**Requirements Specification Documentation**

**Online Convenience Store System**

**Swinburne University of Technology | Semester 2 – 2025**

**SWE30003 – Software Architectures and Design**

**Assignment 1 – Group 6**

**Team members**

*Dat Cao - 104497198*

*Nam Nguyen - 104836937*

*Hoang Nguyen - 104527639*

*Thuan Huynh - 105218471*

Table of Contents

[I. Introduction 4](#_Toc208537283)

[II. Project Overview 4](#_Toc208537284)

[1. Domain Vocabulary 4](#_Toc208537285)

[2. Goals and Objectives 5](#_Toc208537286)

[3. Assumptions 6](#_Toc208537287)

[4. Scope 6](#_Toc208537288)

[III. Problem Domain 8](#_Toc208537289)

[1. Pain points 8](#_Toc208537290)

[2. Domain Entities 8](#_Toc208537291)

[3. Actors 8](#_Toc208537292)

[4. List of Tasks 9](#_Toc208537293)

[IV. Functional Requirements and Task Descriptions 10](#_Toc208537294)

[1. Creating customers’ accounts 10](#_Toc208537295)

[2. Browsing store catalogue 11](#_Toc208537296)

[3. Managing shopping cart 12](#_Toc208537297)

[4. Creating invoices 13](#_Toc208537298)

[5. Creating receipts 14](#_Toc208537299)

[6. Handling payments 15](#_Toc208537300)

[7. Managing goods packaging 16](#_Toc208537301)

[8. Shipment 17](#_Toc208537302)

[V. Workflow 18](#_Toc208537303)

[1. Creating customers’ account 18](#_Toc208537304)

[2. Browsing store catalogue 19](#_Toc208537305)

[3. Managing shopping cart 20](#_Toc208537306)

[4. Creating invoices 21](#_Toc208537307)

[5. Creating receipts 22](#_Toc208537308)

[6. Handling payments 23](#_Toc208537309)

[7. Managing goods packaging 24](#_Toc208537310)

[8. Shipment 25](#_Toc208537311)

[VI. Data Model 26](#_Toc208537312)

[1. Domain Model 26](#_Toc208537313)

[2. Entity Descriptions 26](#_Toc208537314)

[VII. Quality Attributes of System 28](#_Toc208537315)

[1. Reliability 28](#_Toc208537316)

[2. Security 28](#_Toc208537317)

[3. Scalability/Performance 29](#_Toc208537318)

[5. Usability/Accessibility 30](#_Toc208537319)

[VIII. Other Requirements 31](#_Toc208537320)

[1. Menu Management 31](#_Toc208537321)

[2. Accessibility Support 31](#_Toc208537322)

[IX. Validation of Requirements 31](#_Toc208537323)

[1. Objectives of Validation of Requirements 31](#_Toc208537324)

[2. Validation Process Overview 31](#_Toc208537325)

[X. Possible Solutions 31](#_Toc208537326)

# Introduction

This Software Requirements Specification (SRS) documents the online convenience store system for a medium-sized local business. The owners are seeking to expand digitally to reach customers from everywhere.

The purpose of this SRS is to outline the requirements of the system. It specifies various items such as user tasks, workflows, etc. while also identifying critical quality attributes, for instance, security, performance, reliability and usability.

# Project Overview

The proposed system is an online convenience store designed to support and expand the operations of Your Local Shop. Up until now, the business has operated with a single store front for many years, only serving local residents. As online shopping grows, the owners seek to adapt to the trend by establishing an online presence.

Swinsoft Consulting has been approached by the owners for assistance in developing the system, this project is seen as a small to medium scaled retail initiative. The system will provide core online store functionality and expand the business digitally, reaching customers across the city and potentially, worldwide.

## **Domain Vocabulary**

**Customer/user:** A person who creates account, browses products, places orders and makes payments through the online store.

**Authorized internal user:** User who have pre-served email or people who have access to the back-end or can manage the stock of the products or can do configuration for system

**Store catalogue:** A structured list of available products displayed to customers.

**Receipt:** Proof of payment issued after successful transactions, linked to invoices.

**Invoice:** A bill automatically generated when an order is placed, stating items and their costs as well as payment details.

**Shopping cart:** a temporary container that holds customers’ selected products before proceeding to checkout.

**Employee/staff:** store members who manage product listings, packaging and order processing in the back-office system.

**System:** the entire online convenience store solution required by the stakeholders.

**Order:** a customer’s request for selected products, which are then process for payment and delivery.

**Payment/payment methods:** choices for customer to choose to securely pay for their selected goods.

**Delivery/delivery options:** the method by which purchased goods are transported and delivered to the customers’ home.

**Delay:** where an order or a part of it is unable to be delivered within the originally estimated timeframe due to stock shortages, courier issues, failed deliveries, etc.

**RPO:** Recovery Point Objective

**RTO:** Recovery Time Objective

## **Goals and Objectives**

**Expand customer reach and visibility**

* Successfully established an online storefront to serve customers across Melbourne and potentially, nationwide.
* Showcase list of products that larger supermarkets usually do not offer, giving the store a competitive side.

**Create a reliable online presence**

* Customers can access the site 24/7 to browse products, place orders, track deliveries and get customer service/support.
* The system is responsive and operate seamlessly across devices, for instance, desktop or mobile.

**Streamline/automate the shopping experience**

* Easy product locating with search and filter category features.
* Ordering process is simplified with shopping cart functions, automated invoice generation and digital receipts.

**Support home delivery and shipment management**

* Customers can select options such as home delivery or click-and-collect. These are included with status updates and basic tracking.
* Staff are opened to manage packaging, shipping preparing and hand them off to postal workers or couriers, are their own shipping.

**Flexible product catalogue management**

* Allow staff/owners to easily add, update and remove products without technical assistance and raise issues.
* Sync online catalogue with the physical store stock to avoid overselling or mismatched stocking.

**Provide business insights**

* Display sales statistics and reporting tools, broken down daily, weekly, monthly and year-to-date timeframes.
* Owners can identify popular items, sales trends and seasonal demands to restock and configure marketing strategies.

## **Assumptions**

**Technical infrastructure**

* The client already has a suitable web domain as well as the access to hosting services.
* Necessary hardware and internet connectivity are available, ready for deployment and maintenance.

**Operational preparation**

* The staff will be thoroughly trained to use the back-office system for product management and order processing.
* Owners might rely on postal workers or third-party couriers that are already operating in the area.

**User access**

* Customers interact through three main ways – desktop, tablet and phones.
* Individual users are supported for now, no bulk purchase or businesses.

**Data management**

* Staff information and customer data, orders and product information will be stored in a secure external database.
* Payment card details will not be stored after transactions due to privacy guidelines. General information such as name, email, address, etc. can be stored.

## **Scope**

**In project’s scope**

* **Customer-facing operations:** browsing, shopping cart, checkout, payment and order tracking.
* **Back-office operations:** product catalogue management, sales reporting and order fulfillment.
* **Financial documents:** automatic generation of invoices and receipts connected along with customers’ transaction.
* **Delivery coordination:** support packaging and shipment tracking.
* **Inside system boundary:**
  + Web-based storefront accessible on all desktop, tablet and phones.
  + Staff/owners secured portal with catalogue management, data analysis and order processing.
  + Database storing customer, staff, product and order information.
* **Outside system boundary:**
  + Third-party payment methods, gateways.
  + External postal/couriers services.
  + Marketing, discounts and small loyalty features.

**Out of project’s scope**

* Bulk purchases, wholesale and business accounts.
* Mobile application development.
* Social media integrations beyond optional product sharing links.
* Too advanced data analytics, an entire loyalty program.

# Problem Domain

## **Pain points**

**Inventory Accuracy:** Risk of oversale when stock cannot be updated in real-time, hence leads to cancellations, refunds, and overloaded pending orders.

**Checkout drop - off and session loss:** Long or fragile flows (timeouts, lost cart) cause abandonment.

**Price inconsistencies:** Cart total and invoice total can be different due to tax applies, rate changes, or special-address surcharges from shipment.

**Packaging and shipment errors:** Mis - picks, split shipments, and address issues create exceptions and extra cost.

**Payment confirmation delays:** Payments lags can create unsuccessful transactions and errors at Invoices and Receipts’ steps.

## **Domain Entities**

* **Customer:** person who placing orders (guest or registered).
* **Account:** login credentials & preferences (addresses, phone number, email).
* **Product:** sellable item with SKU, name, price, tax class.
* **Category:** groups products for browsing.
* **ShoppingCart:** in-progress basket linked to session/account.
* **CartItem:** product, quantity, selected options.
* **Order:** confirmed purchase intent (snapshot of items, prices).
* **Invoice:** legal request for payment tied to an order.
* **Receipt:** proof of payment tied to invoice/transaction.
* **Payment:** transaction record (amount, method, status).
* **Shipment:** delivery package(s) with carrier, tracking.
* **Address:** billing and shipping details.
* **Report:** sales statistics by day/week/month/YTD.

## **Actors**

* **Customer (Guest/Registered):** Browses, manages cart, places orders.
* **Store Manager:** Curates catalogue, views sales stats, oversees operations.
* **Warehouse/Packer:** Picks, packs, hands over to carrier.
* **Accountant/Finance:** Reviews invoices, reconciles payments, issues receipts.
* **Customer Support:** Handles reissues, refunds, address corrections.
* **Delivery Carrier (External):** Ships packages, updates tracking.
* **Payment Gateway (External):** Authorizes/captures payments.
* **System Administrator:** Manages roles, configuration, and integrations.

## **List of Tasks**

* Create customer accounts
* Browse store catalogue
* Manage shopping cart
* Create invoices
* Create receipts
* Handle payments
* Manage goods packaging
* Manage shipment

# Functional Requirements and Task Descriptions

## **Creating customers’ accounts**

|  |  |
| --- | --- |
| **Task – Create customer accounts** | |
| **Purpose** | Create account for customer, personalize customer experience. |
| **Trigger/Precondition** | New user registration, have never log in before. |
| **Frequency** | Variable, relying on online store traffic or marketing campaign. |
| **Critical** | Too many customer creating accounts at the same time in sale campaign |
| **Work Area** | Front-end of the webpage. |
| **Sub-Task** | **Example Solution** |
| 1. Enter email | * The system will send verification if the email is valid or display error notation if the email is invalid |
| 1. Enter personal information   ***Problem:*** User submit wrong formatted input or incomplete data  ***Problem:*** Unsaved form data which results in users unsastifaction and may cause them to abandon the registration process. | * The system will provide real-time invalid notation for wrong formatted data. * Before any submission, the system will explicitly points out all the error of all fields. * The system will use auto-saves form input by using cookies. |
| 1. Enter password   **Problem:** User choose weak password | * The system will list all the requirements to create strong password (Uppercase and special symbols). * The system will create a real-time warning if the password is easy to guess or too short. |
| **Variant** | |
| 1. User register using social media account | * The system will provide option for social media registration. |
| 1. User is one of the organization members or an employee (authorized internal user) | * The system will have a option to put the security code of the store or company email wich is already apporved. |

## **Browsing store catalogue**

|  |  |
| --- | --- |
| **Task – Browsing store catalouge** | |
| **Purpose** | * Explore available products, comparation among products * Identify the interest of users |
| **Trigger/Precondition** | A user go through the store’s website and choosing the catalogue page |
| **Frequency** | High, this task will be the initial tasks for many next tasks like adding products to cart or making payment |
| **Critical** | Large product catalogue (500+ products) |
| **Work Area** | Front-end of the Webpage |
| **Sub-Task** | **Example Solution** |
| 1. Acess the catalogue   **Problem:** The page of catalogue response slowly | * The system will provide the efficient permission for user to access the catalogue * The system will be implemented efficiently and should respond under 200ms |
| 1. Products searching and navigating   **Problem:** Inputs from users sometimes are misspell or wrong typing | * The system will provide a search bar at a really top of the page * The search results will be precisely even some input is misspell (type “samsun” but the result still return “samsung” |
| 1. View product listing   **Problem:** Small image, not enough product information | * The system will list all the available products in order and can specify by user (price, categories, sale performance) * All the products will have 3 main information (name, main 300x300 pixel image, price) |
| **Variant** | |
| 1. User wants to search for product but does not have a particular word to search | * The search will also return related results with the input from user |
| 1. One product in listing is already sold out or out of stock | * The system will display clearly the state “Out of stock” or “Sold out” right on the image of product with a red or noticable color |

## **Managing shopping cart**

|  |  |
| --- | --- |
| **Task – Managing Shopping Cart** | |
| **Purpose** | * Let customers collect their wanted products to checkout. * Provide a clear view of items, quantities, subtotals, taxes/fees (if applicable) and estimated delivery time/costs. * Maintain continuity such as saving info across session/devices for signed in users |
| **Trigger/Precondition** | * Users browse catalogue, view item and add/update/remove items, review before order. |
| **Frequency** | * High, as this the most visited route in shopping journeys. |
| **Critical** | * Stock of some items in cart runs out while customers are still browsing. |
| **Work Area** | * Product page, catalogue list, shopping cart page itself |
| **Sub-Task** | **Example Solution** |
| 1. Add items to cart   ***Problem:*** Ambiguous items (size/flavors/packs) | * **The system will be able add items to cart from the catalogue view.** * Require mandatory option selection before enabling “Add to cart” function. Show concise summary with labels. |
| 1. Change items’ quantity   ***Problem:*** quantity exceeds available stock. | * **Users are able to increase/decrease items without duplicating or starting new cart.** * Real-time validation against stock, cap to max available or show “Only \_\_ available”. |
| 1. Delivery time/cost estimation   ***Problem:*** instant delivery as customers need the good asap. | * **Users can see the estimated delivery time/costs after inputting their address.** * Include standard/express shipping as choices. |
| 1. Remove item(s)   ***Problem:*** customer accidentally remove an item | * **Users can remove unwanted items when change of mine occurs.** * Implement a one-step undo for 10 seconds rather than re-adding from catalogue. |
| **Variant** | |
| 1. Guest users can also add items to cart. | * It is session-bound. If guests signed in, the cart would merge to their account. |
| 1. Multi-device editing | * Latest server-side wins the conflict; notification tell users about the update. |
| 1. Physical store pickup | * Add click-and-collect option, have it also as a delivery option, always free. |

## **Creating invoices**

|  |  |
| --- | --- |
| **Task – Creating Invoices** | |
| **Purpose** | * Produce legal evidence for the sale. |
| **Trigger/Precondition** | * Customer has completed checkout process; payment method has been confirmed. |
| **Frequency** | * For each successful order. |
| **Critical** | * Got bulk orders with many items. * Partial refunds/returns. |
| **Subtasks** | **Example Solution** |
| 1. Retrieve customer’s details | * **Customers must fill out their information if they are non-registered clients (if registered), then provide payment details.** |
| 1. Retrieve order details from shopping cart   ***Problem*:**   * Empty order (session timeout) * Items run out during checkout | * **The systems then would automatically update the order details in this step to proceed checkout.** * Implement session persistence and the auto-save cart data on the system. * Implement real-time inventory check with warning notifications. |
| 1. Calculate subtotal, tax, shipping fees   ***Problem:***   * Error shipping calculator because of special addresses * Tax rate got change immediately | * **Calculate how much the customers need to pay in total, include tax, shipping fees, …** * Multiple shipping provider integration with backup calculators * Currency API |
| 1. Apply for any discounts/promotions   ***Problem*:** Expired discount codes | * **Give the customers discounts if they are eligible.** * Implement a real-time discount update validation during checkout. |
| 1. Generate unique invoice number   ***Problem*:** PDF service down | * **To generate a unique number for each invoice as evidence of the customers’ sales.** * Should have multiple PDF service providers (primary/backup). |
| **Variants** | |
| 1. Invoices including discounts/promotions versus standard full-price billing. | * System applies discounts (e.g., seasonal sale, voucher code) automatically and reflects this in the final invoice total. * If no valid discounts apply, the system will generate a full-price invoice. |
| 1. Business-to-business (B2B) invoices | * Invoices may include company name, ABN, and purchase order number if the customer is a registered business. * Different template may be applied (e.g., “Tax Invoice – Business Customer”). |

## **Creating receipts**

|  |  |
| --- | --- |
| **Task – Creating Receipts** | |
| **Purpose** | * Provide customers with proof of completed payment and ensure accurate record-keeping. |
| **Trigger/Precondition** | * Payment has been successfully processed and verified by the system. * An associated invoice exists in the system. |
| **Frequency** | With every completed customer transaction. |
| **Critical** | * Customer requests immediate proof for warranty, return, or expense claim purposes. * Multiple items with different payment methods need to be recorded. |
| **Subtasks:** | **Example Solution** |
| 1. Link to existing invoices.   ***Problem:*** Invoice not found or mismatched. | * **By matching the invoices’ number to create new receipts.** * System can validate that the invoice ID exists and matches the transaction. |
| 1. Verify payment confirmation.   ***Problem:*** Delay from third-party payment gateway. | * **System confirms payment via transaction ID.** * **Accepts only captured statuses (not pending/authorized); verify amount, currency, invoice ref, timestamp, and idempotency key to avoid duplicate receipts.** * System checks payment status and ensures funds are received. |
| 1. Deliver receipt to customer.   ***Problem:*** Customers have not registered. | * **System generates a physical receipt (including unique Receipt ID, Invoice ID, itemized payments, masked card digits, GST, business details, date/time), then stores its e-version in the customer’s account.** * Save customers’ receipts via their phone number and name, or the receipts can be saved as for guests. |
| **Variants** | |
| 1. Refund/return receipts | * System will generate a “Refund/return Receipt” (negative amount) that references the original Receipt ID and Invoice ID. |
| 1. Reissued receipts | * System will duplicate a copy of the previous receipts, marked as “Re-printed receipt”, no new transaction ID, Invoice ID; reissue is logged (who/when/why). |

## **Handling payments**

|  |  |
| --- | --- |
| **Task – Handling Payments** | |
| **Purpose** | Enable customers to complete purchase of goods after submitting financial transactions. |
| **Trigger/Precondition** | Customers have finalized their selection and proceed to check out, where they select payment method. |
| **Frequency** | High occurrence, for every successful order placed through the online store. |
| **Critical** | Payment went through, but the screen freeze and customer failed to see the confirmation page, email. |
| **Work Area** | This only operatable only after customers enter their billing address and payment method. |
| **Sub-Task** | **Example Solution** |
| 1. Provide payment information   ***Problem:*** incorrect personal details (e.g. invalid address or card number) | * **Customers provide their personal information to help contacting them and delivering the goods.** * Implement immediate client-side validation for input fields such as format, length checks. Provide concise feedback. |
| 1. Different payment method   ***Problem:*** limited payment methods, causing customers to abandon their cart | * **Offers different payment methods corresponding to different customers’ preferences.** * Integrate a variety of widely-used payment gateways and services – broader range. |
| 1. Payment confirmation   ***Problem:*** failed to see a timely-screened confirmation or an online receipt, having uncertainties. | * **Process payments successfully and send the order to shipping department.** * Add layer showing payment successful or failed. Send transaction record through email customers. |
| **Variant** | |
| 1. Repeating customers | * Registered customers can securely save their payment preferences. Provide an option to select previously saved method. |
| 3a. Partial/mixed payments | * Include third party services that offer that type of payment (e.g. afterpay, ZIP, PayPal) |

## **Managing goods packaging**

|  |  |
| --- | --- |
| **Task – Managing goods packaging** | |
| **Purpose** | Software control of carrier selection, label generation, tracking orchestration, and delivery lifecycle; customer notifications and order status sync. |
| **Trigger/Precondition** | * Packaging publishes PackageReady(packageId, orderId) event. * Validated customer address is present (or pickup selected). |
| **Frequency** | Per package (an order can yield N packages). |
| **Critical** | * Carrier API outages, rate limits, or webhook failures. * Address normalization errors leading to failed deliveries. |
| **Work Area** | Backend: Shipping Service, Carrier Adapter(s), Webhook Processor, Notification Service, Customer Portal. |
| **Sub-Task** | **Example Solution** |
| 1. Carrier rating & selection   ***Problem:*** Suboptimal cost/time, single-point failure. | * Multi-carrier adapters * Rule-based selection (service level, weight/size, region, cutoff time) * Health-check and circuit-breaker fallback. |
| 1. Label generation & manifest   ***Problem:*** Duplicate labels/charges on retries. | * Two- Idempotent label requests keyed by packageId * Store PDF/PNG + metadata * End-of-day manifest creation per carrier |
| **Variant** | |
| 1a. Multi-parcel orders | * Multiple package IDs under one order ID. |
| 2a. Oversize/perishable items | * Alternate templates or skip consolidation. |

## **Shipment**

|  |  |
| --- | --- |
| **Task – Shipment** | |
| **Purpose** | * Orchestrate the software workflow that prepares confirmed orders for dispatch (generate packaging jobs, rules-driven packing instructions, labels, and state transitions) * Ensure data consistency between Order, Inventory, Packaging, and Shipping components. |
| **Trigger/Precondition** | Order status = Paid/Confirmed and inventory reserved; checkout completed. |
| **Frequency** | Once per order; multi-parcel orders may create multiple packaging jobs. |
| **Critical** | Duplicate/partial packaging jobs from retries or timeouts |
| **Work Area** | Backend services: Packaging Service, Order Service, Inventory Service, Event Bus |
| **Sub-Task** | **Example Solution** |
| 1. Determine packaging rules   ***Problem:*** Wrong materials/splits for fragile/oversize items. | * Rules engine on product metadata (fragile, weight, dimensions, temperature) → select template (box type, cushioning) and compute parcel splits (bin-packing heuristic); * Allow Ops override with audit. |
| 1. Finalize & publish   ***Problem:*** State updated but event not emitted (or vice versa). | * Two-phase commit via outbox; publish PackageReady to Shipping Service * Retries with dedup * Dead-letter queue (DLQ) for failures. |
| **Variant** | |
| 1a. Standard vs Express | * Selection rules + distinct SLAs |
| 2a. International | * Customs data generation and HS codes. |

# Workflow

## **Creating customers’ accounts**

A diagram of a flowchart

AI-generated content may be incorrect.

Figure - workflow of customer account creation

## **Browsing store catalogue**

A diagram of a product

AI-generated content may be incorrect.

Figure - workflow of when customers browse store catalogue

## **Managing shopping cart**

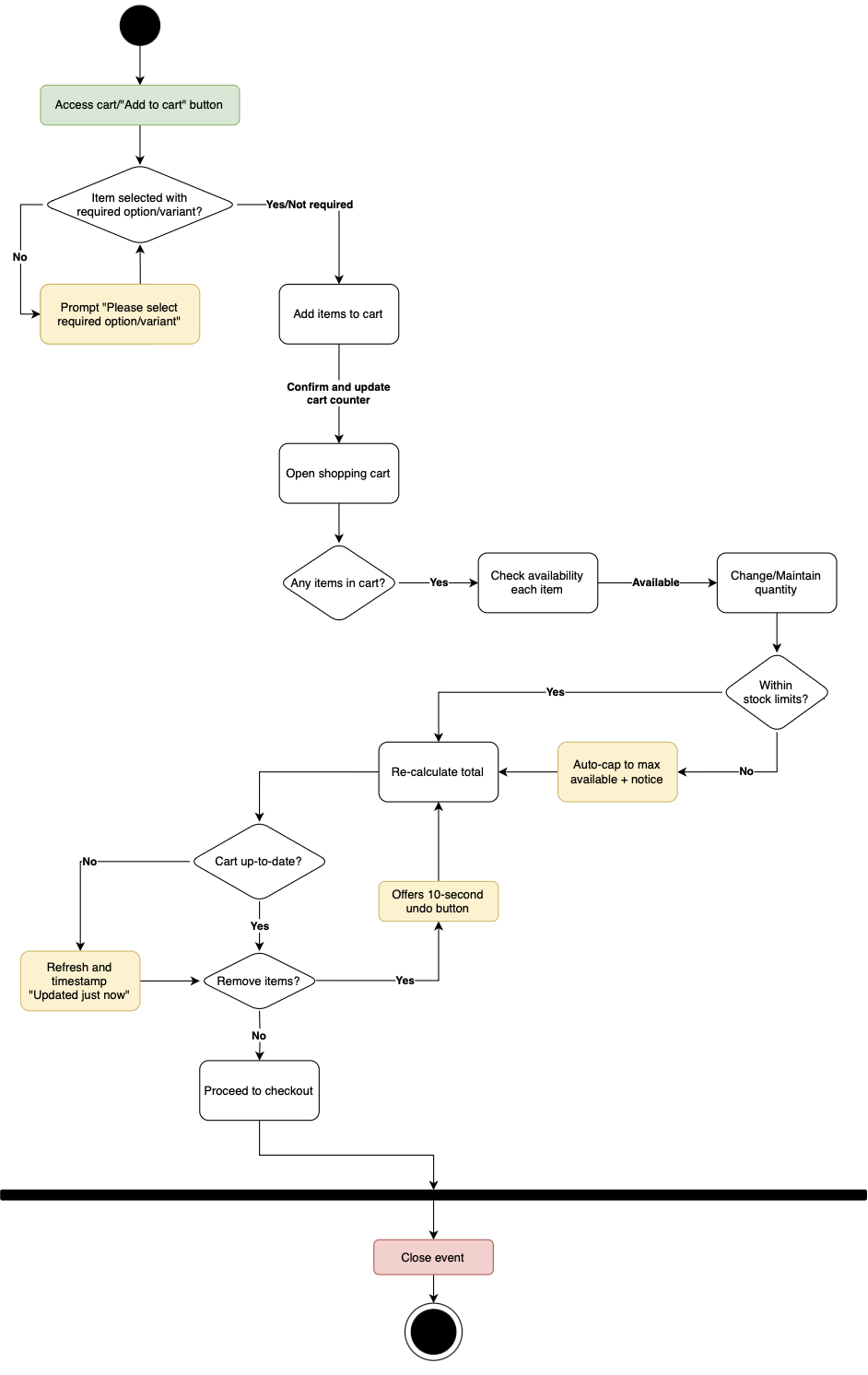


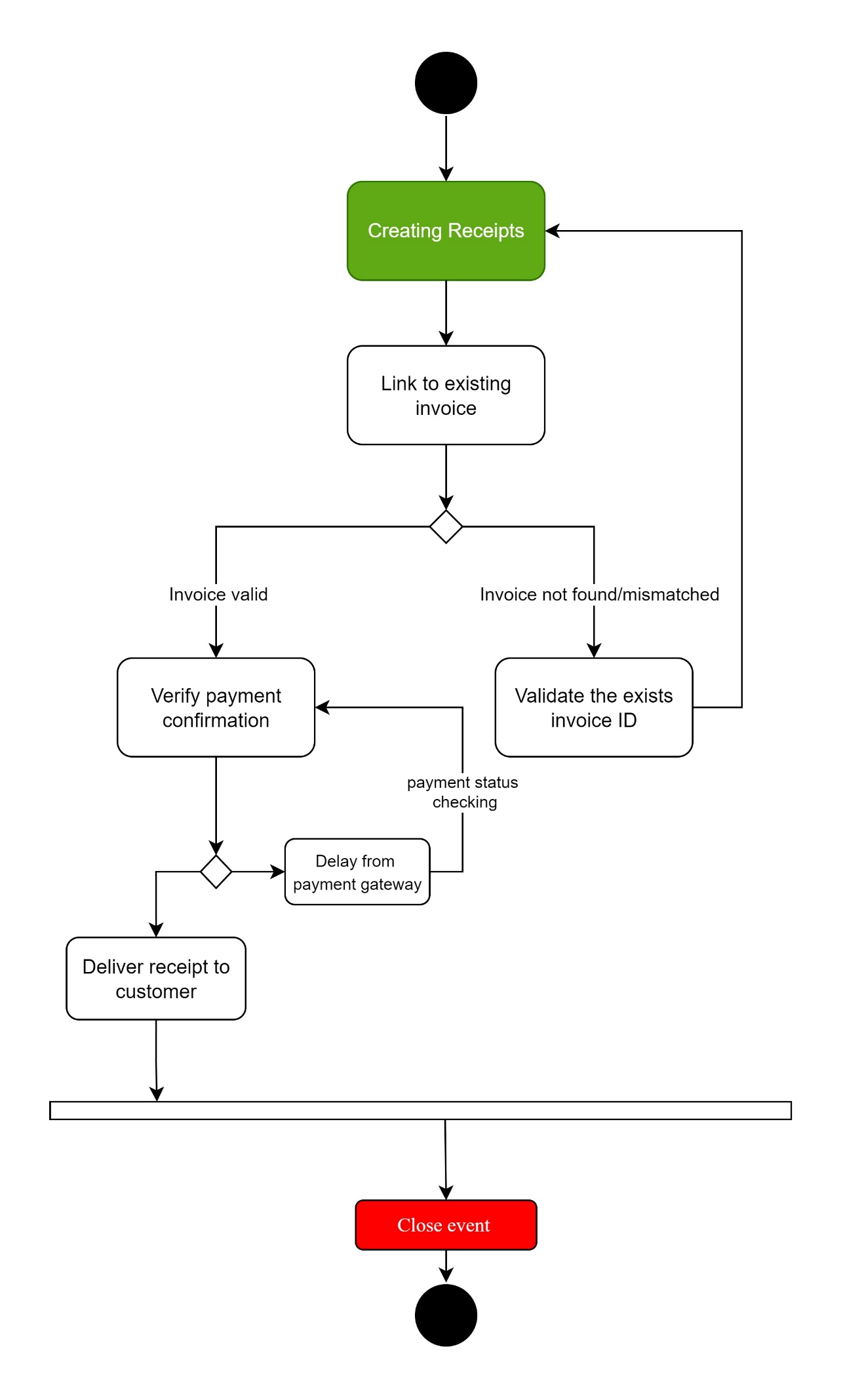
Figure - workflow displaying shopping cart workflow

## **A diagram of a diagram AI-generated content may be incorrect.Creating invoices**

Figure 4 - workflow displaying the task creating invoices

## **Creating receipts**

Figure 5 - workflow displaying the task creating receipts



## **Handling payments**

**A diagram of a flowchart

AI-generated content may be incorrect.**

Figure - workflow of payment handling tasks

## **Managing goods packaging**

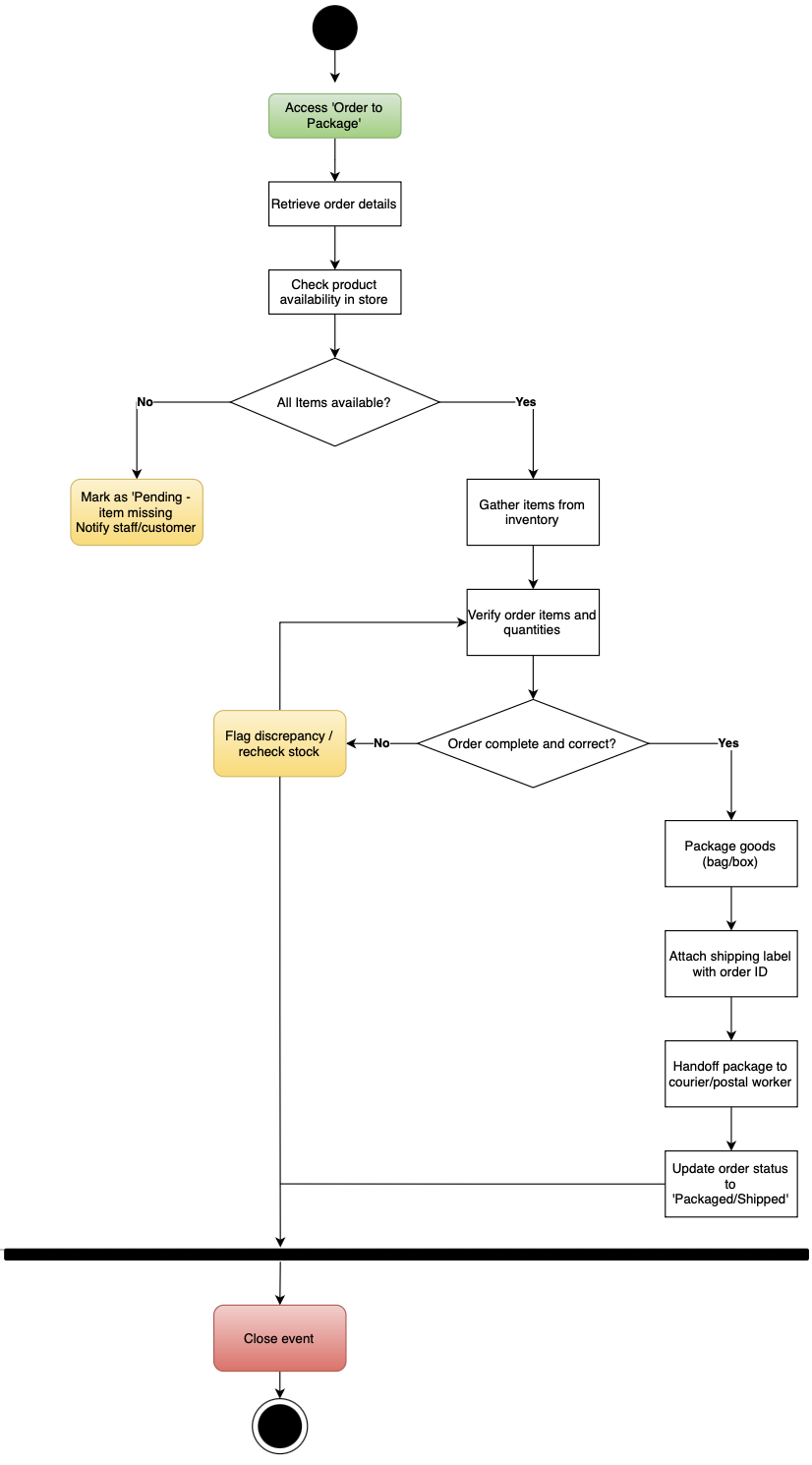
****

Figure - workflow of managing goods packaging

## **Shipment**

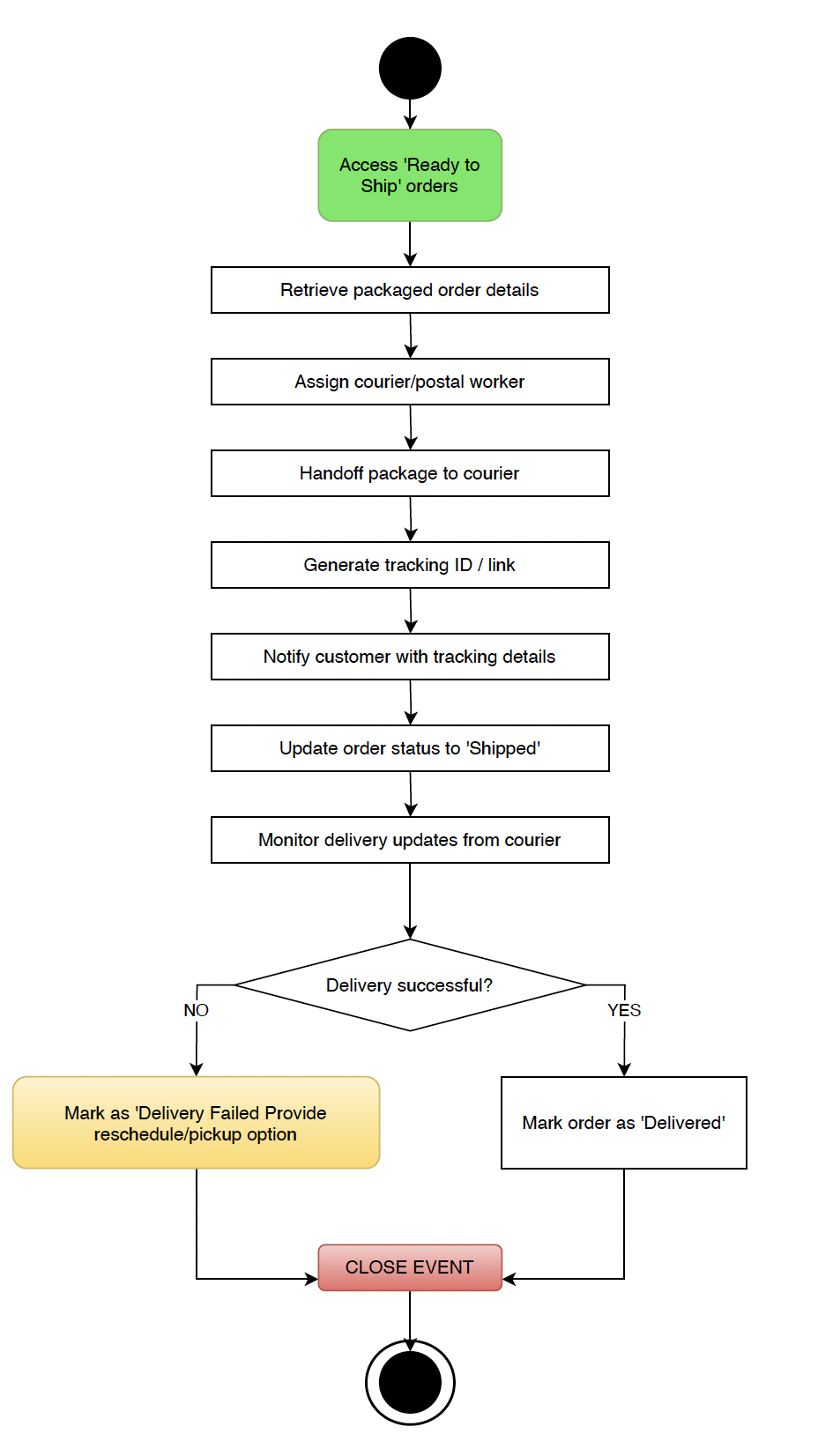


Figure - workflow illustrating shipment procedure

# Data Model

## **Domain Model**

A diagram of a company

AI-generated content may be incorrect.

## **Entity Descriptions**

* **Customer:** The customer is people or user who will potentially buying products from Your Local Shop, they can access the webpage and creating their own account to purchase products online from Your Local Shop.
* **Account:**  Accounts are created by customers who want to buy products from the store, one customer can have multiple accounts
* **Shopping Cart:** A temporary cart that has ability to store all the products that customer want to buy when browsing the catalogue, customer can change the number of the products, delete/ update products and check out the cart
* **Order:** Offical purchase request from customer to the system after considering all their products and finishing the payment process. It has all the customers’s details
* **OrderItem:** This will contain the quantity of the products in a order
* **Shipment:** This department will have the responsibility for packaging and deliver the order to customer
* **Staff:** The employee of Your Local Shop, they have special access to manage the system, they can view the receipt, process the orders or assist the customer
* **Invoice:** This will be sent after getting the details of the user, invoice will include all the number, information and price of the products, including tax and the sum of how much customer need to pay
* **Payment:** This will be created when the customers make payment, has connection with invoice and receipt
* **Receipt:** This will be sent to customer after they finish payment process successfully. It will include information about the products, time and method of payment.
* **Product:** the items that are sold by Your Local Shop, they have different states like Available/Sold out/Out of stock

# Quality Attributes of System

## **Reliability**

**Goal** – The online store needs to be available consistently during normal and peak hours. The system is required to be proficient in tolerating failures (e.g. courier delays, gateway hiccups) without losing orders.

**Avoid** – Critical situations such as “Payment System Unavailable” would halt operations and prevent orders from coming through, hence no funds secured. Aside from that, shopping cart and product listing crashes as well as payment failures would lead to lost sales and reputation damages.

**Key scenarios**

* **Uptime:** given a calendar month, if the availability is measured for the storefront, uptime should be around 99.9% or higher; as for the staff portal, uptime could be lower with 99.5% if not higher.
* **Order durability:** as an order is submitted, if transferring error occurs after a payment authorization then the system will finalize that order; flawlessly rolls back and must not double charges. Order duplication to be under 0.01%.
* **Graceful degradation:** if for instance, a payment gateway shortage happens when customer pays, then the system surfaces a friendly error while leaving the shopping cart intact, no items lost.
* **Backups/Disaster recoveries:** if a data failure, server crash or database corruption occurs, the store’s data should not be lost forever. RPO and RTO are metrics to recover. RPO with 15 minutes or less would have the store lose transactions in that timeframe. RTO with 4 hours or less is the timeframe the store needed to recovery and become available again.

## **Security**

**Goal –** Make security as the top priority of the system. The backend which has the responsibility of storing customer data and process query needs to be designed in principles that can protect the data, payment information from fraudulent activities.

**Avoid** – Critical situation for this attribute will be “Unauthorized access” when the system can not detect strange access from unauthorized users which can lead to data breach/leak or stealing 🡪 violate the legit of the store, lost customers.

**Key scenarios**

* The customer client should be store in compliance with Australian law.
* The system should provide warning when there are more than 5 times. of login request from 1 IP address
* The sensitive information of users like card number, phone number should not be store in plain text, they need to be embedded or encryption by AES-256 algorithms.
* Identify the authorized internal users who have authorization to access the restricted services of the system, which normal users do not have access to such as the IT department or employee working with the store can access to modify the function of the webpage or debug errors while the sale assistances can only view the receipt, change the stock count of the products.

## **Scalability/Performance**

**Goal** – Server responses should response snappy to users’ interactions, for instance, loading product pages, processing payments, etc. Scales with demands (e.g. promos, holidays). Efficient processing of payments and order confirmations is vital for smooth experiences.

**Avoid** – Slow response will frustrate customers and lead to shopping cart abandonment, or even worse, losing customers plus negative reviews. Online shopping environment is where speed is prioritized.

**Key scenarios**

* **Catalogue search response time:** searching, browsing listings should be quick and efficient, despite the number of online users. For instance, with 100 sessions, 95% of the catalogue or search loads should be finished in under 2 seconds, for adding/updating cart, under 1 seconds to instant.
* **Checkout flow:** the checkout progress (cart – address – shipping – payment) must feel smooth with no delay, no sluggish. Same as above, 100 sessions, 95% of users, each step should only take under 3 seconds. The journey from cart to order confirmation should average approximately 25 seconds.
* **Throughput:** the system needs to handle a steady stream of orders without slowing down, at least 10 orders/minute while maintaining response times above. If session count rises above 100, let’s say 300 sessions, then the system needs to be able to scale up, using more resources, servers so there are no performance drops.
* **Payment Gateway Timeout Handling:** third-party payment methods (e.g. PayPal, afterpay) rely on an external gateway, which can be slow at times. Therefore, if a gateway takes more than 30 seconds to response, the system will stop waiting, tell the customer the issue and suggest retrying. The system is not allowed to let the customer hanging or double charge them.

1. **Portability**

**Goal** – The webpage needs to have the capable of running smoothly on different operating system. This will help the system in adapting with high-developing environment like nowadays and which can assist IT developer in implementing new functions or technology without modifying the whole codebase.

**Avoid** – Critical problem like “System only support macOS operating system” will lead to customer unsastifaction and this problem will limit the number of customers who can use the system which is a really bad idea when doing online business.

**Key scenarios**

* The system should be deployable in famous operating systems nowadays like Window, Linux or macOS which help expand the approach to as many customer as possible.
* The system can be used on mobile platforms like Android and iOS and the front-end of the system needs to be responsive so that it can be easily accessed and navigated through mobile phone.

## **Usability/Accessibility**

**Goals** – the system must be simple and accessible for first-time users to learn and use how to browse catalogue, add items to cart, etc. The website should follow the best practice; Swinsoft’s detailed user-interface guidelines.

**Avoid** – difficult and complex system to view and navigate. Those will make customers abandon their purchases, which counters the goal of expanding reach and facilitating online shopping.

**Key scenarios**

* **Task success rate:** given 10 customers, when they are asked to “find a product, add it to cart and place the order”, then 90% of them will be able complete the tasks within the 5-minute period.
* **Form clarity and error handling:** let’s say customer submit the payment form, however, there are invalid inputs. Field-level errors are announced, and focus moves to the first error; then progressively fix all the invalid inputs.
* **Accessibility:** the online store should work for people with disabilities (e.g. low vision, color blind). WCAG 2.1 AA standards, for instance, proper contrast, focus indicators and alt text for images has been chosen. 95% of elements are expected to be compliant.
* **Tablet/mobile responsiveness:** there are customers who use mobile devices to browse and order. Design must adapt to smaller screens. This means no side scrolling, text and images are able to resize depending on the viewport and tap targets such as buttons/links should be at least 44 pixels high, which is the finger-friendly aspect.

# Other Requirements

## **Menu Management**

## **Accessibility Support**

# Validation of Requirements

## **Validation Process Overview**

The validation process include some steps designed to evaluate the feasibility of requirements, we held regular group discussion to review each task and each team member act like actors to validate the tasks and give solutions about how to improve the tasks to align with the requirement.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task/Entity | Customer | Account | Shopping Cart | Order | OrderItem | Invoice | Receipt | Payment | Shipment | Product |
| Create customer account | C,R | C,R |  |  |  |  |  |  |  |  |
| Browsing store catalogue |  |  |  |  |  |  |  |  |  | R |
| Managing shopping cart | R |  | C,R,U,D | C | C,R |  |  |  |  | R |
| Creating invoices | R |  | R | R | R | C,R |  |  |  | R |
| Creating receipts | R |  |  | R | R | R | C,R | R |  |  |
| Handling payments | R |  |  | U |  | U | R | C,R |  |  |
| Managing goods |  |  |  |  |  |  |  |  |  | C,R,U,D |
| Shipment | R |  |  | U |  | R | R |  | C | R |

# Possible Solutions

---- Từ đây trở xuống là nháp ---