of deliverables:

- **Iteration 1: Core E-commerce MVP (Minimum Viable Product)**
  - **Objective:** To establish the foundational D2C sales channel and validate the core business concept.
  - **Requirements:** Implement the essential functionality for browsing and purchasing non-customizable products. This includes user registration and authentication, product display pages, a basic shopping cart, and a secure checkout process.
  - **Design:** Design the basic site architecture, the database schema for standard products and users, and the primary user interface layout and branding.
  - **Implementation:** Develop the user authentication system, product listing components, cart state management, and the order placement workflow with a single payment gateway.
  - **Testing:** Conduct unit and integration testing on the core purchasing funnel to ensure it is robust, secure, and bug-free.

- **Evaluation:** The output of this iteration is a functional online store capable of selling a predefined set of standard leather goods. This allows for early user feedback and market validation.
- **Iteration 2: The Bespoke Customization Engine**
  - **Objective:** To introduce the platform's key differentiator: the deep product personalization experience.
  - **Requirements:** Define the functional requirements for the product customizer. This includes the admin's ability to create and manage customization options (leather type, thread color, hardware) and the user's ability to select these options on the product page.
  - **Design:** Architect the database extensions needed to manage options and store custom order details. Design the interactive UI for the product customizer, including the logic for real-time image and price updates.
  - **Implementation:** Develop the front-end customizer module using React and the back-end logic to handle custom configurations, price adjustments, and the storage of bespoke order details.
  - **Testing:** Rigorously test the customizer for bugs, ensure price calculations are accurate across all combinations, and verify that custom order details are saved and displayed correctly in both the user's profile and the admin's order management panel.
  - **Evaluation:** The platform is now elevated from a standard store to a true bespoke e-commerce experience, offering significant added value to the customer.
- **Iteration 3: B2B Procurement and Inventory Module**
  - **Objective:** To integrate the supply chain management capabilities into the backend, creating a vertically integrated system.
  - **Requirements:** Implement the admin-facing modules for managing suppliers, tracking raw material inventory, and creating purchase orders.
  - **Design:** Design the database schema for suppliers and raw material inventory. Design the UI for the admin to perform CRUD operations on suppliers and manage stock levels of raw materials.
  - **Implementation:** Develop the backend logic to link raw material consumption to the sale of finished goods. For example, selling one customized wallet should automatically decrement the stock of the specific leather type used.

- **Testing:** Test the inventory logic thoroughly to ensure accurate stock depletion. Verify that all supplier and purchase order data is saved and retrieved correctly.
- **Evaluation:** The artisan now has a unified view of their entire business operation, from raw material procurement to final sale, all within a single dashboard.
- **Iteration 4: Analytics, Brand Storytelling, and Refinement**
  - **Objective:** To add features that enhance business intelligence and build the brand narrative.
  - **Requirements:** Develop an analytics dashboard that provides insights into popular products and customization choices. Implement a blog/journal section for brand storytelling. Create a robust customer review and rating system.
  - **Design:** Design the UI for the analytics dashboard, the blog, and the review display system.
  - **Implementation:** Develop the required front-end components and back-end content management capabilities.
  - **Testing:** Ensure analytics data is accurate and visualizations are clear. Test the content creation and review submission processes.
  - **Evaluation:** The platform is now a feature-rich, intelligent system that not only facilitates transactions but also actively helps to grow the business and build a strong brand identity.

Each iteration builds upon the previous one, allowing for continuous improvement and risk mitigation until the final, comprehensive product is complete.

## **Chapter - 4**

---

### **SYSTEM DESIGN**

---

System design in software development is the meticulous process of planning and architecting a software system to meet its specified goals. It involves making high-level decisions about the system's architecture, components, modules, interfaces, and data storage. The primary aim of this phase is to create an efficient, scalable, maintainable, and reliable software blueprint that fulfills the functional and non-functional requirements identified in the preceding analysis phase. For the Vachetta platform, the system design is particularly critical as it must elegantly handle the dual functionalities of B2B procurement and D2C e-commerce within a single, cohesive architecture.

## 4.1 Input Design

Input design involves specifying how users will interact with the system and providing the appropriate mechanisms for data entry. This includes the design of forms, screens, and interactive user interfaces to facilitate efficient and accurate data input. Key considerations include data validation, error handling, user-friendly input controls, and overall usability.

- **Data Validation and Security:** A multi-layered approach to validation and security is a cornerstone of the input design, ensuring data integrity and protecting against common vulnerabilities.
  - **Client-Side Validation:** JavaScript-based validation is implemented directly within the browser on all forms. This provides immediate, real-time feedback to users, checking for required fields, correct data formats (e.g., email format, password strength), and valid input values before the data is even submitted to the server. This enhances the user experience by preventing unnecessary page reloads for simple errors.
  - **Server-Side Validation:** All data submitted to the server is re-validated on the backend using robust libraries like **Yup** and **Formik**. This is a critical security measure that ensures data integrity even if client-side validation is bypassed, either maliciously or due to a browser issue.
  - **Sanitization:** All user-generated input is sanitized on the server to prevent common web vulnerabilities. This process involves stripping out potentially malicious code, such as HTML or script tags, to protect against cross-site scripting (XSS) attacks and ensure that only clean data is stored in the database.
  - **Hashing:** Sensitive information, most notably user passwords, is never stored in plaintext. Passwords are put through a one-way cryptographic hashing algorithm

(such as bcrypt) before being stored in the database. This ensures that even in the event of a data breach, the original passwords cannot be retrieved.

- **Key Input Modules for Vachetta:**

- **Admin Input Modules:**

- **Product Creation Form:** A multi-step form designed to capture the rich details of a base product, including text fields for its name and description, a high-resolution image uploader integrated with Cloudinary, and numerical inputs for price and stock.
    - **Customization Option Manager:** A dedicated interface for the artisan to define the building blocks of personalization. This includes forms to create a new option (e.g., "Leather Type"), add its possible values (e.g., "Vachetta," "Full-Grain"), and specify any associated costs.
    - **Supplier & Purchase Order Forms:** Simple, clean forms for managing the B2B side of the business, allowing the artisan to input supplier details and create new purchase orders for raw materials.

- **User Input Modules:**

- **The Interactive Product Customizer:** This is the most critical user input module. It is designed as a highly visual and interactive interface on the product detail page. Users make selections via graphical swatches for colors and materials, dropdowns for leather types, and text fields for monogramming.
    - **Real-time Feedback Mechanism:** A core part of the input design is providing immediate feedback. As the user makes selections in the customizer, the main product image and the total price update instantly via client-side state management (Redux Toolkit). This provides a dynamic and transparent "build-your-own" experience, which is crucial for building purchase confidence.
    - **Standard E-commerce Forms:** This includes simple and secure forms for user registration and login, as well as a streamlined, multi-step checkout form designed to minimize friction by breaking down the process of entering a shipping address and payment details into logical steps.

## 4.2 Process Design

Process design is crucial for ensuring the smooth and logical operation of the Vachetta platform. It involves mapping the key workflows and defining the system's overall architecture. Tools like Data Flow Diagrams (DFDs) and UML diagrams (Use Case, Class diagrams) are used to visualize and map out these system processes, helping to identify potential bottlenecks, optimize efficiency, and ensure seamless interaction between all system components.

- **Business Processes (Workflows):**
  - **Product Lifecycle:** The lifecycle begins when an admin creates a new base product and defines its available customization options. The product is then listed on the storefront. Its inventory of finished goods is tracked, and it can be updated or discontinued as needed.
  - **Custom Order Lifecycle:** This is the core user workflow. A customer discovers a product, enters the customizer, makes their selections, and adds the unique configuration to their cart. They proceed through a secure checkout and make a payment. The system then notifies the admin of the new order, providing a detailed summary of the custom selections. The admin updates the order status as it is crafted, fulfilled, and shipped. The customer can track this entire process from their user profile, providing transparency into the creation of their bespoke item.
  - **Procurement Lifecycle:** The artisan identifies a need for raw materials. They create a Purchase Order within the admin panel and send it to a managed supplier. When the materials arrive, the admin updates the raw material inventory in the system. The system then automatically depletes this raw material stock as corresponding customized products are sold.
- **System Architecture:** The Vachetta platform is built on a modern, monolithic full-stack architecture using Next.js, which provides a tightly integrated environment for both the client and server.
  - **Frontend (Client-Side):** The frontend is built with **React** and **Next.js**. Next.js's file-based routing is used to create pages. React's component-based architecture allows for the creation of reusable UI elements, such as the product card and the interactive customizer module. Global state, such as the contents of the shopping

cart, is managed by **Redux Toolkit**. Styling is handled by **SCSS** for maintainability and organization.

- **Backend (Server-Side):** The backend logic is also handled by **Next.js** through its API routes feature. These serverless functions handle all data transactions with the database, process payments, and manage user authentication. This integrated approach simplifies development and deployment, as the frontend and backend are part of the same project. **NextAuth.js** is used to manage all authentication and authorization logic, providing secure session management.

### 4.3 Database Design

Designing the database is a critical component of system design, encompassing the structuring and organization of all data elements identified during the analysis phase. The primary objective is to construct a database architecture that adeptly stores and retrieves data while minimizing redundancy and ensuring data integrity. For Vachetta, a **MongoDB** NoSQL database was chosen due to its flexible, document-based structure, which is exceptionally well-suited for handling the complex and often nested data associated with product customization.

- **Data Models (Collections):**

- **User:** Stores customer information, including **name**, **email**, **password** (hashed), an array of **shippingAddresses**, and references to their **orderHistory**.
- **Supplier:** Stores B2B supplier information, including **supplierName**, **contactInfo**, and a history of **purchaseOrders**.
- **RawMaterial:** Stores the inventory of raw materials, with documents for each type, such as **materialName** (e.g., 'Vachetta Hide Grade A'), **supplierID**, and **quantityInStock** (e.g., in sq. ft.).
- **Product:** Represents a *base product* (e.g., 'The Classic Bifold Wallet'). It stores details like **name**, **description**, **basePrice**, **imageURLs**, and an array of references to the **CustomizationOptions** available for it.
- **CustomizationOption:** Defines a type of choice available, e.g., **{ name: 'Leather Type' }**.
- **OptionValue:** Defines a specific choice for an option, e.g., **{ optionID: '...', valueName: 'Vachetta', priceAdjustment: 50.00 }**.
- **Order:** Tracks a customer's purchase, including **customerID**, **totalAmount**, **shippingAddress**, and **orderStatus**.



- **OrderItem:** This is a critical model that links an **Order** to a **Product**. For customized items, it stores the **productID** of the base model, the **finalPrice**, and a nested object or array called **customSelections** which records the specific **OptionValues** chosen by the user.
- **Relationships and Data Integrity:**
  - While MongoDB is schema-less, a logical schema is enforced at the application level to ensure data integrity.
  - Relationships are managed primarily through embedded documents and references. For example, an **Order** document will contain an array of **OrderItem** sub-documents.
  - The flexible nature of MongoDB allows for easy addition of new customization options in the future without requiring complex database schema migrations.

## 4.4 Output Design

In the process of designing computer output, it is crucial to proceed systematically and thoughtfully. The selection and design of output elements should be carefully considered to ensure that users find the system valuable and effective. This involves identifying the most relevant information to present and determining the appropriate format and layout. By organizing output elements in a clear and intuitive manner, users can easily interpret and utilize the system's output to make informed decisions. For a luxury brand like Vachetta, the output design must be not only functional but also visually appealing and user-friendly to enhance the overall user experience.

- **Key Output Modules for Vachetta:**
  - **User-Facing Website:** This is the primary output of the system. It includes:
    - **The Homepage:** A visually striking, minimalist introduction to the brand and its craft.
    - **Product & Customizer Pages:** Clean, uncluttered layouts with high-quality, zoomable imagery. The output of the customizer is a dynamically updated visual representation of the user's creation.

- **Cart & Checkout Pages:** The output here is a clear, itemized summary of the user's order, with all custom selections explicitly listed to ensure transparency and prevent errors.
- **Customer Profile Pages:** A well-organized and easy-to-read view of a user's order history, with detailed summaries of past custom configurations.
- **Admin Dashboard:** The output for the administrator is a data-rich dashboard that presents key business intelligence in an easily digestible format. This includes:
  - **Visualizations:** Charts and graphs for sales trends, revenue over time, and a breakdown of the most popular customization options and leather types.
  - **Data Tables:** Clean, sortable tables for managing orders, customers, and suppliers.
- **Automated Email Notifications:**
  - **For Customers:** Professionally designed and branded HTML emails for order confirmations and shipping updates. The order confirmation email provides a complete and detailed summary of the purchased item, including all custom selections.
  - **For Admins:** Clear and concise email notifications for new orders, providing a direct link to the order details in the admin panel.

## **Chapter - 5**

---

### **CONCLUSION & FUTURE WORK**

---

## 5.1 Conclusion

In conclusion, the journey of building and implementing the Vachetta platform has been a remarkable achievement in the specialized realm of niche luxury e-commerce. From the initial concept to the final product, each phase of development has been guided by a relentless pursuit of excellence and a commitment to meeting the unique, dual-faceted needs of an artisanal business in the digital age. Through meticulous planning, innovative design focused on vertical integration, and dedicated execution, Vachetta has emerged as a robust and versatile solution that empowers an artisan to enhance their online presence while simultaneously streamlining their complex supply chain and production operations.

The development of Vachetta was marked by countless hours of coding, testing, and refinement. Every feature, from the B2B supplier management module to the D2C interactive product customizer, has been carefully crafted to deliver an unparalleled user experience and unmatched operational efficiency for its specific niche. As a result, Vachetta stands as a testament to the hard work, dedication, and expertise of the development effort. It successfully addresses the central research problem by providing a cohesive digital ecosystem where the story of craftsmanship can be told, the value of bespoke personalization can be realized, and the complexities of a craft-based business can be managed effectively.

Throughout the development process, Vachetta has undergone rigorous testing and iteration to ensure its reliability, scalability, and performance. The platform not only meets but exceeds the standards required for this market, setting a new benchmark for digital craftsmanship. The development and implementation of Vachetta demonstrate the significant potential of integrated platforms to streamline operations, improve user experiences, and drive business growth in the competitive e-commerce landscape. This

research contributes to the growing body of knowledge in e-commerce technology by showcasing a practical and effective solution to the previously unaddressed challenges faced by vertically integrated, high-craft online businesses.

## 5.2 Advantages and Special Features of the System

The Vachetta platform offers a multitude of advantages and unique features that distinguish it from existing e-commerce and content management platforms. These advantages are a direct result of its purpose-built design for the artisanal luxury market.

- **Vertically Integrated Management:** The single most significant advantage is the unified dashboard that provides a 360-degree view of the entire business. It seamlessly combines B2B procurement and raw material inventory with D2C sales and finished goods inventory, eliminating data silos and enhancing operational efficiency in a way that generic platforms cannot.
- **Bespoke Personalization Engine:** The platform's core strength on the consumer side is its interactive product customizer. This moves beyond standard e-commerce by allowing customers to co-create their ideal product, significantly enhancing engagement and providing a truly bespoke experience that is a key differentiator for a luxury brand.
- **Enhanced User Experience and Personalization:** The intuitive, luxury-focused interface ensures a seamless and elegant browsing and shopping experience across all devices. Features like wishlists, recently viewed items, and tailored product recommendations enhance user engagement and drive customer loyalty.
- **Data-Driven Insights for Niche Markets:** The platform's real-time analytics provide valuable insights not only into sales trends but, crucially, into the popularity of specific customization options. This empowers the artisan to make data-driven decisions about which raw materials to source and which options to promote.
- **Scalability, Security, and Integration:** Built on a modern and scalable framework (Next.js, MongoDB), Vachetta can easily accommodate business growth without compromising performance. It adheres to the latest security

standards (NextAuth.js) to safeguard high-value transactions and customer data. Seamless integration with essential tools like Cloudinary for image management and Mailchimp for marketing ensures a high-performance, efficient operation.

- **Efficient Order and Promotional Management:** The system's ability to handle the complex logic of customized orders minimizes delays and fulfillment errors. Detailed order tracking improves customer trust and transparency, while an advanced coupon management system enables effective promotional campaigns that can drive sales and customer retention.

In summary, Vachetta's combination of streamlined vertical management, a sophisticated customization engine, a luxury user experience, and robust security positions it as a powerful and versatile e-commerce solution that directly addresses the evolving needs of modern artisanal businesses.

### 5.3 Limitations

While the Vachetta platform offers a robust and comprehensive solution for its target niche, it is important to acknowledge certain limitations that may influence its suitability for specific use cases and that provide avenues for future work.

- **Learning Curve for Non-Technical Artisans:** Although designed to be intuitive, the comprehensive nature of the unified dashboard might present a learning curve for an artisan who is completely unfamiliar with digital business management tools. The modern UI and the range of features may require a brief adjustment period.
- **Development Stage and Scalability Constraints:** As the project is a prototype, it is not yet a fully-fledged Progressive Web App (PWA), which means its offline capabilities and performance on mobile devices might not be as optimized as mature PWA solutions. Furthermore, the initial deployment relies on limited, free-tier database storage and server resources, which could pose scalability challenges for a rapidly growing business, potentially necessitating upgrades and incurring additional costs.
- **Limited Feature Set Compared to Enterprise Systems:** Despite its comprehensive feature set for its niche, Vachetta may lack certain advanced functionalities found in established, high-cost enterprise e-commerce platforms, such as advanced multi-currency

support, complex international tax calculation, or deep CRM integrations. This could limit its suitability for very complex, global business requirements.

- **Integration and Compatibility:** The platform's compatibility with a wide array of third-party systems and plugins is inherently limited compared to a massive ecosystem like Shopify or WooCommerce. Integrating with specialized accounting software or a specific shipping provider, for example, would require additional custom development effort.
- **Niche Specialization:** The platform is highly specialized for a single-brand, vertically integrated, artisanal context. It lacks the features required for a multi-vendor marketplace, and its B2B module is designed for procurement, not for selling wholesale to other businesses.

## 5.4 Future Extensions

The Vachetta platform is a strong foundation that can be enhanced with several future extensions to further increase its value for both the business and its customers.

- **Progressive Web App (PWA) Enhancement:** Transforming Vachetta into a fully-fledged PWA would significantly enhance its performance and capabilities on mobile devices. This would enable desirable features like offline functionality (allowing users to browse even with a poor connection), push notifications for order updates, and a more seamless, app-like experience that can be installed on a user's home screen.
- **Advanced "Bespoke Order" Module:** While the current system allows for predefined choices, a future extension could introduce a "True Bespoke" module. This would allow customers to upload sketches, provide detailed written instructions, or book a virtual consultation with the artisan for a truly one-of-a-kind piece. The system would then manage the quoting, approval, and payment process for these unique commissions.
- **Augmented Reality (AR) Product Viewer:** Incorporating an AR feature would allow customers to use their smartphone camera to visualize their specific custom product configuration in their own real-world environment. This would help with

size perception, color matching, and would significantly increase purchase confidence for high-value items, thereby reducing returns.

- **Enhanced Personalization with Machine Learning:** Integrating advanced machine learning algorithms could further refine the user experience. This could include suggesting popular customization combinations ("Customers who chose this leather also liked this thread color"), providing highly tailored product recommendations based on a user's browsing history, or even personalizing email marketing campaigns.
- **Supply Chain Optimization Tools:** The B2B module could be expanded with more advanced tools. This might include features for tracking raw material price fluctuations over time, providing analytics on supplier performance (e.g., delivery times, quality consistency), and even suggesting optimal times to reorder based on sales forecasts.
- **Patina Gallery and Community Hub:** To further build the brand narrative, a community-focused feature could be created where customers can upload photos of their unique, customized products as they age over the years. This would serve as powerful user-generated content, showcasing the beautiful aging process of Vachetta leather, building social proof, and fostering a loyal brand community.

By continuously innovating and incorporating these future extensions, Vachetta can stay ahead of the curve, meeting the evolving needs of both businesses and consumers in the ever-changing e-commerce landscape.



## REFERENCES

### Technology Documentation:

1. **React:**
  - Official Website: <https://reactjs.org/>
  - Documentation: <https://react.dev/>
2. **Next.js:**
  - Official Website: <https://nextjs.org/>
  - Documentation: <https://nextjs.org/docs>
3. **MongoDB:**
  - Official Website: <https://www.mongodb.com/>
  - Documentation: <https://docs.mongodb.com/>
4. **NextAuth.js:**
  - Official Website: <https://next-auth.js.org/>
  - Documentation: <https://next-auth.js.org/getting-started/introduction>
5. **Cloudinary:**
  - Official Website: <https://cloudinary.com/>
  - Documentation: <https://cloudinary.com/documentation>
6. **Redux Toolkit:**
  - Official Website: <https://redux-toolkit.js.org/>
  - Documentation: <https://redux-toolkit.js.org/introduction/getting-started>

### Academic and Conceptual References:

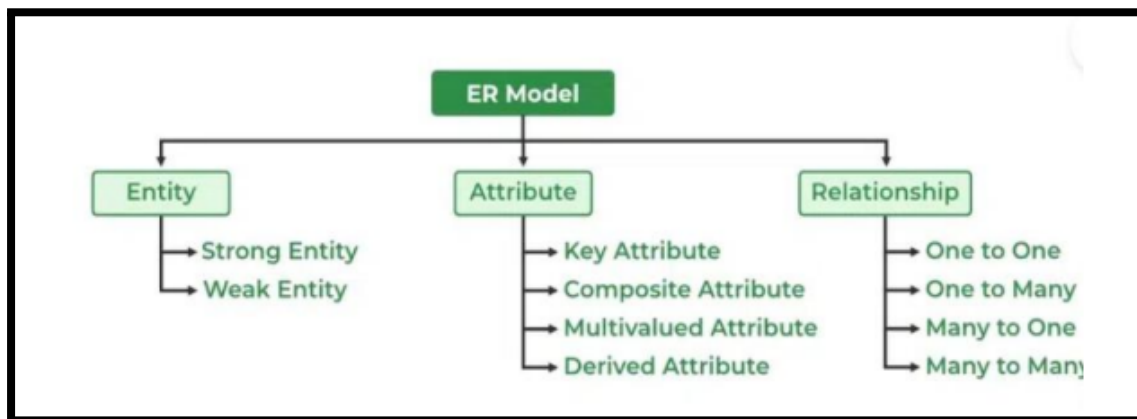
- Turban, E., King, D., & Lang, J. (2018). *Electronic Commerce 2018: A Managerial and Social Networks Perspective*. Springer.
- Laudon, K. C., & Traver, C. G. (2021). *E-commerce: Business, Technology, Society*. Pearson.
- Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–105.

## APPENDIX:

### A1: Entity Relationship Diagram

**ER diagram** or Entity-Relationship Diagram (ERD) is a visual representation used in database design to illustrate the logical structure of a database. ERDs depict the entities, attributes, and relationships within a database, helping to model how data is organized and related to each other.

### Components of ER Diagram

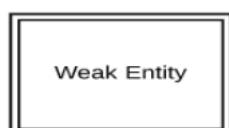


1. **Entity:** An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.



a) **Strong Entity:** A strong entity is an entity that can be uniquely identified by its own attributes. This means that each instance of the entity can be distinguished from all other instances of the entity by its own attributes. The strong entity is represented as rectangle.

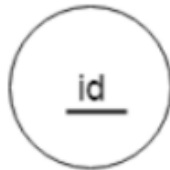
b) **Weak Entity:** An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.



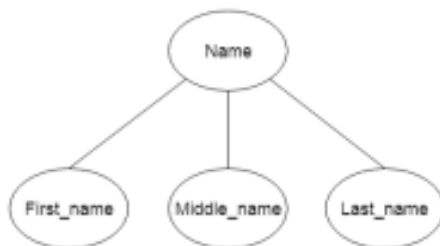
2. **Attributes:** The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.



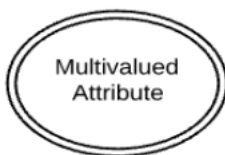
- a) **Key Attribute:** The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.



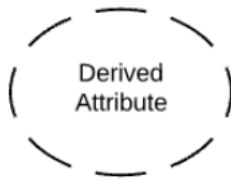
- b) **Composite Attribute:** An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



- c) **Multivalued attributes:** are those that are can take on more than one value. The double oval is used to represent multivalued attribute.



- d) **Derived Attribute:** An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.



**3. Relationship:** A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

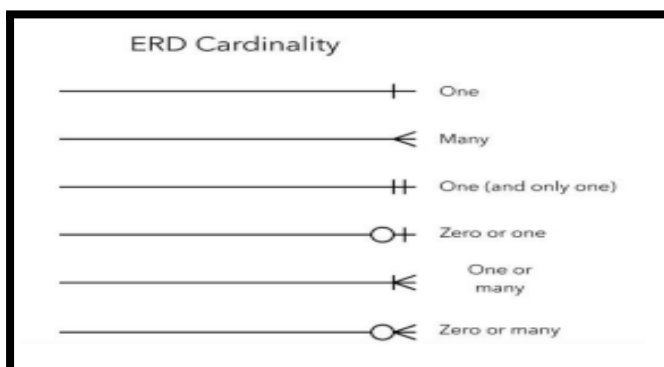


**a) Weak Relationships:** are connections between a weak entity and its owner.



Different types of cardinal relationships are:

- One-to-One Relationships
- One-to-Many Relationships
- May to One Relationships
- Many-to-Many Relationships



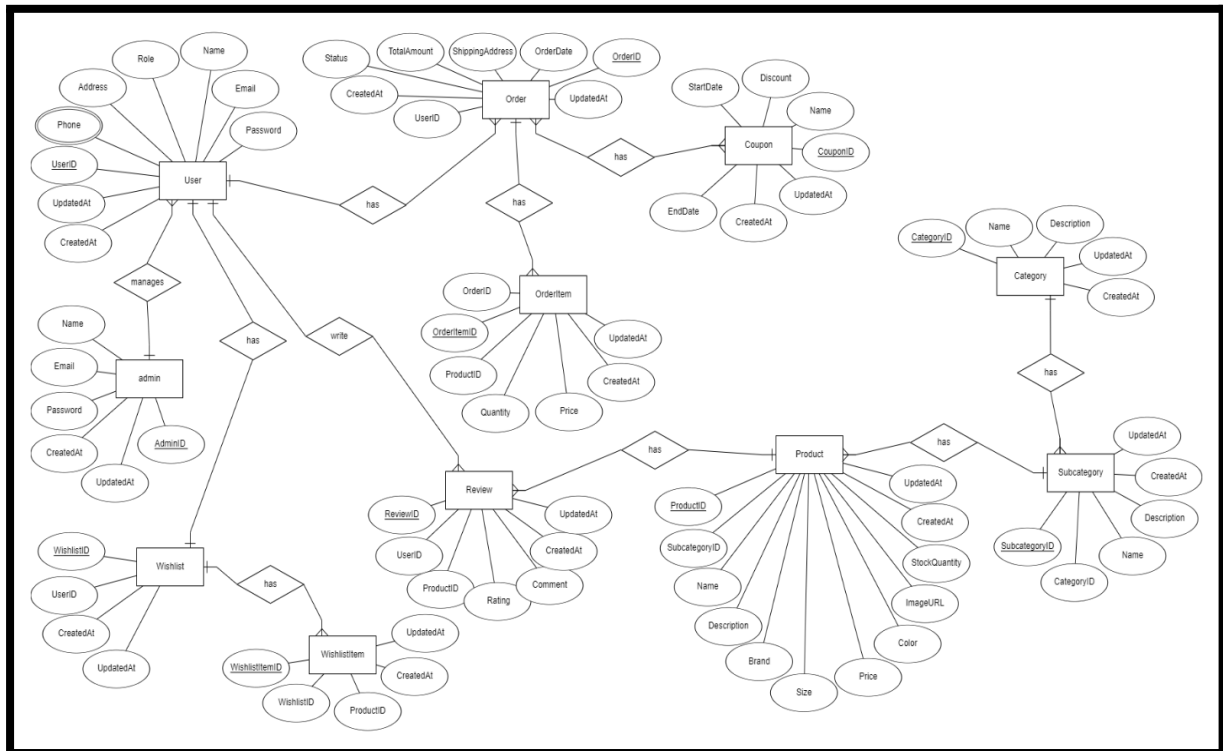


Fig. Entity-Relationship Diagram

## A-2: Data Flow Diagram

A Data Flow Diagram (DFD) represents the flow of data within a system or process, providing insight into the inputs, outputs, and processes involved. Unlike flowcharts, DFDs do not include control flow, loops, or decision rules. They belong to structured-analysis modeling tools and are popular for visualizing major steps and data involved in software-system processes.

### Components of DFD

The Data Flow Diagram has 4 components:

- **Process:** Transforms input into output within a system. Represented by rectangles with rounded corners, ovals, rectangles, or circles. Named with a short sentence, one word, or a phrase to express its essence.
- **Data Flow:** Describes the transfer of information between system parts. Represented by arrows. Named to identify the type of information being transferred.
- **Warehouse:** Storage for data to be used later. Represented by two horizontal lines. Can be data files, folders, optical discs, filing cabinets, etc. When data flows from the warehouse, it is considered reading; when data flows to the warehouse, it is called data entry or update.
- **Terminator:** An external entity that interacts with the system. Example:- Organizations like banks, customer groups, or different departments of the same organization that are not part of the system.

## Symbols and Notations Used in DFDs

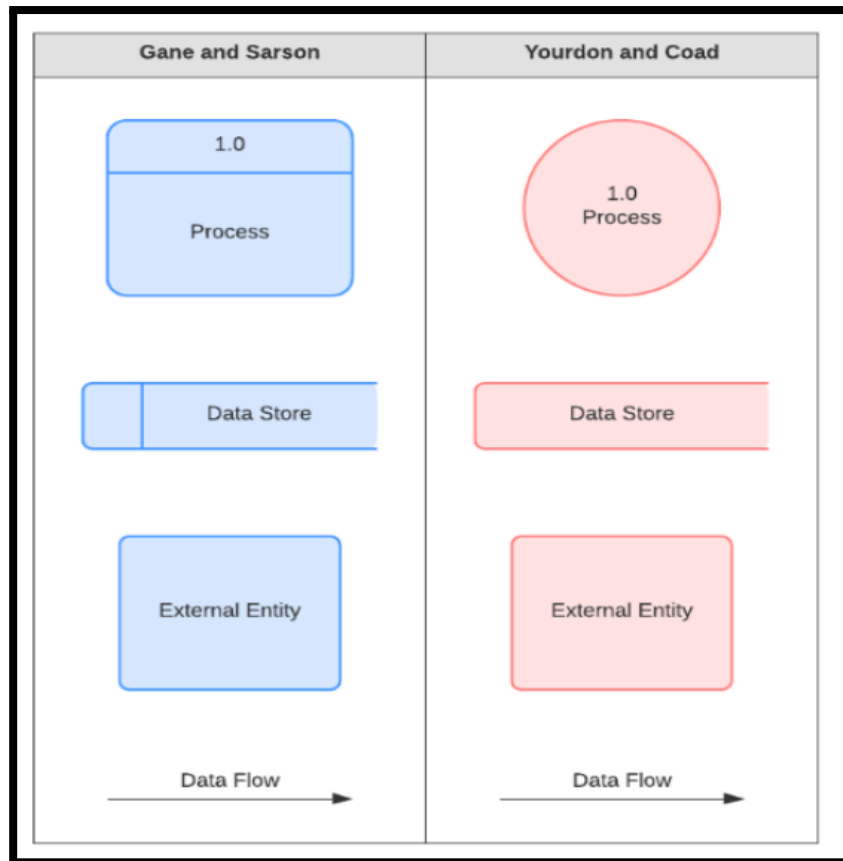
Two common systems of symbols are named after their creators:

- Yourdon and Coad
- Gane and Sarson

One main difference in their symbols is that Yourdon-Coad use circles for processes, while Gane and Sarson use rectangles with rounded corners, sometimes called lozenges. There are other symbol variations in use as well, so the important thing to keep in mind is to be clear and consistent in the shapes and notations you use to communicate and collaborate with others.

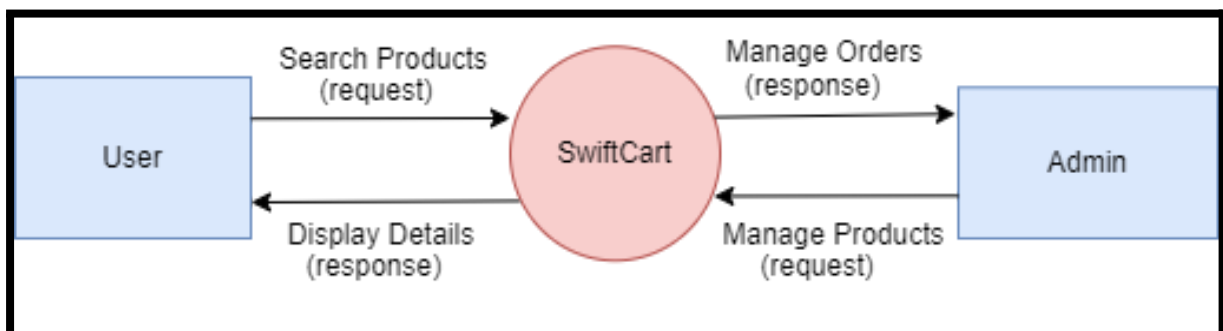
Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

- **External Entity:** An outside system or person that interacts with the system, sending or receiving data. Symbol: A rectangle.
- **Process:** Transforms input data into output. Symbol: A rectangle with rounded corners or a circle.
- **Data Store:** Repositories where data is stored for later use. Symbol: Two parallel horizontal lines.
- **Data Flow:** Routes data between external entities, processes, and data stores. Symbol: An arrow.



## Zero Level Data Flow Diagram

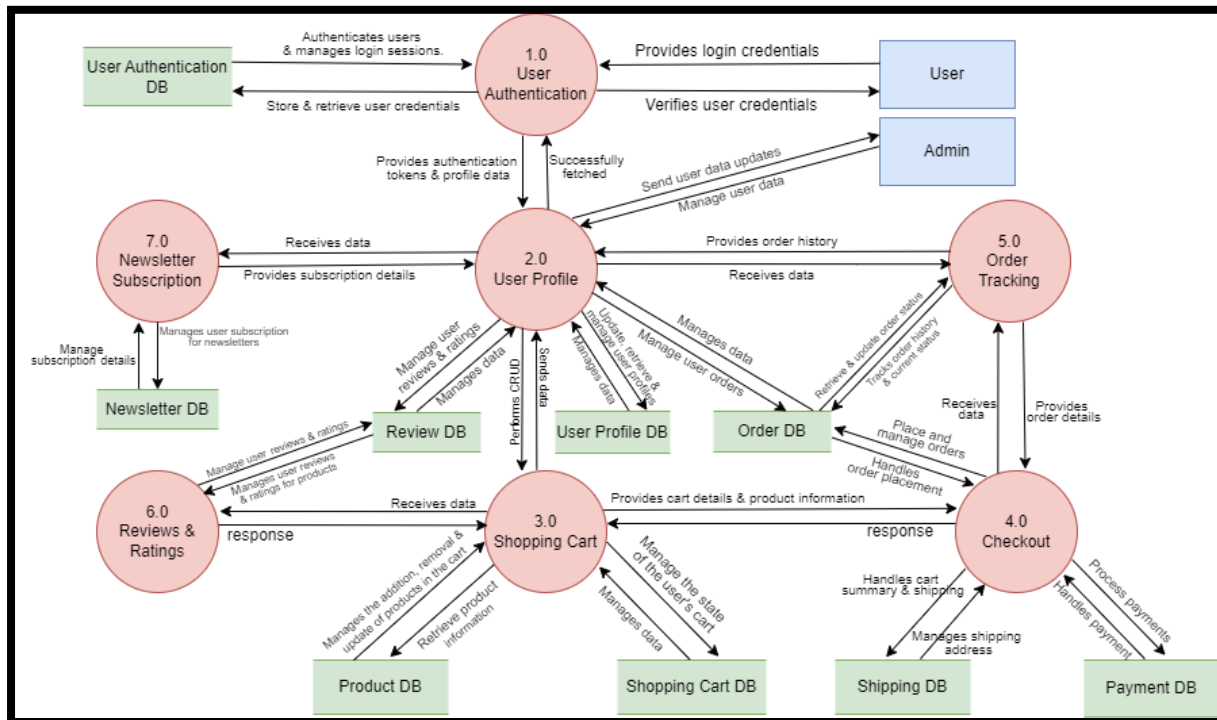
It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.



## Level – 1 Data Flow Diagram I

In level 1 DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes.

A level 1 data flow diagram highlights main functions of a system. Visualize incoming data flow, processes and output data flows.



### A3: UML Use Case Diagram

A use case diagram in the UML is a type of behavioural diagram. A use case diagram is used to represent the dynamic behaviour of a system. It is a visual representation of how a system interacts with its external actors (which can be users or other systems) and the various actions or use cases that the system performs in response to those interactions.

Key elements of a use case diagram include:

- 1. Actors:** Represents an external entity that interacts with the system. Actors can be individuals, other systems, or even hardware components.





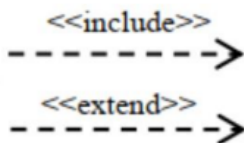
- 2. Use Cases:** Use cases represent specific functionalities or actions that the system can perform. Each use case is a discrete piece of functionality that provides value to one or more actors. Use cases are represented as ovals in the diagram.



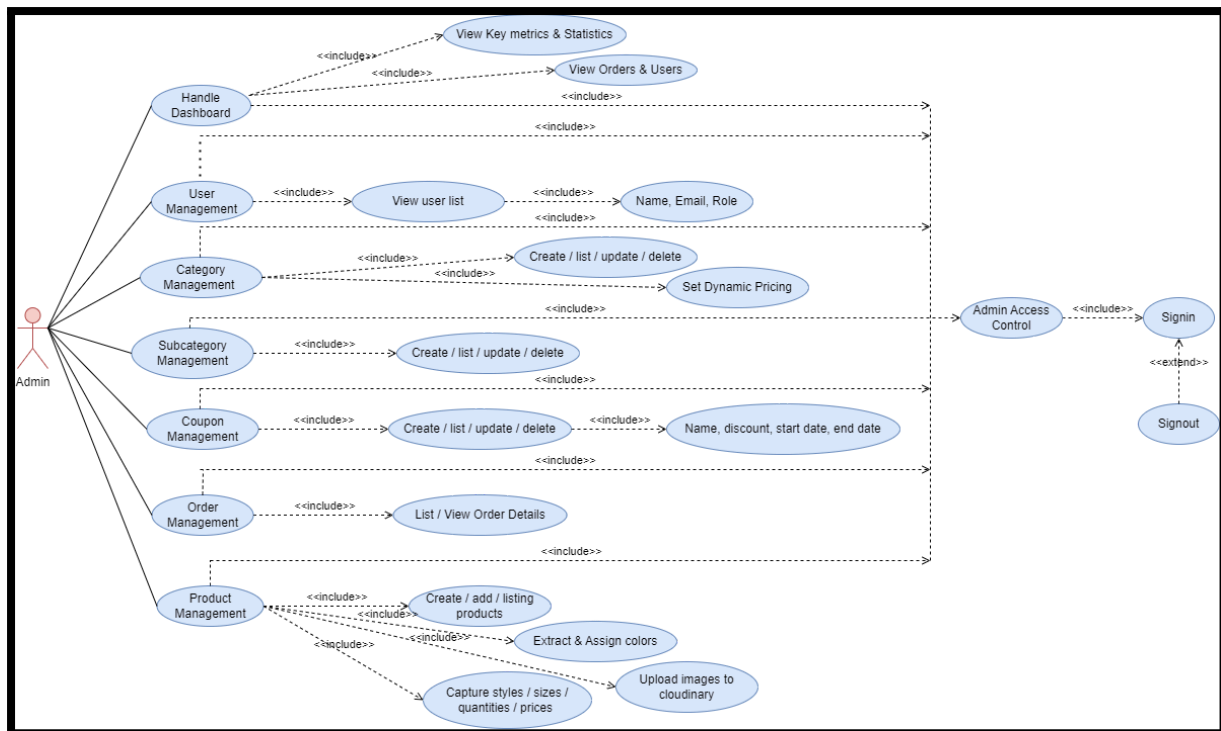
- 3. Associations:** Lines (also known as associations) connect actors to use cases. These lines show which actors are involved in each use case. For example, if you're designing a banking application, an actor might be a "Customer," and use cases could include "Withdraw Money" and "Check Balance." The lines would connect the "Customer" actor to these use cases.



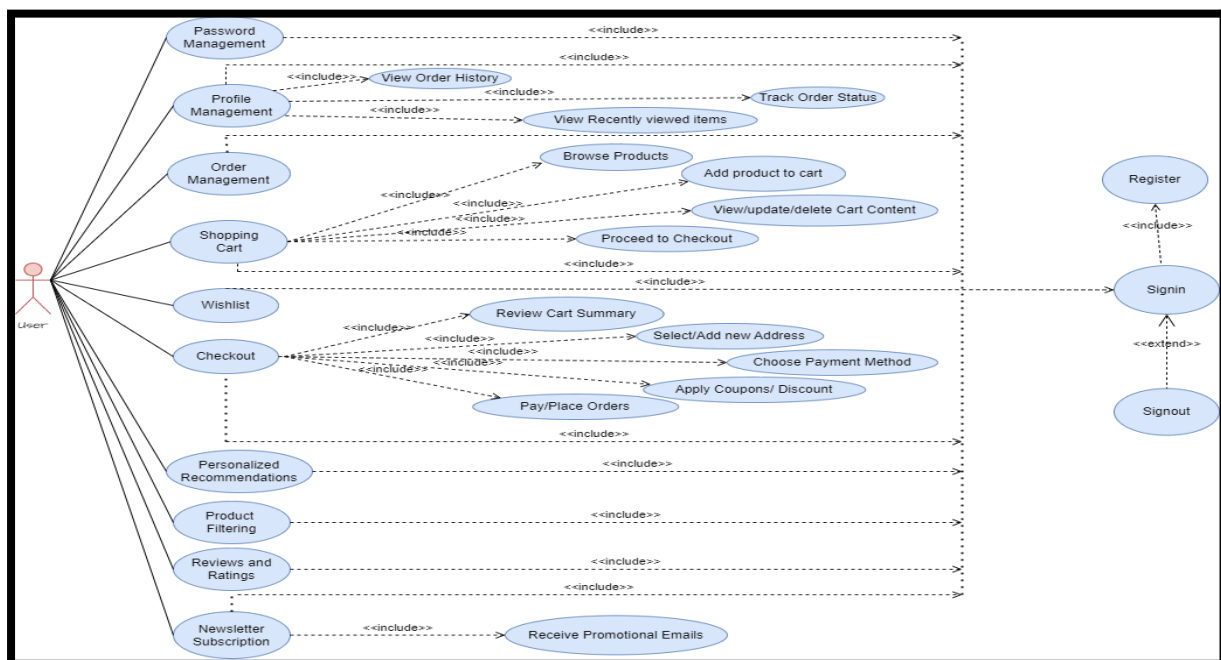
- 4. Include and Extend Relationships:** Sometimes, use cases have relationships with other use cases. "Include" relationships indicate that one use case includes another as part of its behavior. "Extend" relationships indicate that one use case can extend the behavior of another under certain conditions.



### Use Case for Admin Module:



## Use Case for User Module:



## A4: Class Diagram

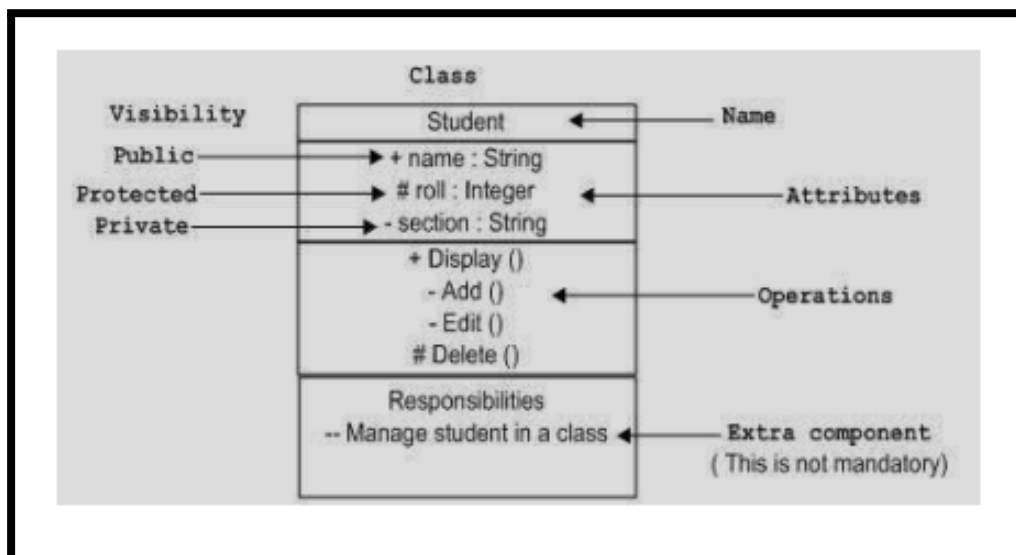
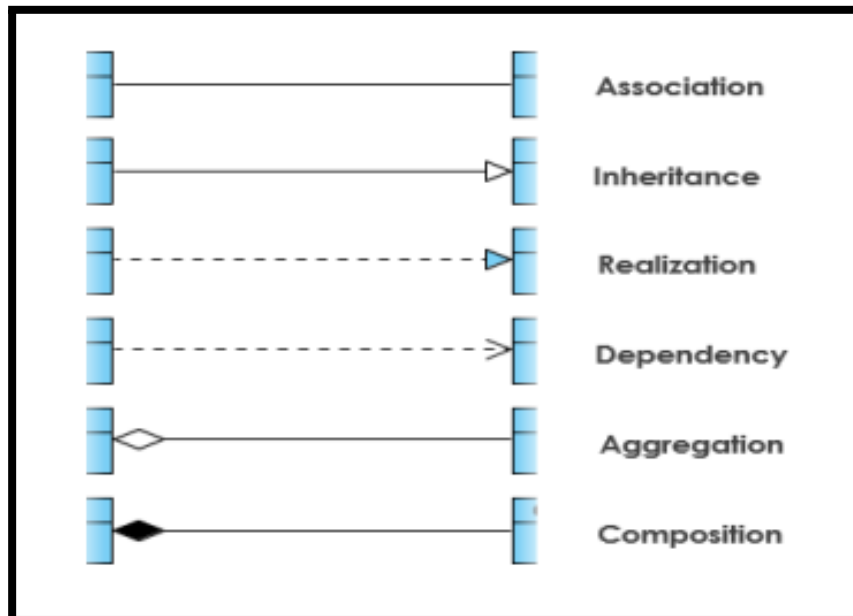
Class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Class diagrams are used to model the static structure of a system, meaning that they show the different types of objects that exist in the system and how they are related to each other. Class diagrams can be used to model any type of system, from simple websites to complex enterprise applications.

Class diagrams are made up of the following elements:

- **Classes:** Classes are the basic building blocks of class diagrams. A class represents a type of object, such as a customer, a product, or an order.
- **Attributes:** Attributes represent the characteristics of a class. For example, a customer class might have attributes such as name, address, and phone number.
- **Operations:** Operations represent the actions that can be performed on a class. For example, a customer class might have operations such as `placeOrder()` and `viewOrderHistory()`.
- **Relationships:** Relationships show how different classes are related to each other. There are four main types of relationships in class diagrams:
  - ❖ **Association:** An association shows that two classes are related to each other in some way. For example, a customer class might have an association with an order class, indicating that a customer can place orders.
  - ❖ **Aggregation:** An aggregation shows that one class is a part of another class. For example, an order class might have an aggregation with an order line class, indicating that an order is made up of order lines.
  - ❖ **Composition:** A composition shows that one class cannot exist without the other class. For example, an order line class might have a composition with an order class, indicating that an order line cannot exist without an order.

- ❖ **Generalization:** A generalization shows that one class is a subclass of another class. For example, a customer class might be a subclass of a person class, indicating that a customer is a type of person.



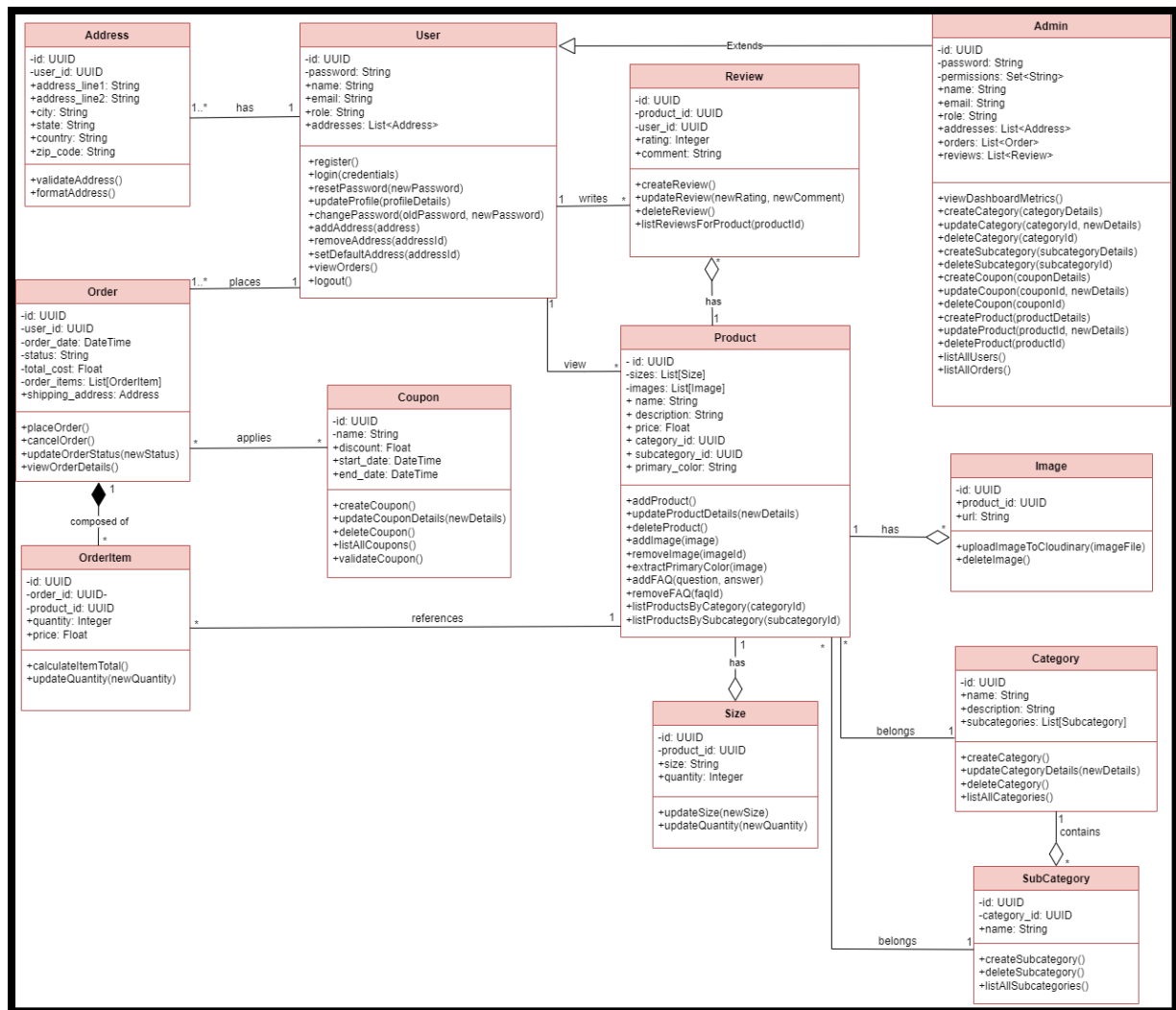


Fig: Class Diagram