*Carnegie Mellon University – NSE TalentSprint*

*Advanced Certificate Program in Software Architecture*

*Batch 02: Feb – Aug 2024*

*Capstone Project – Case Study*

*ModernMaven – StyleMart*

*Project Report*

*1.* *Prashant Sajjan*

*2.* *Johnson V*

*3.* *Sanjeev Chavan*

*4.* *Trupti Jogi*

*5.* *Sharavanakkumar*

*6.* *Manjunathswamy B*

Table of Contents

[Introduction 3](#_Toc173662996)

[Abstract 3](#_Toc173662997)

[Objective 3](#_Toc173662998)

[Background 4](#_Toc173662999)

[Project Context 5](#_Toc173663000)

[Requirements 6](#_Toc173663001)

[Quality Attributes 9](#_Toc173663002)

[Architecture Styles 14](#_Toc173663003)

[Architecture Diagrams 14](#_Toc173663004)

[Architecture Decision Records 14](#_Toc173663005)

[Future Steps 15](#_Toc173663006)

[References and Tools 15](#_Toc173663007)

[Other Applicable Information [Optional] 15](#_Toc173663008)

# Introduction

**ModernMaven** – **StyleMart** is a capstone project aimed at creating an e-commerce platform for ModernMaven, a renowned fashion brand. The project outlines the vision for StyleMart, focusing on high-level requirements and constraints. The primary goal is to provide a seamless shopping experience for fashion-forward professionals worldwide, leveraging ModernMaven’s reputation and the growing trend of online shopping. The platform aims to boost sales by 20% and generate $10 million in revenue within the first year.

[Give a brief introduction to your project]

# Abstract

ModernMaven – StyleMart is an e-commerce platform designed to provide stylish workwear for the modern workforce globally. The project aims to leverage ModernMaven’s strong brand reputation to tap into the growing online shopping market, particularly targeting the upcoming holiday sales.

**Key Aspects:**

**Objective**: Create a visually appealing, user-friendly platform showcasing ModernMaven’s latest collections, enhancing the shopping experience.

Functionality: Includes inventory management, search and filters, shopping cart, user accounts, multilingual support, secure payment, order management, and customer support.

**System Goals**: Focus on visual appeal, performance, scalability, mobile responsiveness, and secure transactions.

**Constraints and Risks:** Launch within six months, compatibility with various devices and browsers, adherence to regional regulations, and potential technical and market challenges.

[Write a short summary in one paragraph of 300 words or less, the major aspects of the entire paper]

# Objective

The primary objective of StyleMart is to create a cutting-edge e-commerce platform that leverages ModernMaven's strong brand reputation to expand its market reach globally. This application aims to offer stylish workwear for modern professionals online and include the drop-shipping capabilities to enhance product variety and customer satisfaction.

# Background

ModernMaven is a well-known fashion brand recognized for its aspirational designs and rave reviews at important fashion events. With a strong reputation in the industry, ModernMaven seeks to expand its market reach by leveraging the growing trend of online shopping and opportunities of upcoming holiday sales season. By launching StyleMart, ModernMaven aims to tap into the global market, catering to customers from different regions, languages, and cultures.

**Business Goal**

● **Expand Market Reach:** Leverage the growing trend of online shopping to reach a global customer base and increase sales revenue by 20% in the first year.

**Business Constraints:**

● Launch Timeline: Launch the e-commerce website within six months to capitalize on the holiday sales. Careful consideration between buy v/s build decision to meet the timeline

● Global Presence: Cater to customers from different regions, languages, and cultures.

**Technical Constraints**

● [Compatibility: Website must be compatible with popular web browsers, devices, and operating systems](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin11,udsdlpconsent,udscstart,cspgrd,&shellsig=4612f2194b92629ea808c14a45999171b5458ce5&setlang=en-US&lightschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C4%7Ca3379a25-33f6-4cd7-b5f1-0a070d2be370).

● Legal and Cultural Compliance: Adhere to regional legal and cultural requirements, including data protection and e-commerce regulations.

● Integration with Third party services (shipping, payment, seller, etc)

**Functional Requirements**

● [Inventory Management: StyleMart shall offer a curated collection of fashionable workwear](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin11,udsdlpconsent,udscstart,cspgrd,&shellsig=4612f2194b92629ea808c14a45999171b5458ce5&setlang=en-US&lightschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C1%7Ca3379a25-33f6-4cd7-b5f1-0a070d2be370).

● Search and Filters: StyleMart shall provide functionality to efficiently search and filter various product options.

● Shopping Cart: StyleMart shall be providing a customer a functionality to be able to add items to their shopping cart, review the cart, and proceed to checkout

● User Accounts: StyleMart shall allow customers to create accounts, manage personal information, track orders, and view purchase history

●  [Multiple Languages:](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin11,udsdlpconsent,udscstart,cspgrd,&shellsig=4612f2194b92629ea808c14a45999171b5458ce5&setlang=en-US&lightschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C2%7Ca3379a25-33f6-4cd7-b5f1-0a070d2be370) The website must support English, French, and Spanish languages to cater to a global customer base.

●  [**Secure Payment:**](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin11,udsdlpconsent,udscstart,cspgrd,&shellsig=4612f2194b92629ea808c14a45999171b5458ce5&setlang=en-US&lightschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C3%7Ca3379a25-33f6-4cd7-b5f1-0a070d2be370) StyleMart must implement secure payment gateways to ensure safe and reliable transactions.

● **Order Management**: ModernMaven should be able to manage and track orders efficiently, including shipping and delivery details

● **Customer Support**: StyleMart should offer customer support features, such as a help center, live chat, and email support, to address customer queries and concerns promptly.

**Key Quality Attributes to be met by the system:**

**●** [**Usability**: Aesthetically pleasing design with intuitive navigation](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,udsedgeshop,channelstable,ntpquery,devtoolsapi,udsinwin11,udsdlpconsent,udscstart,cspgrd,&shellsig=4612f2194b92629ea808c14a45999171b5458ce5&setlang=en-US&lightschemeovr=1&udsps=0&udspp=0#sjevt%7CDiscover.Chat.SydneyClickPageCitation%7Cadpclick%7C5%7Ca3379a25-33f6-4cd7-b5f1-0a070d2be370).

● **Performance**: Fast page loading times and minimal downtime.

● **Scalability**: Handle increasing customer traffic during peak periods.

● **Security**: Secure payment gateways and encryption protocols.

● A**vailability:** The system should be available for 99.99% of the time

# Project Context

**Overview**

StyleMart is an online platform that connects fashion enthusiasts, designers, and shoppers. The system aims to provide a seamless shopping experience while ensuring security, scalability, and usability.

**Stakeholders**

1. **Users/Customers**: Fashion-conscious users who browse and purchase clothing, accessories, and beauty products.
2. **Administrators**: Responsible for managing the platform, handling disputes, and maintaining system health, maintaining inventory, provide customer support to the users.
3. **Designers and Sellers**: Individuals or brands who showcase their collections on StyleMart.

**Requirements**

**Usability Requirement**

* **Objective**: Create an intuitive and user-friendly interface.
* **Measures**:
  + Conduct usability testing with representative users.
  + Evaluate navigation, search functionality, and checkout process.
  + Measure user satisfaction through surveys.

**Security Requirement**

* **Objective**: Safeguard user data and prevent unauthorized access.
* **Measures**:
  + Implement secure authentication (e.g., OAuth, JWT).
  + Encrypt sensitive data (user profiles, payment details).
  + Regularly audit for vulnerabilities.
  + Monitor logs for suspicious activities.

**Performance Requirement**

* **Objective**: Ensure responsive performance even during peak traffic.
* **Measures**:
  + Set response time targets (e.g., page load time < 3 seconds).
  + Monitor server response times and optimize database queries.
  + Load test the system to handle concurrent users.

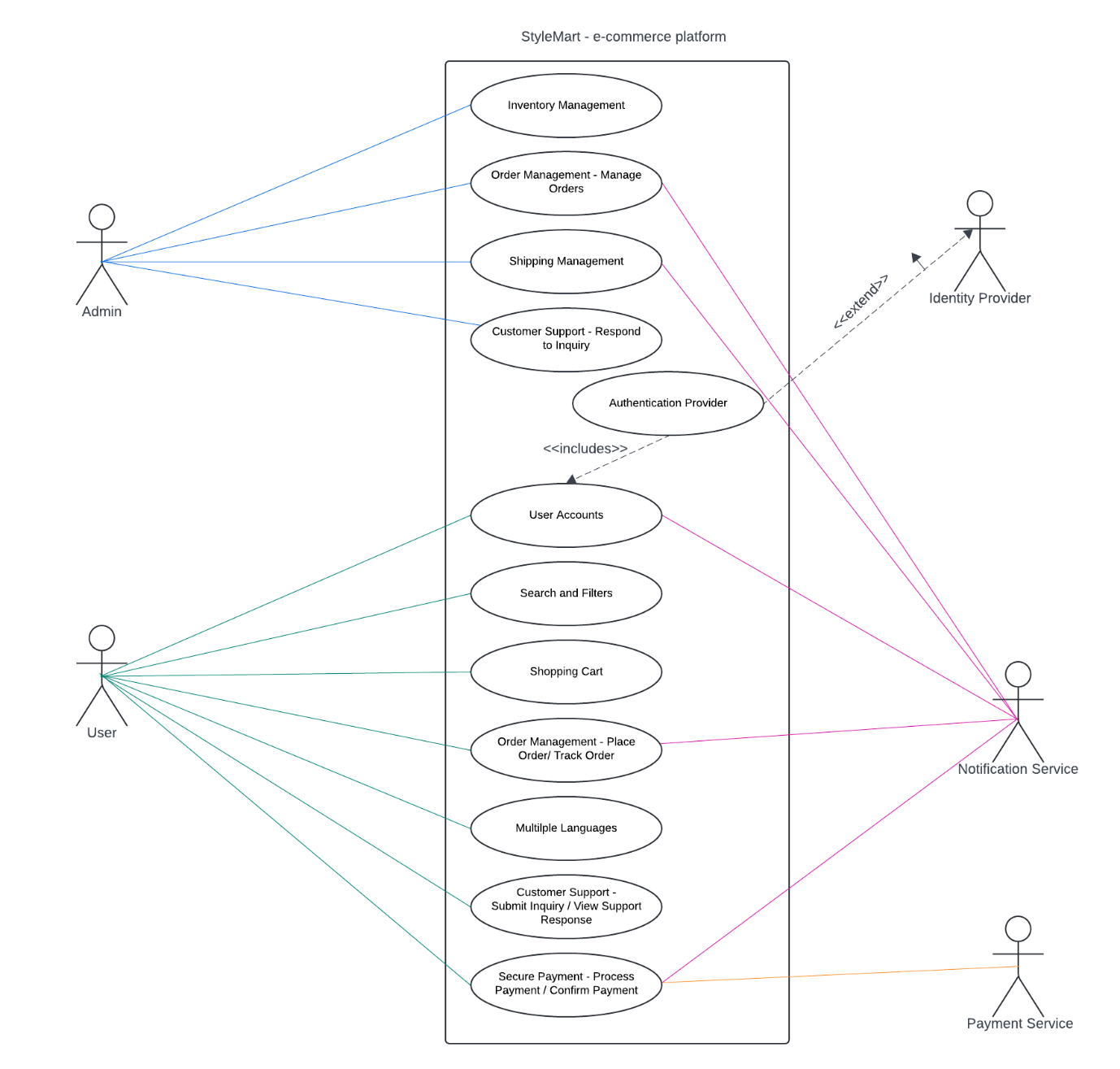
**Scalability Requirement**

* **Objective**: Accommodate growth in users and products.
* **Measures**:
  + Design a horizontally scalable architecture (e.g., microservices).
  + Monitor resource utilization and scale dynamically.
  + Set thresholds for auto-scaling.

**Risks**

1. **Data Breach**: Exposure of customer information.
2. **Downtime**: Loss of revenue during system outages.
3. **Poor Performance**: Frustrated users due to slow page loads.

**Use Case Diagram:**

[https://lucid.app/lucidchart/fdc33558-9025-42cd-82e5-72df236f2ea2/edit?invitationId=inv\_39a70e4f-d002-4c5d-957e-4443e1490d74&page=0\_0#](https://lucid.app/lucidchart/fdc33558-9025-42cd-82e5-72df236f2ea2/edit?invitationId=inv_39a70e4f-d002-4c5d-957e-4443e1490d74&page=0_0) 

[Describe the project context, including how the system relates to its key stakeholders and how it relates to third-party systems that it will need to interact with (e.g., a system context diagram)]

# Requirements

**Summary in Favor of Building a Custom Solution for StyleMart – Trade-off between build v/s design decision**

**Performance**

A custom-built platform can be fine-tuned to meet the specific performance needs of StyleMart. This tailored approach can lead to better performance, especially under the unique traffic patterns and usage scenarios of the business.

**Scalability** This can be designed from the ground up, ensuring the platform grows seamlessly with the business.

**Modifiability**

This provides the ability to customize every aspect of the platform. Pre-built solutions often come with limitations in terms of customization. With a custom build, you have full control over the architecture and features, allowing you to implement unique functionalities that align perfectly with the business needs. This flexibility can be crucial as the business evolves and new requirements emerge.

**Cost Effectiveness**

While the initial development costs for a custom solution can be high, they can be offset by long-term savings. Pre-built solutions typically involve ongoing licensing fees, and additional costs for customizations, integrations, and scaling. A custom-built platform, once developed, only incurs maintenance and infrastructure costs, which can be more predictable and manageable over time.

**Future Limitations**

Pre-built solutions can impose constraints that may hinder future growth and innovation. Custom building eliminates these limitations, providing a platform that can evolve with the business. You can avoid being locked into a vendor’s roadmap and timeline, ensuring that any new features or changes can be implemented according to the business’s priorities.

**Further Customization**

As the business grows, the need for further customization will inevitably arise. A custom-built platform allows for continuous development and enhancement, ensuring that the platform remains aligned with the business goals. Whether it’s integrating new technologies, adding advanced features, or redesigning the user interface, a custom solution offers unmatched flexibility.

**Specific Considerations**

**Search and Filters**: Tailoring search algorithms and UI design to StyleMart's specific needs can significantly enhance user experience, offering personalized search results and advanced filtering options that match the product offerings.

**Shopping Cart**: Building a custom shopping cart allows for unique features that provide a seamless user experience, including advanced promotion handling, custom discount rules, and a highly optimized checkout process.

**User Accounts**: Implementing custom user account management can offer unique interactions and personalized experiences, differentiating the brand from competitors. Custom solutions can better handle specific requirements for user authentication, authorization, and profile management.

**Order Management**: Custom order management systems can be designed to perfectly align with the business processes, providing more efficient and effective handling of order tracking, returns, and inventory management.

**Payment**: Custom payment solutions offer flexibility in choosing and integrating with various payment providers, ensuring secure handling of payment information and compliance with relevant regulations while providing a tailored checkout experience.

**Business Assumptions**

* Target Market: The primary target market is young professionals who are fashion savvy
* Product Mix: Range of products offered will be clothing, accessories, footwear.
* Pricing Strategy: There will be discounted items, premium items and competitive pricing.
* Marketing and Sales Channels: Primary marketing and sales channels will be social media, email, search engine marketing).
* Customer Support: Email or live chat support for initial customer inquiries. Followed by phone support.

**Technical Assumptions**

* Traffic Volume: Estimate the expected website traffic, peak load, and transaction volume. This will determine the required infrastructure capacity and performance optimization strategies.
* Data Volume: Assess the volume of product data, customer data, and order data to be managed. This will impact database design, storage requirements, and data processing capabilities.
* Integration: Identify external systems that need to be integrated (e.g., payment gateways, shipping carriers, marketing platforms). This will determine the integration points and data exchange formats.
* Security Requirements: Define the security standards and compliance requirements (e.g., PCI DSS, GDPR). This will influence security measures, encryption, and access controls.
* Scalability: Determine the expected growth rate of the business and the scalability requirements of the architecture. This will impact system design, infrastructure choices, and deployment strategies.

**Operational Assumptions**

* Development Methodology: Choose the development methodology (e.g., Agile, Waterfall) and the project timeline. This will influence the development process, team structure, and delivery approach.
* Technology Stack: Select the core technology stack (e.g., programming languages, frameworks, databases). This will impact development efficiency, code maintainability, and integration capabilities.
* Cloud vs. On-Premises: Decide whether to host the application in the cloud or on-premises. This will affect infrastructure costs, management overhead, and scalability options.
* Third-Party Services: Determine the use of third-party services (e.g., content delivery network, payment gateway, email service). This will impact cost, performance, and security.
* Monitoring and Logging: Define the monitoring and logging requirements to track system performance, user behavior, and error conditions. This will influence the monitoring infrastructure and data analysis capabilities.

TODO: Need to write requirements yet

[A core set of at least five and no more than ten main functional requirements of the system to come up with a Minimum Viable Product (MVP) of the system. Any assumptions related to functionality must be clearly stated]

# Quality Attributes

**Identified ASRs**

1. **Availability**

When a system or any component (e.g., hardware failure or software crash) having critical features fail under normal operating conditions or peak load, the system or its component switches to a backup component or recovers from the failure within 1 minute resulting into 99.99% availability for those features.

|  |  |
| --- | --- |
| Stimulus | A system or its components dealing with critical features |
| Source of stimuli | Failure |
| Environment | Under normal operating conditions or peak load |
| Artifacts stimulated | Affected system or components |
| Response | The system switches to a backup component or recovers from a failure |
| Response Measure | Overall availability of the system should be 99.99% and recover from failure within 1 minute |

1. **Security**

* Implementation of secure payment systems to protect user data.
* Protection against potential cyberattacks and data breaches.
* Compliance with data protection laws and e-commerce regulations.

When a potential attack, malicious login attempts, fraudulent transactions or any threat happens in the system the intrusion detection system should be able to monitor, detect and prevent threats, safeguard user data according to legal compliance with its strong security effectiveness such as regular policy review, update to the process with compliance alignments, industry standard encryption protocols, log and prevent any unauthorized access or fraud transaction. The time to repair attack should be minimal with no damage to PII, PCI related data.

|  |  |
| --- | --- |
| Stimulus | When a potential attack, malicious login attempts, fraudulent transactions or any threat happens |
| Source of stimuli | Security threats |
| Environment | During normal operations |
| Artifacts stimulated | Intrusion detection system |
| Response | Detect, prevent threats and safeguard user’s data |
| Response Measure | Regular policy review, compliance alignment, industry standard encryption protocols, log and prevent unauthorized access or fraud transaction |

1. **Usability**

* The website should have a visually pleasing and user-friendly interface that reflects the brand.
* The platform should work seamlessly across different devices, including mobile phones and tablets.
* Support for multiple languages (English, French, Spanish) to cater to a global audience.

When a website is used by different users globally across different devices, it should have a visually appealing and user-friendly interface which can support multiple languages such as English, French, Spanish.

|  |  |
| --- | --- |
| Stimulus | A website used by different users across different devices |
| Source of stimuli | Global users and different devices |
| Environment | Under normal operating conditions |
| Artifacts stimulated | The visual interface of the system |
| Response | Multiple language & different device support |
| Response Measure | User friendly interface across multiple devices & language |

1. **Performance**

* The website must load quickly to enhance user experience.
* Efficient search algorithms and indexing to ensure users can quickly find products.
* High availability is crucial, especially during peak periods like holiday sales.

When a user browses the catalog and search for the product or place an order in the system under normal operation conditions, the website response time should be less than 1 second for each activity. Retrieved product should be updated in real time based on available inventory.

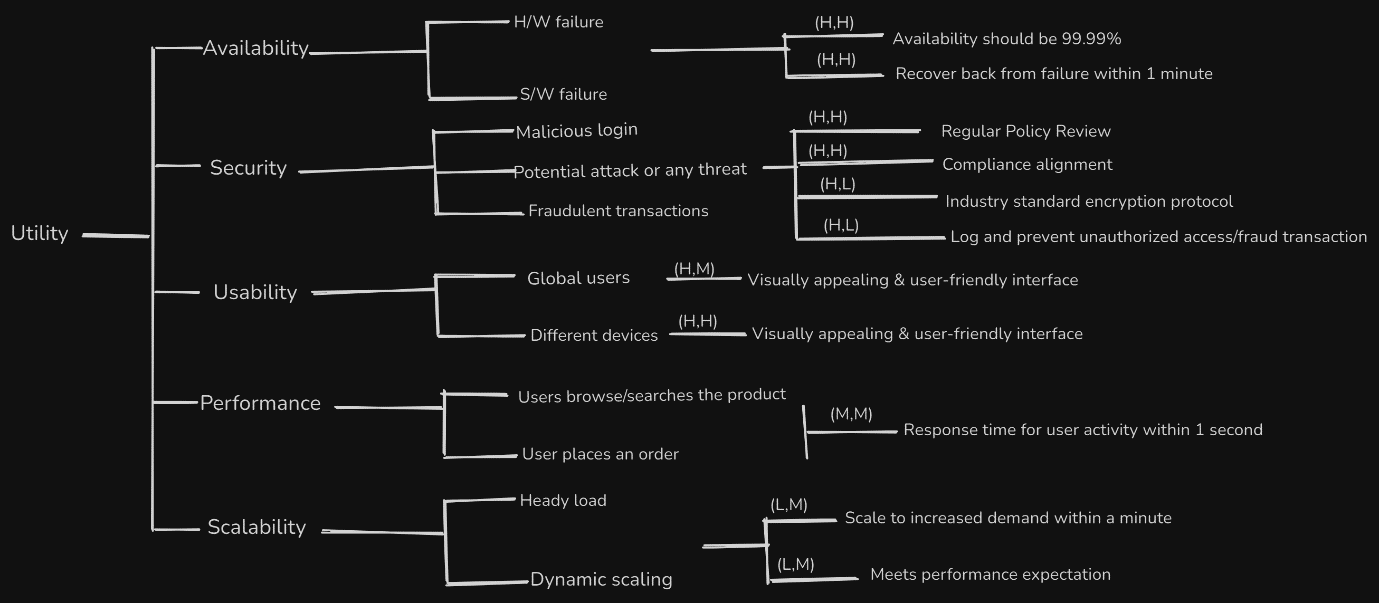
|  |  |
| --- | --- |
| Stimulus | When a user searches the product, or places an order |
| Source of stimuli | User who browses the catalogue or search the product or place an order |
| Environment | Under normal operating conditions |
| Artifacts stimulated | The product databases, UI, order or product services |
| Response | System displaying the relevant product or responding to user activity |
| Response Measure | Ensure response time within 1 second |

1. **Scalability**

* The system should support scaling up for high traffic, especially during peak seasons.
* It must support many concurrent users without degradation in performance.

A sudden surge in user requests as shoppers visit the website during a major sale event such as black Friday, the system component should be able to scale dynamically within a minute without performance degradation under heavy load.

|  |  |
| --- | --- |
| Stimulus | A sudden surge in user requests as shoppers visit the website during a major sale event |
| Source of stimuli | Increased user activity |
| Environment | Under heavy load |
| Artifacts stimulated | System components |
| Response | Scale dynamically without performance degradation |
| Response Measure | Adapts to increased demand within a minute |



Other considerations:

1. **Functionality**

* Effective management of products, including clothing, accessories, and footwear.
* Features for account creation, order tracking, and management.
* Integration of support features like live chat, email, and a help center.

1. **Compliance**

* Compliance with the legal, cultural, and regulatory requirements of different regions, including language and currency support.

**Prioritization:**

**1. Scalability vs Availability**

Critical for Business Growth: As StyleMart aims to cater to a global customer base, the system must handle increasing traffic and user demands, especially during peak times like holiday sales.

Future-Proofing: Ensuring scalability from the outset will allow for smooth expansion and the addition of new features, such as drop-shipping partnerships.

Trade-Offs:

Performance: Achieving scalability might require a more distributed system architecture, potentially complicating the optimization for low-latency and high-speed transactions.

Cost: Scalable infrastructure, such as cloud services with auto-scaling features, can lead to higher operational costs.

Scalability is essential for supporting the platform's growth and handling peak loads, especially during critical sales periods like the holidays. It directly impacts the platform's ability to serve a global customer base effectively.

**2. Security**

Customer Trust: Security is paramount for protecting sensitive customer data, such as payment information and personal details. This is crucial for building and maintaining customer trust.

Legal Compliance: Compliance with data protection regulations, such as GDPR, is mandatory and can have significant legal and financial repercussions if not adequately addressed.

Trade-Offs:

Usability: Enhanced security measures, like two-factor authentication, may add complexity to the user experience, potentially deterring customers.

Performance: Security protocols, like encryption and secure payment gateways, can add overhead and potentially impact system performance.

Security is critical for protecting customer data and ensuring legal compliance, which are non-negotiable aspects of operating an e-commerce platform. It is crucial for maintaining customer trust and avoiding legal repercussions.

**3. Performance vs Usability Vs Scalability**

User Experience: Fast page loading times and responsive interactions are crucial for a positive user experience, which directly impacts conversion rates and customer retention.

Competitive Advantage: In a highly competitive market, superior performance can differentiate StyleMart from other e-commerce platforms.

**Trade-Offs:**

Complexity: Achieving high performance may require complex caching strategies, optimized database queries, and efficient coding practices.

Scalability and Cost: Balancing performance with scalability can be challenging, as optimizing for one may impact the other. Additionally, investing in high-performance infrastructure can be costly.

**Usability:**

Performance directly affects the user experience and can significantly influence customer satisfaction and conversion rates. In the competitive fashion e-commerce market, performance can be a key differentiator.

[Decide the two or three most important quality attribute requirements that will be used as the primary drivers for architecting the system. Elaborate these in a utility tree similar to what was presented in class. Make sure to indicate the importance and difficulty of the tree.]

# Architecture Styles

Call-Return - Microservices Architecture Style

Event-based - publish-subscribe

Add architectural worksheet

[The architectural style(s) that have been used in the project. Provide a rationale for why the particular style(s) was chosen and what qualities you are promoted or hindered as a result.]

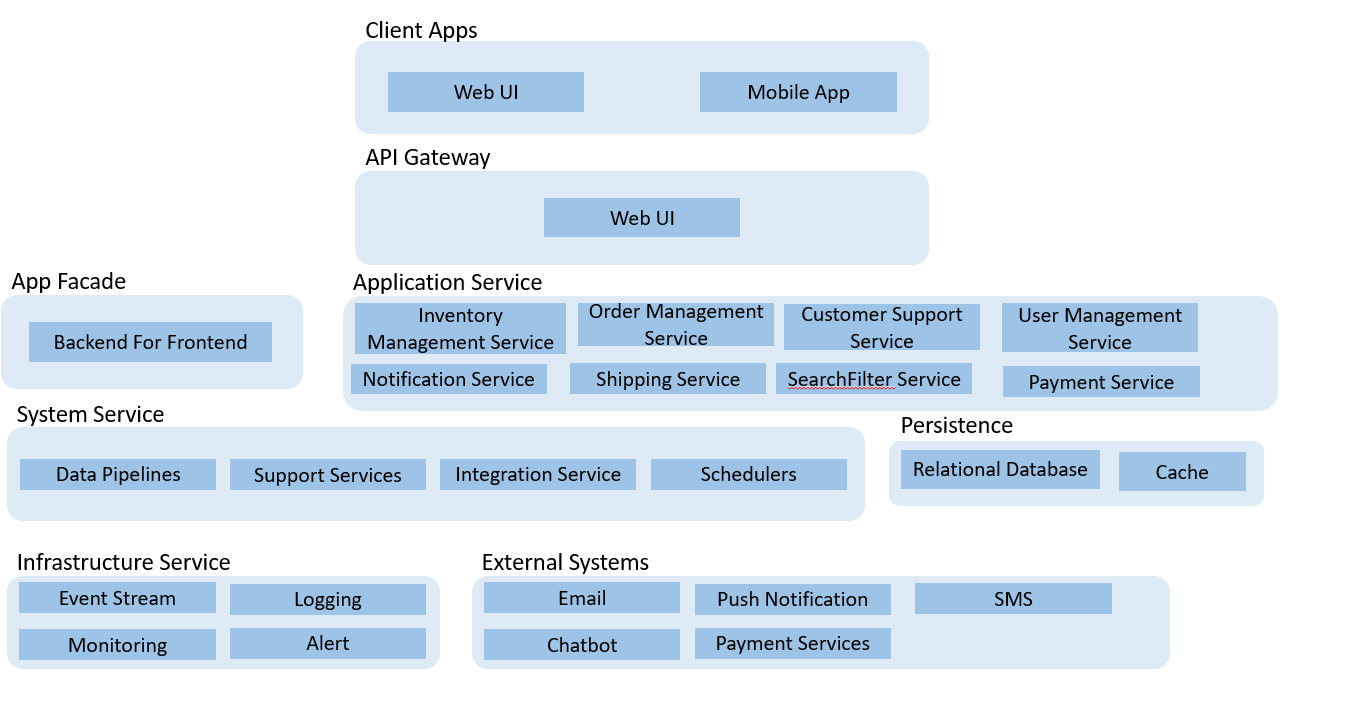
# Architecture Diagrams

[Document your architecture in the form of diagrams using appropriate views. You should at least have a system context diagram and one component and connector diagram. Depending on your quality attributes, a module view and/or deployment view may also be necessary.]

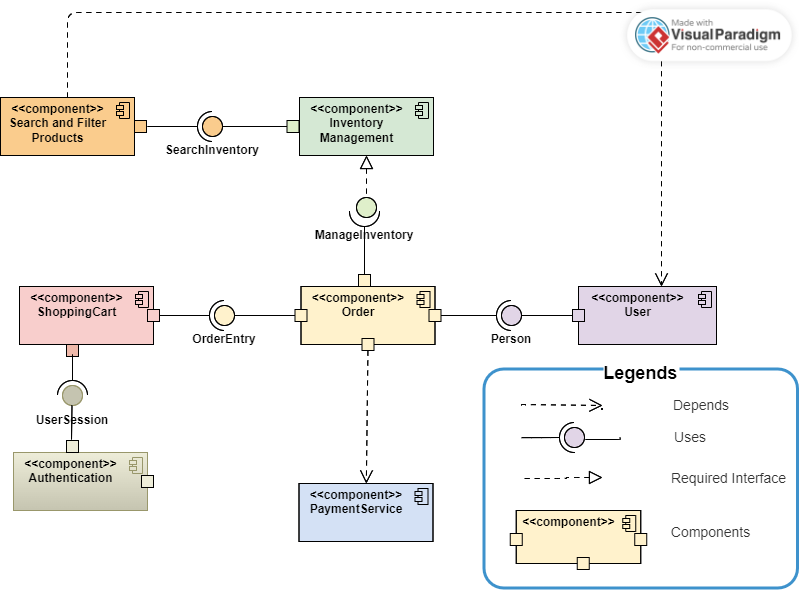
<https://lucid.app/lucidchart/63aa1bdc-1a03-4968-b74e-0ed3fdbbf0cf/edit?invitationId=inv_4a966a98-e542-42fb-ae4a-d8ef99aa7ede&page=0_0>

Module View



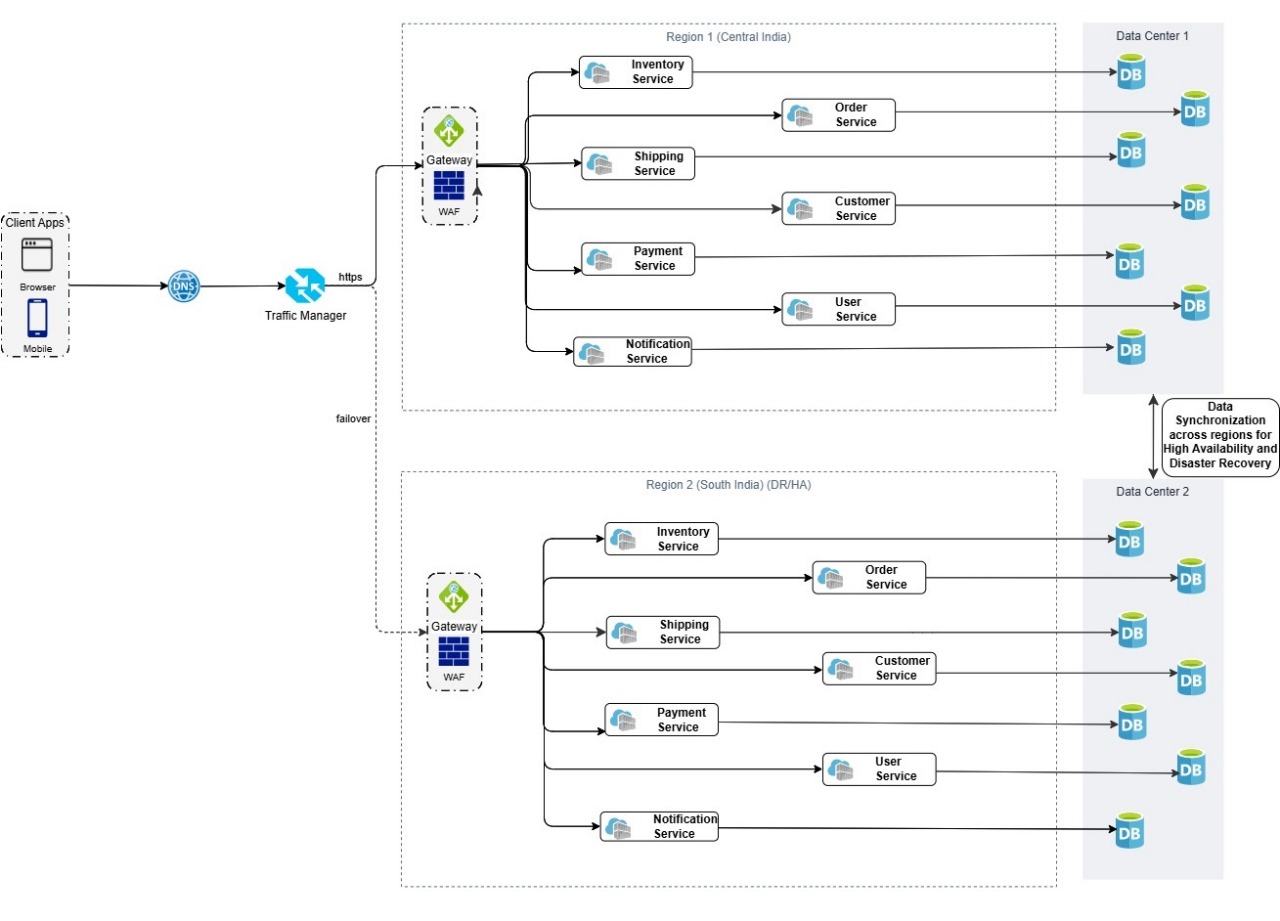
TODO: Below Module View, Shall we remove?

C & C View



Deployment View

TODO : Shall we add Legends in below diagram?



# Architecture Decision Records

[Capture at least two decisions in Architecture Decision Records that address the quality attributes you have identified above. Remember that it is good practice to aim for three consequences. Use the template defined in [architecture-decision-record/locales/en/templates/decision-record-template-by-michael-nygard at main · joelparkerhenderson/architecture-decision-record · GitHub](https://github.com/joelparkerhenderson/architecture-decision-record/tree/main/locales/en/templates/decision-record-template-by-michael-nygard) The ADR should be clear in discussing rationale, alternatives considered, and trade-offs]

# Future Steps

[Describe the next steps for refining the architecture. Be as specific as possible, for example, outline prototypes or experiments that you have carried out to resolve issues, decide between alternatives, and/or reduce risk and uncertainty that you have.]

# References and Tools

1. **Books and Articles:**
   1. Software Architecture in Practice" by Len Bass, Paul Clements, Rick Kazman.
   2. Documenting Software Architectures View and Beyond second edition by Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers
   3. Designing Data-Intensive Applications by Martin Kleppmann.
   4. Fundamentals of Software Architecture by Mark Richards & Neal Ford
   5. Architecting Software Intensive Systems by Anthony J. Lattanze
2. **Documentation and Guidelines:**
   1. PCI DSS (Payment Card Industry Data Security Standard) documentation for payment processing security requirements.
   2. OWASP (Open Web Application Security Project) guidelines for web application security best practices.
   3. ISO standards related to e-commerce, data protection, and internationalization.
3. **Case Studies and Whitepapers:**
   1. Case studies from companies known for successful e-commerce platforms (e.g., Amazon, Shopify) regarding scalability and performance optimization.
   2. Whitepapers from cloud service providers (e.g., AWS, Azure) on best practices for deploying scalable applications.
4. **Tools:**
   1. Draw.io, LucidChart, Excalidraw, Visual Paradigm as diagramming tool
   2. Gliffy – an extension in confluence as a diagramming tool
   3. Confluence, JIRA, Github

[List the sources that you have referred to. Also, mention the tools that you have used.]

# Other Applicable Information [Optional]

[Any other information that you might want to highlight]