Software requirement specification document for Weirdough Cookies & Coffee Website

Salah Elabd, Rwan Haitham, Erfan Nada, Fajr Reda, Mohammed Ammar
Supervised by: Dr. Mohamed Labib

April 10, 2025

1 Introduction

1.1 Purpose of this document

This document outlines the requirements for the Weirdough Cookies & Coffee website project. It is intended for stakeholders including business owners, project managers, developers, system engineers, and end users. The document aims to define the objectives, scope and business context to ensure successful development of the website.

Business Owners: To align the website with the mission and goals of the company.

Developers and Designers: To implement the website according to the specified requirements.

Marketing and Sales Teams To ensure that the website supports business growth and customer engagement.

1.2 Scope of this document

The requirements definition effort includes collaboration between users, customers, system engineers and developers to build a functional and engaging website for Weirdough Cookies & Coffee. This website will provide information on the brand, its products, locations, and will allow customers to place orders online.

The scope includes:

- Designing a user-friendly interface.
- Implementing an e-commerce platform for online orders.

- Providing a content management system for updates and promotions.

Constraints:

- The development time is limited and must be completed within a strict deadline.
- The brand has not allocated a budget for this project, which means that resources must be optimized efficiently.

1.3 Overview

The Weirdough Cookies & Coffee website is a digital platform designed to enhance customer engagement and streamline online operations. As a result of the requirements elicitation process, the website will serve as a central hub for customers to explore the brand, purchase products, and interact with the business.

Key features of the website include:

- **Product Catalog**: Displaying a wide range of cookies, coffee, and other offerings.
- **Online Ordering**: Allowing customers to place orders for delivery or pickup.
- **Store Locator**: Helping users find nearby Weirdough locations.
- **Promotions and Loyalty Programs**: Highlighting discounts, special offers, and rewards for loyal customers.
- **Brand Story**: Sharing the mission, values, and story behind Weirdough Cookies & Coffee.

The website will be designed to provide a seamless and pleasant user experience, ensuring accessibility, mobile responsiveness, and secure transactions. It aims to support business growth by driving online sales, increasing brand awareness, and fostering customer loyalty in the Kingdom of Saudi Arabia (KSA).

1.4 Business Context

The Weirdough Cookies & Coffee website is being developed under the sponsorship of Weirdough Cookies & Coffee, a growing brand specializing in artisanal cookies and premium coffee. The brand is expanding its presence in the Kingdom of Saudi Arabia (KSA) and aims to establish a strong digital footprint to support its growth and customer engagement efforts.

Mission Statement:
"To deliver exceptional hand-made cookies and coffee that bring joy to every moment, while fostering a sense of community and sustainability."

Organizational Objectives:

- **Market Expansion**: Increase brand awareness and customer base in the KSA region.
- **Customer Engagement**: Build lasting relationships with customers through an intuitive and engaging online platform.
- **Sales Growth**: Drive online and in-store sales by providing a seamless ordering experience and promoting special offers.
- **Operational Efficiency**: Streamline order management, customer support, and promotional activities through an integrated digital platform.

The development of the Weirdough Cookies & Coffee aligns with these objectives by providing a robust digital platform to connect with customers, enhance brand visibility, and support business operations in the region.

2 General Description

2.1 Product Functions

The Weirdough Cookies & Coffee website will serve as an interactive platform for customers to browse products, place orders, and engage with the brand. Key functionalities include:

- **Product Browsing:** Allowing users to explore the menu, including cookies, coffee, and other offerings.
- **Online Ordering:** Enabling customers to place orders for delivery or pickup.
- **Store Locator:** Helping users find nearby Weirdough locations.
- **Promotions and Loyalty Programs:** Displaying discounts, special offers, and rewards for loyal customers.
- **Content Management:** Allowing administrators to update product information, promotions, and other content.
- **User Accounts:** Providing customers with the ability to create accounts, track orders, and manage preferences.

2.2 Similar System Information

This website will function as a stand-alone platform but may integrate with third-party services for payment processing, order fulfillment and tracking, marketing and customer engagement. The website will not be a component of a larger product but will serve as the primary digital touchpoint for the brand in the KSA region.

2.3 User Characteristics

The user community for the Weirdough Cookies & Coffee website includes:

- **General Customers**: Individuals of all ages who may or may not have technical expertise. The website must be intuitive and easy to navigate.
- **Franchise Owners and Managers**: Users who require access to backend systems for managing orders, inventory, and promotions.
- **Administrators**: Technical staff responsible for maintaining and updating the website.

The design must accommodate users with varying levels of technical expertise, ensuring accessibility and ease of use for all.

2.4 User Problem Statement

The current challenges faced by the user community include:

- **Limited Online Presence**: Customers cannot easily access product information or place orders online.
- **Inefficient Order Management**: Franchise owners struggle to manage orders and promotions effectively.
- **Lack of Brand Engagement**: Customers have limited opportunities to interact with the brand and stay informed about new offerings.

The website aims to address these issues by providing a centralized platform for online interactions and transactions.

2.5 User Objectives

From the users' perspective, the system should achieve the following objectives:

- **Seamless Ordering**: Provide a fast and intuitive process for placing orders online.
- **Accessible Information**: Ensure that product details, promotions, and store locations are easily accessible.
- **Personalized Experience**: Allow users to create accounts, save preferences, and receive tailored recommendations.
- **Mobile-Friendly Design**: Ensure the website is fully responsive and accessible on all devices.
- **Secure Transactions**: Implement robust security measures to protect user data and payment information.

These objectives align with the business goals of increasing customer satisfaction and driving sales.

2.6 General Constraints

- The website must be optimized for both desktop and mobile users.
- Integration with a payment gateway must comply with financial regulations.
- Hosting and maintenance should be cost-effective due to budget constraints.
- Security measures must be in place to protect customer data and transactions.

3 Functional Requirements

Table 1: Functional Requirements for Weirdough Cookies & Coffee Website

Function Name	Short, imperative sentence stating highest ranked functional requirement.
Description	A full description of the requirement.
Critically	Describes how essential this requirement is to the overall system.
Technical issues	Describes any design or implementation issues involved in satisfying this requirement.
Cost and schedule	Describes the relative or absolute costs associated with this issue.
Risks	Describes the circumstances under which this requirement might not be able to be satisfied, and what actions can be taken to reduce the probability of this occurrence.
Dependencies with other requirements	Describes interactions with other requirements.
Pre-Condition	The state of the system before the running of the function.
Post-Condition	The state of the system after the run of the function.

3.1 Shopping Cart Management

- **Function Name**: Manage Shopping Cart
- **Description**: Allow users to add, remove, and update items in their shopping cart.
- **Critically**: High. Essential for purchase flow.
- **Technical Issues**: Real-time updates and synchronization with inventory.
- **Cost and Schedule**: Moderate. Requires frontend and backend updates.
- **Risks**: Cart data loss or incorrect totals.
- **Dependencies**: Product catalog and order system.
- **Pre-Condition**: User views a product.
- **Post-Condition**: Product is added or updated in the cart.

3.2 Guest Checkout

- **Function Name**: Enable Guest Checkout
- **Description**: Allow customers to check out without creating an account.
- **Critically**: Medium. Encourages conversions from new users.
- **Technical Issues**: Collect and store temporary user data.
- **Cost and Schedule**: Low. Minimal backend changes needed.
- **Risks**: Missed opportunities for customer retention.

- **Dependencies**: Checkout and payment system.
- **Pre-Condition**: User initiates checkout.
- **Post-Condition**: Order is placed without account creation.

3.3 Order Tracking

- **Function Name**: Track Order Status
- **Description**: Allow users to view their order status and delivery updates.
- **Critically**: Medium. Enhances customer trust.
- **Technical Issues**: Sync with delivery provider APIs.
- **Cost and Schedule**: Moderate. Requires integration work.
- **Risks**: Out-of-sync status updates or API errors.
- **Dependencies**: Order management and delivery APIs.
- **Pre-Condition**: Order has been placed.
- **Post-Condition**: Status is visible to the user.

3.4 User Profile Management

- **Function Name**: Edit User Profile
- **Description**: Users can update their contact info, passwords, and preferences.
- **Critically**: Medium. Useful for ongoing account management.
- **Technical Issues**: Secure access and data validation.
- **Cost and Schedule**: Moderate. Backend and frontend work required.
- **Risks**: Data breaches or validation errors.
- **Dependencies**: User database.
- **Pre-Condition**: User is logged in.
- **Post-Condition**: Profile data is updated.

3.5 Newsletter Subscription

- **Function Name**: Subscribe to Newsletter
- **Description**: Allow users to sign up for email updates.
- **Critically**: Low. Optional user engagement feature.
- **Technical Issues**: Email list management.
- **Cost and Schedule**: Low. Simple integration.
- **Risks**: Spam complaints or unsubscribes.
- **Dependencies**: Email service provider integration.
- **Pre-Condition**: User submits email address.
- **Post-Condition**: Email is added to subscriber list.

3.6 Product Reviews

- **Function Name**: Submit Product Reviews
- **Description**: Allow verified users to review and rate products.
- **Critically**: Medium. Builds customer trust.
- **Technical Issues**: Prevent spam and ensure moderation.
- **Cost and Schedule**: Moderate. Requires database and interface updates.
- **Risks**: Fake or malicious reviews.
- **Dependencies**: User and product system.
- **Pre-Condition**: User has purchased a product.
- **Post-Condition**: Review is submitted and visible.

3.7 Social Media Integration

- **Function Name**: Integrate Social Sharing
- **Description**: Allow users to share products on social platforms.
- **Critically**: Low. Adds brand exposure.
- **Technical Issues**: API integration with platforms like Instagram and Facebook.
- **Cost and Schedule**: Low. Simple frontend implementation.
- **Risks**: Broken links or outdated API versions.
- **Dependencies**: Social platform APIs.
- **Pre-Condition**: User clicks share button.
- **Post-Condition**: Content is shared on social media.

3.8 Admin Order Management

- **Function Name**: Admin Order Panel
- **Description**: Allow admins to view, update, or cancel customer orders.
- **Critically**: High. Essential for internal operations.
- **Technical Issues**: Role-based access control and security.
- **Cost and Schedule**: Moderate. Admin backend required.
- **Risks**: Unauthorized access or human error.
- **Dependencies**: CMS and order system.
- **Pre-Condition**: Admin logs in.
- **Post-Condition**: Orders are updated accordingly.

3.9 Inventory Management

- **Function Name**: Track Inventory
- **Description**: Automatically update inventory as products are sold.
- **Critically**: High. Prevents overselling and tracks stock levels.
- **Technical Issues**: Real-time sync and fail-safes needed.
- **Cost and Schedule**: High. Significant backend work.
- **Risks**: Stock misrepresentation or database sync failures.
- **Dependencies**: Product catalog and order system.
- **Pre-Condition**: Order is placed.
- **Post-Condition**: Inventory count is updated.

3.10 Contact Form Submission

- **Function Name**: Submit Contact Form
- **Description**: Allow users to contact the store with inquiries.
- **Critically**: Low. Helpful for customer support.
- **Technical Issues**: Email delivery and spam prevention.
- **Cost and Schedule**: Low. Simple implementation.
- **Risks**: Spam messages or email misdelivery.
- **Dependencies**: Mail server integration.
- **Pre-Condition**: User fills out contact form.
- **Post-Condition**: Message is sent to admin.

3.11 Gift Card Purchase

- **Function Name**: Buy Gift Cards
- **Description**: Enable users to purchase digital gift cards.
- **Critically**: Medium. Encourages gift-giving and brand exposure.
- **Technical Issues**: Unique code generation and email delivery.
- **Cost and Schedule**: Moderate. Requires backend development.
- **Risks**: Delivery issues or invalid codes.
- **Dependencies**: Payment system and email service.
- **Pre-Condition**: User selects gift card option.
- **Post-Condition**: Gift card is purchased and sent.

3.12 Gift Card Redemption

- **Function Name**: Redeem Gift Cards
- **Description**: Allow customers to use gift card codes at checkout.
- **Critically**: Medium. Complements gift card purchase feature.
- **Technical Issues**: Code validation and balance tracking.
- **Cost and Schedule**: Moderate. Backend changes needed.
- **Risks**: Fraudulent or expired codes.
- **Dependencies**: Checkout and gift card systems.
- **Pre-Condition**: User enters gift card code.
- **Post-Condition**: Discount is applied.

3.13 Coupon Code Functionality

- **Function Name**: Apply Coupons
- **Description**: Users can enter promotional codes to receive discounts.
- **Critically**: Medium. Supports marketing campaigns.
- **Technical Issues**: Code validation and conditional logic.
- **Cost and Schedule**: Moderate. Requires database and checkout logic.
- **Risks**: Code abuse or stacking issues.
- **Dependencies**: Promotions system and checkout process.
- **Pre-Condition**: User enters code at checkout.
- **Post-Condition**: Discount is applied.

3.14 Product Customization

- **Function Name**: Customize Products
- **Description**: Allow customers to customize cookie flavors or coffee ingredients.
- **Critically**: Medium. Enhances user experience and personalization.
- **Technical Issues**: Dynamic forms and order logic.
- **Cost and Schedule**: Moderate. Requires product variation management.
- **Risks**: Misconfigured or unavailable options.
- **Dependencies**: Product and order systems.
- **Pre-Condition**: User selects a customizable product.
- **Post-Condition**: Custom product is added to cart.

3.15 Multi-language Support

- **Function Name**: Support Multiple Languages
- **Description**: Offer content in multiple languages for broader accessibility.
- **Critically**: Low. Enhances accessibility for diverse users.
- **Technical Issues**: Language selection and content translation.
- **Cost and Schedule**: High. Requires translation management.
- **Risks**: Inconsistent translations.
- **Dependencies**: CMS and UI framework.
- **Pre-Condition**: User selects a different language.
- **Post-Condition**: Website content updates to selected language.

3.16 Mobile Responsiveness

- **Function Name**: Ensure Mobile Compatibility
- **Description**: Ensure all website features work seamlessly on mobile devices.
- **Critically**: High. Majority of users access from mobile.
- **Technical Issues**: UI testing on different screen sizes.
- **Cost and Schedule**: Moderate. Requires responsive design.
- **Risks**: Broken layout or poor UX.
- **Dependencies**: All frontend components.
- **Pre-Condition**: User accesses site from a mobile device.
- **Post-Condition**: Mobile-friendly UI is presented.

3.17 Accessibility Compliance

- **Function Name**: Implement Accessibility Standards
- **Description**: Ensure the website complies with WCAG standards for accessibility.
- **Critically**: High. Legally and ethically important.
- **Technical Issues**: ARIA roles, contrast ratios, and keyboard navigation.
- **Cost and Schedule**: High. Requires thorough audit and implementation.
- **Risks**: Non-compliance or poor UX for disabled users.
- **Dependencies**: UI framework and frontend logic.
- **Pre-Condition**: User accesses site.
- **Post-Condition**: All features accessible by assistive technology.

3.18 Loyalty Program

- **Function Name**: Manage Loyalty Points
- **Description**: Reward repeat customers with points that can be redeemed for discounts.
- **Critically**: Medium. Boosts customer retention.
- **Technical Issues**: Point tracking and redemption logic.
- **Cost and Schedule**: Moderate. Requires backend and UI updates.
- **Risks**: Miscalculated points or redemption issues.
- **Dependencies**: User account and checkout system.
- **Pre-Condition**: User completes a purchase.
- **Post-Condition**: Points are awarded or redeemed.

3.19 Store Locator

- **Function Name**: Find Nearby Stores
- **Description**: Help users find physical store locations using a map interface.
- **Critically**: Low. Useful for local customers.
- **Technical Issues**: Map API integration and geolocation.
- **Cost and Schedule**: Moderate. Needs frontend and API setup.
- **Risks**: Incorrect locations or permissions errors.

- **Dependencies**: Map API and store database.
- **Pre-Condition**: User accesses store locator.
- **Post-Condition**: Store locations are shown on a map.

3.20 Live Chat Support

- **Function Name**: Provide Live Chat
- **Description**: Allow users to chat with support in real time.
- **Critically**: Medium. Enhances customer service.
- **Technical Issues**: Integration with live chat services.
- **Cost and Schedule**: Moderate. May require third-party service.
- **Risks**: Downtime or unavailable agents.
- **Dependencies**: Chat provider API and user session.
- **Pre-Condition**: User initiates chat.
- **Post-Condition**: Conversation is started.

3.21 Product Recommendation Engine

- **Function Name**: Recommend Products
- **Description**: Suggest related or popular products based on user activity.
- **Critically**: Medium. Encourages upsells.
- **Technical Issues**: User behavior tracking and algorithm logic.
- **Cost and Schedule**: High. Requires personalization engine.
- **Risks**: Irrelevant or repetitive suggestions.
- **Dependencies**: Product catalog and user behavior data.
- **Pre-Condition**: User browses products.
- **Post-Condition**: Personalized recommendations are shown.

3.22 Order History

- **Function Name**: View Past Orders
- **Description**: Show users a list of their previous purchases.
- **Critically**: Medium. Useful for repeat purchases.
- **Technical Issues**: Data retrieval and display.
- **Cost and Schedule**: Low. Standard backend functionality.
- **Risks**: Missing or incomplete history.
- **Dependencies**: User account and order database.
- **Pre-Condition**: User is logged in.
- **Post-Condition**: Order history is displayed.

3.23 Password Recovery

- **Function Name**: Reset Password
- **Description**: Let users reset their password via email.
- **Critically**: High. Basic user account functionality.
- **Technical Issues**: Secure token handling.
- **Cost and Schedule**: Low. Standard implementation.
- **Risks**: Security vulnerabilities.
- **Dependencies**: User authentication system and email service.
- **Pre-Condition**: User forgets password.
- **Post-Condition**: Password is reset.

3.24 Order Confirmation Email

- **Function Name**: Send Order Email
- **Description**: Send customers a confirmation email after purchase.
- **Critically**: High. Confirms transaction.
- **Technical Issues**: Email delivery reliability.
- **Cost and Schedule**: Low. Simple integration.
- **Risks**: Missed emails or spam filtering.
- **Dependencies**: Email service and order system.
- **Pre-Condition**: Order is placed.
- **Post-Condition**: Confirmation email is sent.

3.25 User Login and Registration

- **Function Name**: User Authentication
- **Description**: Enable users to register and log in to their accounts.
- **Critically**: High. Core functionality.
- **Technical Issues**: Secure data handling and session management.
- **Cost and Schedule**: Moderate. Standard account setup.
- **Risks**: Brute-force attacks or weak password handling.
- **Dependencies**: User database and authentication service.
- **Pre-Condition**: User accesses login or registration form.
- **Post-Condition**: User is authenticated.

3.26 Saved Favorites

- **Function Name**: Save Favorite Products
- **Description**: Let users bookmark or favorite items.
- **Critically**: Low. Enhances browsing experience.
- **Technical Issues**: Store preferences in database.
- **Cost and Schedule**: Low. Simple UI and backend support.
- **Risks**: Data loss or incorrect state.
- **Dependencies**: User account and product system.
- **Pre-Condition**: User views product.
- **Post-Condition**: Product is added to favorites list.

3.27 Product Availability Alerts

- **Function Name**: Notify Product Restock
- **Description**: Notify users when out-of-stock items become available.
- **Critically**: Medium. Prevents lost sales.
- **Technical Issues**: Inventory monitoring and email automation.
- **Cost and Schedule**: Moderate. Needs backend logic.
- **Risks**: Missed alerts or spam issues.
- **Dependencies**: Inventory system and email service.
- **Pre-Condition**: Product is out of stock.
- **Post-Condition**: Notification is sent when restocked.

3.28 Cookie Policy Consent

- **Function Name**: Display Cookie Consent Banner
- **Description**: Inform users about cookie usage and request consent.
- **Critically**: High. Legal compliance (e.g., GDPR).
- **Technical Issues**: Consent tracking and script control.
- **Cost and Schedule**: Low. Simple implementation.
- **Risks**: Non-compliance with privacy laws.
- **Dependencies**: Frontend and session logic.
- **Pre-Condition**: User visits the website.
- **Post-Condition**: Consent is recorded.

3.29 Analytics Integration

- **Function Name**: Track User Behavior
- **Description**: Integrate analytics tools to monitor website usage.
- **Critically**: Medium. Supports business decisions.
- **Technical Issues**: Proper tagging and event tracking.
- **Cost and Schedule**: Low. Uses third-party tools like Google Analytics.
- **Risks**: Misconfigured or inaccurate data.
- **Dependencies**: Analytics service.
- **Pre-Condition**: User interacts with site.
- **Post-Condition**: Data is recorded.

3.30 Scheduled Delivery Option

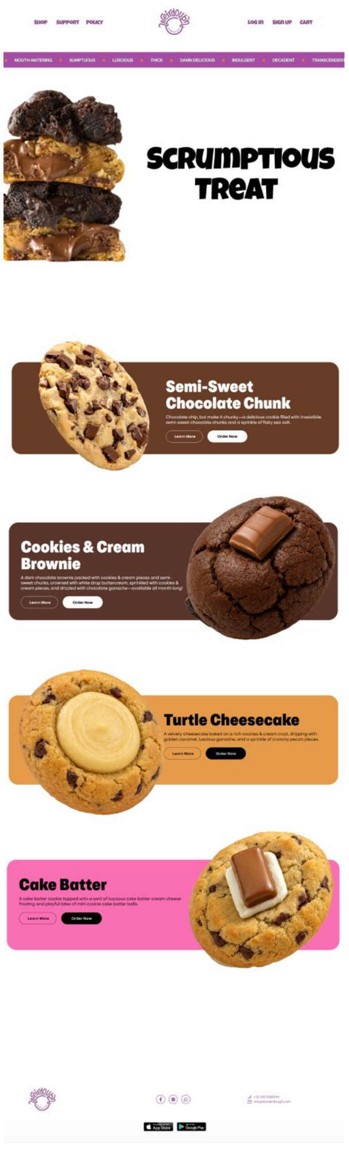
- **Function Name**: Choose Delivery Date
- **Description**: Allow customers to select a preferred delivery date at checkout.
- **Critically**: Medium. Useful for gifts and planning.
- **Technical Issues**: Delivery logic and calendar integration.
- **Cost and Schedule**: Moderate. Requires UI and logic update.
- **Risks**: Missed or incorrect delivery windows.
- **Dependencies**: Checkout and delivery system.
- **Pre-Condition**: User proceeds to checkout.
- **Post-Condition**: Delivery is scheduled for selected date.

4 Interface Requirements

4.1 User Interfaces

4.1.1 GUI

The graphical user interface will be designed to provide a visually appealing and user-friendly experience. It will include a homepage displaying featured products, a categorized menu for easy navigation, an interactive cart, and a secure checkout process. The admin panel will feature a dashboard for managing orders, inventory, and customer data. UI mockups will be provided to illustrate design consistency across desktop and mobile platforms.



Chocolate chip, but make it chunky—a delicious cookie filled with irresistible semi-sweet chocolate chunks and a sprinkle of flaky sea salt.

A dark chocolate brownie packed with cookies & cream pieces and semi-sweet chunks, crowned with white drop buttercream, sprinkled with cookies & cream pieces, and drizzled with chocolate ganache—available all month long!

A velvety cheesecake baked on a rich cookies & cream crust, dripping with golden caramel, luscious ganache, and a sprinkle of crunchy pecan pieces.

A cake batter cookie topped with a swirl of luscious cake batter cream cheese frosting and playful bites of mini cookie cake batter balls.

Figure 1: Landing Page

4.1.2 CLI

The system will not include a command-line interface (CLI) as it is primarily web-based and user-driven through the GUI. However, developers and administrators

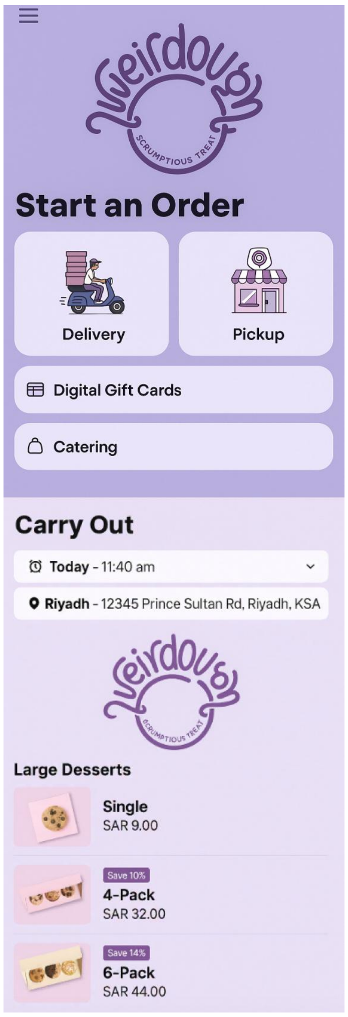


Figure 2: Order Placement

may utilize CLI tools for system maintenance, database management, and server deployment.

4.1.3 API

The system may include an API to facilitate third-party integrations, such as payment gateways (e.g., Stripe, PayPal, Mada) and delivery services. API endpoints will handle authentication, order management, product retrieval, and transaction processing. Each API function will be documented with request parameters, response structures, and example calls.

4.1.4 Diagnostics or ROM

The system will include logging mechanisms to track errors, transactions, and system performance. Debugging tools and admin-level logs will help diagnose and resolve issues efficiently.

4.2 Hardware Interfaces

The website will be compatible with standard hardware devices such as desktops, laptops, tablets, and smartphones. It may also integrate with POS (Point of Sale) systems for in-store transactions and barcode scanners for inventory management.

4.3 Communications Interfaces

The system will support secure HTTPS communication for data encryption and security. Email and SMS notifications will be used for order confirmations and promotional updates. Real-time chat support may be integrated to enhance customer service.

4.4 Software Interfaces

The platform will interact with a relational database for managing user accounts, product inventory, and order histories. It will also interface with third-party authentication services (e.g., Google and Apple Sign-In) and cloud storage solutions for media assets. The system will adhere to industry security protocols to protect sensitive user data.

5 Performance Requirements

The Weirdough Cookies & Coffee website must meet the following performance requirements to ensure a smooth and responsive user experience:

5.1 Speed Requirements

- **Page Load Time**: All web pages must load within **2 seconds** under normal operating conditions.
- **Search Response Time**: Product search results must be displayed within **1 second** of the user submitting a query.
- **Order Processing Time**: Online orders must be processed and confirmed within **3 seconds** of payment completion.

5.2 Memory Requirements

- **Server Memory**: The system must be able to handle up to **10,000 concurrent users** without performance degradation.
- **Client Memory**: The database should be optimized to support high read/write operations without excessive memory consumption.

5.3 Scalability

- The system must support a **20% annual growth** in user traffic without requiring significant architectural changes.

- The website must be able to handle **peak traffic** during promotional events (e.g., up to **50,000 concurrent users**).

5.4 Availability

- The website must have an **uptime of 99.9%**, ensuring it is accessible to users at all times, except during scheduled maintenance.

5.5 Error Handling

- The system must recover from errors (e.g., network interruptions, server failures) within **10 seconds** and provide users with appropriate feedback.

6 Design Constraints

6.1 Standards Compliance

The website must follow web standards, including security measures like HTTPS and data protection. It should be easy for everyone to use, including people with disabilities. Search engine optimization guidelines should also be followed to help the site rank better on search engines.

6.2 Hardware Limitations

The website must be optimized to run smoothly on various devices, including desktops, tablets, and mobile phones, without requiring high processing power or large storage. It should be compatible with common web browsers and work efficiently on standard hosting servers.

7 Other non-functional attributes

7.1 Security

- Security measures include password protection, encryption, HTTPS, secure login methods, and role-based access control.
- The system must protect user data from threats like hacking, unauthorized access, or data breaches.

7.2 Reliability

- The system should work consistently without unexpected failures.
- Downtime or crashes should be minimized to ensure a smooth user experience.
- Backup and recovery mechanisms should be in place in case of failures.

7.3 Portability

- The system should work across different operating systems, devices, and web browsers.
- It should be easy to move the website from one hosting server to another if needed.

7.4 Re-usability

- The code, components, or modules should be reusable for future projects or different parts of the system.
- Common functionalities (such as authentication, payment processing, and database management) should be implemented in a way that allows them to be used again.

8 Preliminary Object-Oriented Domain Analysis

8.1 Inheritance Relationships

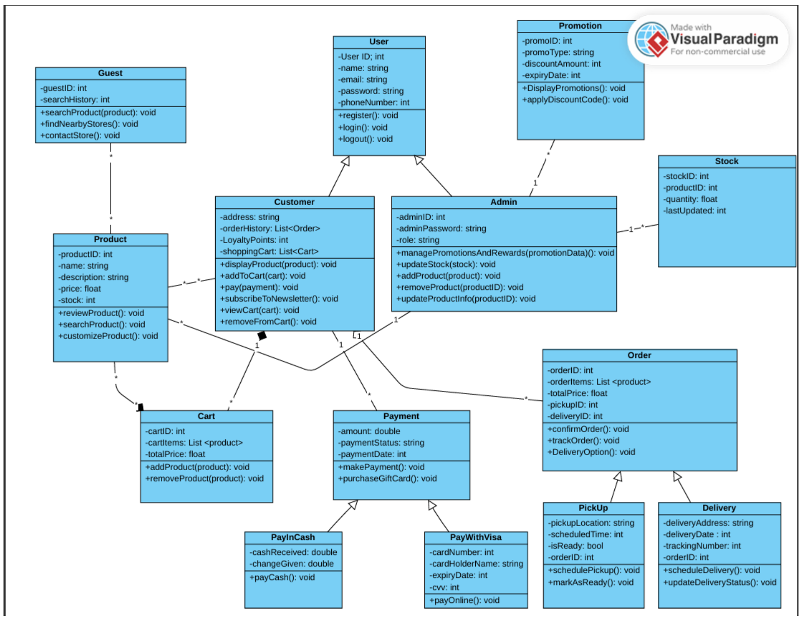


Figure 3: Inheritance Relations

8.2 Class descriptions

8.2.1 User

- **Abstract or Concrete:** Abstract
- **Superclasses:** None
- **Subclasses:** Customer, Admin

- **Purpose:** Represents a generic user with authentication and profile details.
- **Collaborations:** Interacts with Orders, Promotions, and system authentication.
- **Attributes:**
  - userID: int
  - name: string
  - email: string
  - password: string
  - phoneNumber: int
- **Operations:**
  - +register(): void
  - +login(): void
  - +logout(): void
- **Constraints:** Emails must be unique; Password must be securely hashed.

8.2.2 Customer

- **Abstract or Concrete:** Concrete
- **Superclass:** User
- **Subclasses:** None
- **Purpose:** Represents registered customers who can shop and place orders.
- **Collaborations:** Product, Cart, Payment, Order.
- **Attributes:**
  - address: string
  - orderHistory: List¡Order¿
  - LoyaltyPoints: int
  - shoppingCart: List¡Cart¿
- **Operations:**
  - +displayProduct(product): void
  - +addToCart(cart): void
  - +pay(payment): void
  - +subscribeToNewsletter(): void
  - +viewCart(cart): void
  - +removeFromCart(): void

8.2.3 Admin

- **Abstract or Concrete:** Concrete
- **Superclass:** User
- **Subclasses:** None
- **Purpose:** Manages the backend operations of the system.
- **Collaborations:** Stock, Product, Promotion.
- **Attributes:**
  - adminID: int
  - adminPassword: string
  - role: string
- **Operations:**
  - +managePromotionsAndRewards(promotionData): void
  - +updateStock(stock): void
  - +addProduct(product): void
  - +removeProduct(productID): void
  - +updateProductInfo(productID): void

8.2.4 Guest

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Allows unregistered users to browse products.
- **Collaborations:** Product.
- **Attributes:**
  - guestID: int
  - searchHistory: int
- **Operations:**
  - +searchProduct(product): void
  - +findNearbyStores(): void
  - +contactStore(): void

8.2.5 Product

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Represents products available in the store.
- **Collaborations:** Stock, Admin, Customer, Guest.
- **Attributes:**
  - productID: int
  - name: string
  - description: string
  - price: float
  - stock: int
- **Operations:**
  - +reviewProduct(): void
  - +searchProduct(): void
  - +customizeProduct(): void

8.2.6 Cart

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Holds products selected by the customer.
- **Collaborations:** Product, Customer.
- **Attributes:**
  - cartID: int
  - cartItems: List¡Product¿
  - totalPrice: float
- **Operations:**
  - +addProduct(product): void
  - +removeProduct(product): void

8.2.7 Payment

- **Abstract or Concrete:** Abstract
- **Superclass:** None
- **Subclasses:** PayInCash, PayWithVisa
- **Purpose:** Handles payment transactions.
- **Collaborations:** Customer.
- **Attributes:**
  - amount: double
  - paymentStatus: string
  - paymentDate: int
- **Operations:**
  - +makePayment(): void
  - +purchaseGiftCard(): void

8.2.8 PayInCash

- **Abstract or Concrete:** Concrete
- **Superclass:** Payment
- **Subclasses:** None
- **Purpose:** Handles cash payments.
- **Collaborations:** Payment.
- **Attributes:**
  - cashReceived: double
  - changeGiven: double
- **Operations:**
  - +payCash(): void

8.2.9 PayWithVisa

- **Abstract or Concrete:** Concrete
- **Superclass:** Payment
- **Subclasses:** None
- **Purpose:** Handles Visa card payments.

- **Collaborations:** Payment.
- **Attributes:**
  - cardNumber: int
  - cardHolderName: string
  - expiryDate: int
  - cvv: int
- **Operations:**
  - +payOnline(): void

8.2.10 Order

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Represents the order placed by a customer.
- **Collaborations:** Customer, Payment, Pickup, Delivery.
- **Attributes:**
  - orderID: int
  - orderItems: List¡Product¿
  - totalPrice: float
  - pickupID: int
  - deliveryID: int
- **Operations:**
  - +confirmOrder(): void
  - +trackOrder(): void
  - +deliveryOption(): void

8.2.11 Stock

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Manages product inventory.
- **Collaborations:** Admin, Product.

- **Attributes:**
  - stockID: int
  - productID: int
  - quantity: float
  - lastUpdated: int
- **Operations:** None

8.2.12 Promotion

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Represents promotions available to users.
- **Collaborations:** Admin, User.
- **Attributes:**
  - promoID: int
  - promoType: string
  - discountAmount: int
  - expiryDate: int
- **Operations:**
  - +displayPromotions(): void
  - +applyDiscountCode(): void

8.2.13 Pickup

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Handles pickup orders.
- **Collaborations:** Order.
- **Attributes:**
  - pickupLocation: string
  - scheduledTime: int
  - isReady: bool

    - orderID: int

- **Operations:**
    - +schedulePickup(): void
    - +markAsReady(): void

8.2.14 Delivery

- **Abstract or Concrete:** Concrete
- **Superclass:** None
- **Subclasses:** None
- **Purpose:** Handles delivery of orders.
- **Collaborations:** Order.
- **Attributes:**
    - deliveryAddress: string
    - deliveryDate: int
    - trackingNumber: int
    - orderID: int
- **Operations:**
    - +scheduleDelivery(): void
    - +updateDeliveryStatus(): void

9 Operational Scenarios

9.1 Scenario 1: Browsing Products

- **Actor**: Customer.
- **Description**: A customer visits the website to explore the menu and learn about available products.
- **Steps**:
    1. The customer opens the website and navigates to the **MENU** page.
    2. The customer uses filters (e.g., category, price range) to narrow down the product list.
    3. The customer clicks on a product to view detailed information (e.g., price, availability).
    4. The customer adds the product to their cart.
- **Expected Outcome**: The customer can easily browse products, apply filters, and view detailed information.

9.2 Scenario 2: Managing User Account

- **Actor**: Registered Customer.
- **Description**: A registered customer updates their profile information and views order history.
- **Steps**:
  1. The customer logs into their account.
  2. The customer navigates to the **Profile** page and updates their information (e.g., address, phone number).
  3. The customer navigates to the **Order History** page to view past orders.
  4. The customer logs out of their account.
- **Expected Outcome**: The customer can easily manage their profile and view order history.

9.3 Scenario 3: Admin Updating Product Information

- **Actor**: Administrator.
- **Description**: An administrator updates product details.
- **Steps**:
  1. The administrator logs into the CMS.
  2. The administrator navigates to the **Product Management** section.
  3. The administrator selects a product and updates its details (e.g., price, description, availability).
  4. The administrator saves the changes.
- **Expected Outcome**: The product information is updated and reflected on the website.

10 Preliminary Schedule Adjusted

This section provides an initial version of the project plan, including the major tasks to be accomplished, their interdependencies, and their tentative start/stop dates. The plan also includes information on hardware, software, and resource requirements. A Gantt chart is provided to visualize the project timeline.

10.1 Major Tasks

The following table outlines the major tasks and their descriptions.

Task	Description
Requirements Elicitation	Gather and document project requirements.
System Design	Design the system architecture and user interface.
Frontend Development	Develop the user interface and client-side functionality.
Backend Development	Develop server-side logic, databases, and APIs.
Integration	Integrate frontend and backend components.
Testing	Conduct unit testing, integration testing, and user acceptance testing.
Deployment	Deploy the website to the production environment.
Post-Launch Support	Monitor the system, fix bugs, and provide technical support.

Table 2: Major Tasks and Descriptions

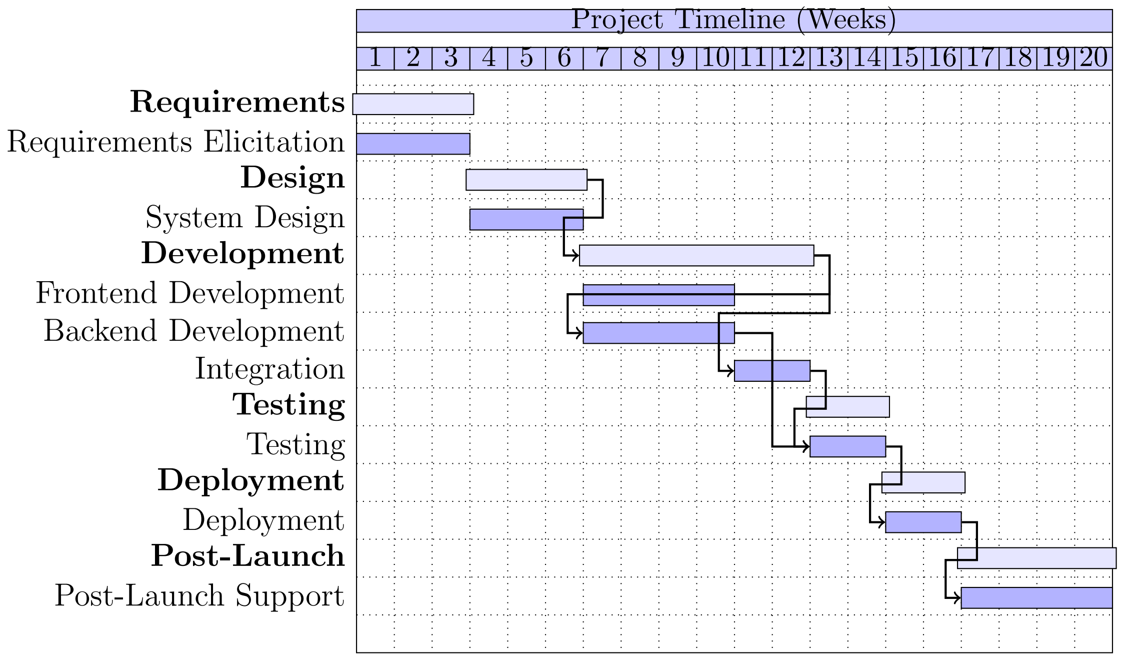
10.2 Resource Requirements

The following resources are required to complete the project:

- **Hardware**:
  - Development servers.
  - Testing devices (e.g., desktops, laptops).
- **Software**:
  - Frontend: HTML, CSS, JavaScript.
  - Backend: PHP.
  - Project Management: Trello (`https://trello.com/b/SfhC7x8e/weird-dough`).

10.3 Gantt Chart

The Gantt chart below visualizes the project timeline and task dependencies:



11 Preliminary Budget Adjusted

This section provides the expected total cost of the project in terms of time. Each functional requirement has an associated time cost (e.g., 1 hour, 2 hours, or more). The total time for all functional requirements is calculated, and this will be the expected time cost.

11.1 Functional Requirements and Time Estimates

The following table lists the functional requirements and their estimated time costs:

Functional Requirement	**Estimated Time (Hours)**
User Registration and Authentication	10
Product Browsing	12
Online Ordering	18
Store Locator	8
Promotions and Loyalty Programs	12
Content Management System (CMS)	20
Total	**80 Hours**

Table 3: Functional Requirements and Time Estimates

11.2 Total Expected Time Cost

Based on the functional requirements, the total expected time cost for the project is **80 hours**. Assuming a standard workweek of 40 hours, this translates to approximately **2 weeks** of full-time work.

12 Appendices

This section specifies other useful information for understanding the requirements. All SRS documents should include at least the following two appendices:

12.1 Definitions, Acronyms, and Abbreviations

Below are the definitions of key terms, acronyms, and abbreviations used in this document:

Term	**Definition**
SRS	Software Requirements Specification.
KSA	Kingdom of Saudi Arabia.
API	Application Programming Interface.
UI	User Interface. The visual and interactive part of the website that users interact with.
UX	User Experience. The overall experience of a user when interacting with the website.
SEO	Search Engine Optimization.

12.2 Collected Material

This section includes additional material collected during the requirements elicitation process, such as:

- **Stakeholder Interviews**:
  - Transcripts of interviews with business owners, franchise managers, and customers.
  - Key insights and feedback gathered during these interviews.
- **Competitor Analysis**:
  - A report analyzing competitors' websites, including features, user experience, and performance.
- **Design Mockups**:
  - Initial design mockups and wireframes created during the system design phase.

13 References

References

[1] Weirdough cookies & coffee instagram page. Accessed: 2023-10-15. [Online]. Available: https://www.instagram.com/itsweirdoughsa/?hl=en

[2] Google. (2023) Google maps api documentation. Accessed: 2023-10-15. [Online]. Available: https://developers.google.com/maps/documentation

[3] R. Pecinovsky, *OOP - Learn Object Oriented Thinking and Programming.* Tomáš Bruckner, 2013, accessed: 2023-10-15. [Online]. Available: https://books.google.com.eg/books?id=xb-sAQAAQBAJ

The Weirdough Instagram page [1] provides updates on new products and promotions.

The Google Maps API [2] will be used for the store locator feature.

The book [3] provides insights into object-oriented programming concepts.