SOFTWARE ENGINEERING PROJECT

System Requirement Specification.

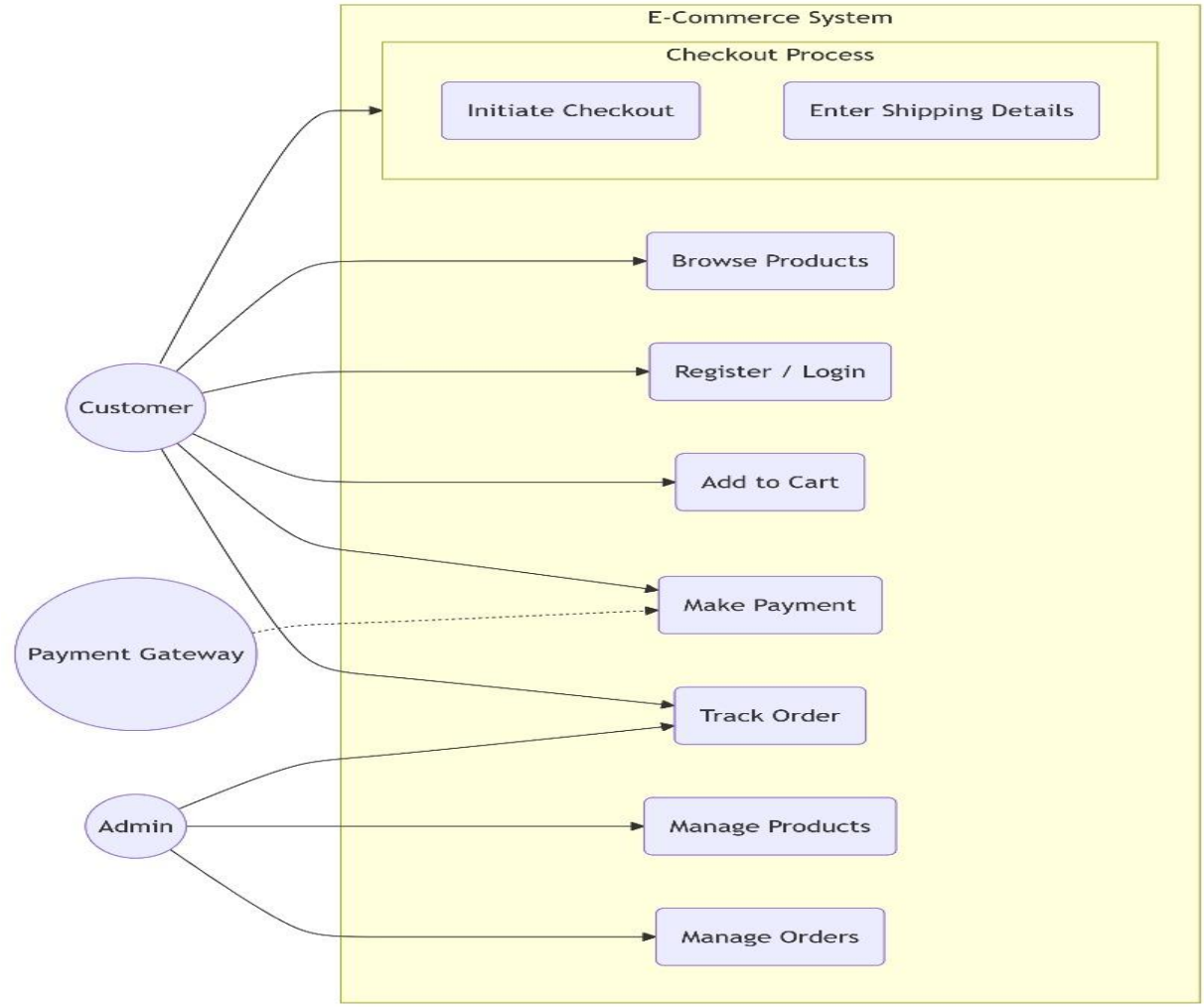
Project: **Baby_Boo Kids Dress & Unisex Underwears E-Commerce Platform.**

Functional Overview & Use-Case Model:

Problem Recap:

Baby_Boo Kids Dress & Unisex Underwears currently operates only through a physical store, limiting its market reach and customer convenience. Parents and shoppers increasingly demand secure, accessible online shopping experiences with flexible payment options. The absence of an e-commerce platform restricts visibility, sales growth, and customer satisfaction. The proposed solution is a modern, responsive, and secure online store with integrated payments, product catalog management, and customer account features to expand reach and build customer loyalty.

Use-Case Diagram:



Use-Case Description:

UC-1: Browse Products

- ***Actor:*** Customer
- ***Trigger:*** User clicks *"Checkout"*
- **Normal Flow:** Customer views product categories, applies filters, selects an item.

UC-2: Checkout & Payment

- ***Actor:*** Customer, Payment Gateway
- ***Trigger:*** User clicks *"Checkout"*
- **Normal Flow:** System verifies cart → prompts delivery & payments details → integrates with payment gateway → confirms transcation → generates order.

UC-3: Manage Products

- ***Actor:*** Admin
- ***Trigger:*** Admin logs into dashboard
- **Normal Flow:** Admin adds, updates, or remove product details (name, price, stoock, images).

User Stories (INVEST Format)

- ***US-1:*** As a customer, I want to search and filter products so that I can easily find items that suit my child's needs.
- ***US-2:*** As a customer, I want to add items to a shopping cart so that I can buy multiple products at once.
- ***US-3:*** As a customer, I want to pay securely online so that I feel confident my money and data are safe.
- ***US-4:*** As an admin, I want to update product stock so that customers only see items that are available.
- ***US-5 (Epic):*** As a customer, I want to create and manage my account so that I can track orders and manage personal information.

- ***Decomposed into:***

1. ***US-5a:*** *Register new account*
2. ***US-5b:*** *LogIn/LogOut*
3. ***US-5c:*** *View/Edit profile*

Preconditions & Postconditions

· ***US-1: Browse Products***

- Precondition: Customer is on website; product catalog exists in database.
- Postcondition: Search/filter results displayed on UI.

· ***US-2: Add to Cart***

- Precondition: User has selected a product; product is in stock.
- Postcondition: Cart updated in database; cart summary displayed.

· ***US-3: Secure Payment***

- Precondition: User has valid items in cart; payment gateway online.
- Postcondition: Transaction recorded, confirmation receipt generated, order status = "Paid".

Story Sizing & Epics

Epic E1: Customer Account Management (US-5)

- Split into US-5a (Register), US-5b (Login/Logout), US-5c (Edit Profile).

Epic E2: Order Management (Admin + Customer tracking)

- Split into US-6 (Place Order), US-7 (Track Order), US-8 (Admin Updates Order Status).

Non-Functional Requirements

ID	Quality Attribute	Requirement	Rationale	Measurement/Test
NFR-SEC-01	Security	All transactions encrypted with SSL/TLS	Protect customer data	Verify SSL installed
NFR-PERF-01	Performance	Page loads ≤ 3 seconds	Better user experience	PageSpeed test
NFR-USE-01	Usability	Mobile-responsive design	Customers shop via phones	Test on multiple devices
NFR-REL-01	Reliability	99.9% uptime guarantee	Ensure store availability	Hosting SLA, uptime monitoring

Glossary & References

- ✧ ***Cart:*** A virtual container for selected products before checkout.
- ✧ ***Checkout:*** The process of confirming order and making payment.
- ✧ ***Payments Gateway:*** *A service to authorize and process online payments seccurely.*
- ✧ ***SSL:*** Secure Sockets Layer, protocol for encrypted connections.
- ✧ **User Story:** A story description of a feature from the end-user perspective.

References:

Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.

Kendall, K. E., & Kendall, J. E. (2019). *Systems Analysis and Design* (9th ed.). Pearson.

Supplementary Diagrams

Data Flow Diagram (DFD)

DFD shows interactions among system components, users & databases

DFD Level 0 – (Context Diagram)

Represents the e-commerce system as a single central process.

External Entities

Customer – (places order, makes payment, provides necessary info)

Admin User – (manages inventory, users)

Payment Gateway(Momo) – (handles transactions & payment authorizations)

E-mail/Message Service – (sends transaction messages, delivery updates)

DFD Level 1

Breaks main system into sub-systems.

User management – handles registration, login, profile updates.

Product Catalogue – handles product search & browsing.

Shopping management – Add/ remove items, update quantities.

Order processing – make, track, and cancel orders.

Payment processing – communicate with payment systems.

Inventory management – update product inventory.

Data Stores

User Databases

Product Databases

Order Database

Payment Database

E-Commerce Application (Crow's Foot Notation):

- **Core Entities:**
  - **User:** user_id (PK), email, password_hash, first_name, last_name
  - **Address:** address_id (PK), user_id (FK to User), street, city, GPS_address
  - **Product:** product_id (PK), name, description, price, stock_quantity
  - **Category:** category_id (PK), name
  - **Order:** order_id (PK), user_id (FK to User), order_date, total_amount, status
  - **Order_Item:** order_item_id (PK), order_id (FK to Order), product_id (FK to Product), quantity, price_at_time
  - **Shopping_Cart_Item:** cart_item_id (PK), user_id (FK to User), product_id (FK to Product), quantity
- **Key Relationships:**
  - **User** *has* one or many **Addresses** (1:M).
  - **Product** *belongs to* one **Category**; **Category** *contains* many **Products** (M:1).
  - **User** *places* one or many **Orders** (1:M).
  - **Order** *contains* one or many **Order_Items** (1:M). (This is a critical relationship).
  - **User** *has* one **Shopping Cart** (composed of multiple **Shopping_Cart_Items**) (1:M).