<Project Name>

System-Wide Requirements Specification

1. Introduction

This document contains information about the system-wide requirement specifications of Doy! including information about system-wide function and quality requirements, system interfaces, business rules, system constraints, compliance and documentation.

2. System-Wide Functional Requirements (AI Prompt (3))

2.1. Auditing

The system shall log the timestamp, user ID, and device ID for every order placement, modification, or cancellation.

The system shall maintain audit logs for payment transactions with user ID and device ID.

The system shall maintain a record of all delivery status updates (e.g., "order accepted," "out for delivery," "delivered") with timestamps and the associated delivery personnel ID.

The system shall track and log all customer reviews, including timestamps, user IDs, and the nature of the query, storing this information in an audit trail for future reference.

The system shall maintain a record of all updates of user modification made by admin with associated user information.

2.2. Authentication and Authorization

The system shall require users (customers, delivery personnel, and restaurant staff) to authenticate using a registered email or phone number and a password.

The system shall store encrypted passwords and sensitive user information (e.g., payment details) in the database to ensure security.

The system shall allow admins to manage user roles (e.g., customer, delivery personnel, restaurant owner) and restrict access to specific features based on these roles.

The system shall support two-factor authentication (2FA) for admin accounts to enhance security.

The system has already defined the login information of admin.

All users can only access their permitted functionalities.

Authentication shall be implemented using OAuth 2.0/JWT Tokens for secure session management.

Users shall be able to reset their passwords through a secure email-based process.

2.3. Payment Processing

The system shall integrate with secure payment gateways (e.g., 3D Secure) to process online payments via the web .

The system shall encrypt all payment-related data during transmission and storage to ensure user security.

The system shall log all payment transactions, including successes and failures, with timestamps and user IDs for auditing purposes.

The system shall notify users of payment status (e.g., "payment successful," "payment failed") immediately after a transaction attempt.

2.4. Printing

The system shall generate printable order receipts for restaurants, including order details, customer information, and delivery instructions, upon order confirmation.

If a printing process fails (e.g., at a restaurant's printer), the system shall display an error message to the restaurant staff and log the failure for troubleshooting.

The system shall allow delivery personnel to print delivery labels (if needed) through the web page, including customer address and order number.

All the printable information will be available in PDF format.

2.5. Reporting

The system shall provide admins with access to a dashboard displaying real-time metrics, such as the number of active orders, delivery times, and customer ratings.

The system shall generate daily, weekly, and monthly reports for restaurant partners, summarizing their order volume, revenue, and customer feedback.

The system shall allow admins to view a log of all failed transactions or delivery issues on a dedicated reporting page for analysis and resolution.

2.6. Error Handling and Notifications

The system shall display clear and user-friendly error messages when a failure occurs (e.g., "Unable to process payment, please try again" or "Restaurant is currently unavailable").

The system shall send email and push notifications to users for critical events, such as order confirmations, delivery delays, or failed payment attempts.

The system shall log all critical errors, such as payment gateway failures, delivery assignment issues, or system outages, for troubleshooting and monitoring purposes.

3. System Qualities (AI Prompt (2))

3.1. Usability

3.1.2. Ease of Use

The system shall provide an intuitive and user-friendly interface.

Jobs such as browsing restaurants, placing orders, and tracking deliveries shall be achievable in 5 clicks or fewer.

The system shall use clear labels for buttons, menus, and forms to reduce user confusion.

The interface shall provide real-time feedback for user actions, such as success messages after order placement or error messages for invalid passwords.

3.1.3. Ease of Learning

The system shall provide tooltips and help icons throughout the interface to explain features.

The system shall offer a FAQ tab for common tasks.

The system shall use consistent design across all pages.

3.1.4. Usability Standards

The system shall adhere to Nielsen's 10 Usability Heuristics.

The system shall undergo usability testing with real users to identify and address pain points in the interface.

The system shall provide consistent feedback mechanisms like loading spinners to keep users informed about the status of their actions.

3.1.5. Localization

The system shall adapt to regional date, time, and currency formats based on the user's location or preferences.

The system shall be able to provide the user restaurants near their location.

3.2. Reliability

3.2.0. Availability

The system shall maintain an uptime of 99.9% during peak hours.

The system shall be designed to handle high traffic loads during peak ordering times without degradation in performance.

The system shall provide real-time monitoring of server health and resource usage.

3.2.1. Frequency and Severity of Failures

The system shall be designed to minimize the frequency of failures, with no more than 1 critical failure per month. Critical failures are system-wide outages or failures that prevent users from placing orders or completing payments.

3.2.2. Recoverability

The system shall maintain daily backups of all critical data (e.g., user accounts, orders, payment records) to enable quick recovery in case of data loss.

The system shall include rollback mechanisms to revert to a stable state in case of failed updates or deployments.

The system shall perform rollbacks for processes and transactions that fail.

3.3. Performance

3.3.1. Response Time

The system shall respond to user actions like browsing restaurants, placing orders etc. within 2 seconds under normal load conditions.

Payment processing shall be completed within 15 seconds, including communication with the payment gateway.

The system shall provide instant feedback for user inputs.

3.3.2. Startup / Shutdown Times

The system shall have a startup time of less than 3 seconds for all critical components.

3.4. Supportability

3.4.1. Adaptability and Upgrading

The system shall be designed with modular architecture to allow for easy maintanence and updates to source code.

The system shall be supported with version control, allowing for rolling back to previous versions in case of issues.

The system shall use automated testing for new updates to make sure old features still work.

3.4.2. Compatibility

The application shall be compatible with major web browsers like Google Chrome, Microsoft Edge and Firefox.

The system shall support integration with third-party services like Google Maps and 3D Secure.

3.4.3. Configurability

The system shall allow restaurants to configure their menu items and pricing.

The system shall allow customers to change their dietary preferences like enabling “vegan” mode or blocking allergens.

3.4.4. System Support and Monitoring

The system shall provide 24/7 monitoring and support for critical components, with alerts sent to administrators in case of issues.

4. System Interfaces (AI Prompt (1))

4.1. User Interfaces

4.1.1. Look & Feel

The interface should have a modern, clean, and user-friendly design that aligns with the branding of the food delivery system. The design should evoke a sense of trust and simplicity for users, whether they are customers, restaurants, or couriers.

Color Scheme: Use a neutral beige background with brown accents for headers and buttons, creating a

warm and inviting feel. Red is used for action buttons like "Sipariş Ver" (Place Order) to draw attention.

Typography: Use a clean and readable font for all text. Font sizes should vary for headings, subheadings, and body text to ensure clarity.

Icons: Use simple, intuitive icons (e.g., house icons for restaurants, burger icons for menus) to enhance navigation and understanding.

Dark Mode: Provide a toggle for light and dark modes to improve accessibility and user comfort.

Branding: The logo and branding elements (e.g., "Doy! Food Delivery") should be consistently displayed across all screens.

4.1.2. Layout and Navigation Requirements

The layout should be intuitive, with clear grouping of elements and logical navigation paths for different user roles (customers, restaurants, couriers).

- **Header**: The header includes the logo, a location selection bar, and login/logout buttons. It remains consistent across all screens.

- **Main Content Area**:
    - For customers: Displays recommended restaurants and menus with ratings and "Sipariş Ver" buttons.
    - For restaurants: Displays order management sections (e.g., "Onay Bekleyenler," "Onaylanmış," "Hazırlanıyor," "Hazır Siparişler").
    - For couriers: Displays assigned orders with delivery details.

- **Footer**: Includes social media links and branding.

- **Navigation**:
    - Customers can search for restaurants using the search bar.
    - Restaurants can manage orders using a Kanban-style layout.
    - Couriers can view and update delivery statuses.

- **Responsiveness**: The layout should adapt to different screen sizes (desktop, tablet, mobile).

4.1.3. Consistency

Consistency ensures that users can predict the behavior of the system, reducing the learning curve and improving usability.

Navigation Controls: Buttons like "Sipariş Ver" and "Sipariş Detayı" should always appear in the same style and location.
Screen Layouts: The header, main content area, and footer should remain consistent across all screens.
Terminology: Use consistent terms like "Sipariş" (Order) and "Kurye" (Courier) throughout the system.
Visual Design: Maintain consistent colors, fonts, and icon styles across all screens.
Feedback: Buttons should provide visual feedback (e.g., color change on hover or click).

4.1.4. User Personalization & Customization Requirements

The system should allow users to personalize their experience and customize certain settings.

Personalization:
- Display recommended restaurants based on the user's location and past orders.

- Show personalized dashboards for restaurants (order management) and couriers (assigned deliveries).

Customization:
- Allow users to save their preferred delivery addresses.
- Enable restaurants to customize menu items and availability.
- Provide couriers with options to update their availability status.

Dark Mode: Users can toggle between light and dark modes for better accessibility.

4.2. Interfaces to External Systems or Devices

The system must interface with external systems for payment processing, location services, and notifications.

4.2.1. Software Interfaces

Payment Gateway:
- Integrate with a third-party payment gateway (e.g., 3D-Secure) for secure online transactions.
- Use HTTPS for communication and JSON for data exchange.

Location Services:
- Use Google Maps API to fetch and display restaurant and delivery locations.
- Data format: JSON.

4.2.2. Hardware Interfaces

Doy! does not have any hardware interfaces.

4.2.3. Communications Interfaces

Real-Time Updates: WebSocket for real-time order status updates.

Email Notifications: SMTP to send order confirmations and updates to customers.

5. Business Rules (AI Prompt (4))

5.1. Registration & Authentication Rules

BR001 – Unique Account Credentials

Each user (customer, courier, restaurant owner, admin) must register using a unique email or phone number. Duplicate registration is not allowed.

BR002 – Email/Phone Verification Requirement

Users must verify their email/phone upon registration. Unverified accounts will be deactivated if not confirmed within timeout time.

BR003 – Admin Approval for Business Entities

Restaurant owners and couriers must be approved by an admin after registration. Accounts remain inactive until approval is granted.

5.2. Login & Security Rules

BR004 – Secure Login Requirement
All login operations must be protected by strong authentication mechanisms such as CAPTCHA and/or two-factor authentication (2FA).

5.3. Restaurant Rules

BR005 – Menu Management
Restaurants may update their menus to ensure consistency of food contents.

BR006 – Order Acceptance
Restaurants must accept or reject orders. Orders not responded to are auto-rejected.

BR007 – Profile Update Validation
Restaurant profile changes (e.g., delivery range, contact info) must be validated by the system before applying.

BR008 – Out-of-Stock Handling
Restaurants must mark unavailable items as “Temporarily Unavailable” rather than deleting them.

5.4. Order and Payment Rules

BR009 – Payment Method Restriction
Only digital payments are accepted (credit card, wallet). No cash handling is allowed.

BR010 – Refund Request Window
Refund requests can only be submitted within a timeframe of order completion.

BR011 – Refund Resolution
Admins must resolve refund or issue reports as soon as possible.

5.5. Delivery & Courier Rules

BR012 – Courier Availability Control
Couriers must manually set their status to “Available” to receive delivery tasks.

BR013 – Delivery Request Timeout
Couriers must respond to delivery requests within **2 minutes**, or it is auto-rejected.

BR014 – Consecutive Rejection Penalty
If a courier rejects most of the orders on purpose, they are punished with respect to some kind of rules in that day like timeout.

BR015 – Proof of Delivery
Couriers may be required to upload photo or signature proof after delivery.

BR016 – Statistic Access Rights
Only couriers with completed deliveries can access earnings, stats, and performance data.

5.6. Customer Rules

BR017 – Review Eligibility
Only verified customers who completed an order can leave a rating or review.

BR018 – Favorites Feature
Customers may save restaurants as favorites for quick access.

BR019 – Dietary Preference Alerting
Customers are warned when their selected food items contradict their allergy or diet settings.

BR020 – Track Order Access
Customers can track active orders in real time via status updates and courier location.

5.7. Admin & System Control Rules

BR021 – Account Suspension Policy
Admins may temporarily suspend or permanently ban couriers/restaurants for policy violations. Affected parties are notified via email.

BR022 – Platform Configuration Control
Admins can modify system-wide settings like delivery fees or promotions. Changes require validation and may trigger notifications.

BR023 – Complaint Escalation Process
Unresolved disputes may be escalated from admin to legal or third-party resolution teams.

BR024 – Registration Review History
The platform must log every decision (approval/rejection) related to courier/restaurant onboarding for transparency.

BR025 – Admin Audit Logging
Every action performed by an admin on the user database must be logged and timestamped for audit purposes.

5.8. Review Rules

BR026 – Authenticity Requirement
Only reviews tied to completed orders are allowed. Suspicious reviews are flagged by the system.

BR027 – Moderation Rule
All reviews go through moderation. Offensive, irrelevant, or spam content is removed.

BR028 – Conflict of Interest Prohibition
Users cannot leave reviews for businesses they own or are directly affiliated with.

BR029 – Review Edit Limitation
Each review can only be edited once. The edit must also comply with content rules.

BR030 – Business Response Visibility
Restaurant owners may respond to reviews. Responses are public and must remain professional

6. System Constraints

Development Platform Constraints:
- The system will be developed as a web application.
- The Model-View-Controller (MVC) design pattern must be used.

Programming Language Constraints:
- The system must be developed using Java and JavaScript programming languages.
- For backend development, Spring Boot framework must be used.
- For frontend development, React JS framework must be used, along with Figma UI as the UI kit.

Database Management System Constraints:
- For database systems, PostgreSQL must be used.

Security & Compliance Constraints:
- The system must implement HTTPS using TLS/SSL certificates.
- It must adhere to OWASP security guidelines.
- Strong encryption algorithms must be used for data at rest and in transit.

Deployment & Compatibility Constraints:
- The system must be compatible with major browsers, including Chrome, Safari, Edge, Opera, and Mozilla.
- Apache Tomcat may be used as the web server in case of deployment.
- The system must be capable of supporting a high number of concurrent users efficiently.
- The operating system for deployment should be a stable and secure Linux-based system.

Legal Constraints:
- The system must adhere to all relevant legal and regulatory standards, including but not limited to data protection laws and accessibility guidelines.

7. System Compliance

7.1. Licensing Requirements

The software system will comply with the guidelines set by Hacettepe University BBM384 Software Engineering Laboratory course and operates with the privacy principle. During the course, all source code components and documentation will remain private and accessible only to authorized persons involved in Google Drive and Gitlab. This confidentiality ensures the integrity of the learning experience and protects the intellectual property of both students and the university. However, upon completion of the course, all source code will be made publicly available on our teammates' s Github repositories and our team Gitlab repository. The shift from private to public access allows for transparency, information sharing, and potential collaboration with the broader community of developers and enthusiasts interested in the project. By adhering to this approach, we promote an environment of openness and collaboration in software development while maintaining the academic standards of the course.

7.2. Legal, Copyright, and Other Notices

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

7.3. Applicable Standards

ISO 27001	Information Security Management	Specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system.
ISO 9241	Ergonomics of Human-System Interaction	Provides guidance on ergonomic principles for the design of interactive systems, ensuring that human-computer interaction is optimized for efficiency, safety, and user satisfaction.
ISO 9001	Qualitiy Management Systems	Sets the criteria for the quality management system and is based on a set of quality management principles such as strong customer focus, senior management involvement. process approach and continuous improvement.
ISO 10002	Customer Satisfaction Management	Provides guidance on the process of handling customer complaints cllectively and efficiently, leading to increased customer satisfaction and retention.
ISO 25010	Software Quality Model	Defines a quality model for software product evaluation, covering characteristics such as functionality, reliability. Usability, efficiency, maintainability, and portability.
IEEE P7003	Standard for Algorithmic Bias Considerations	Provides guidelines for identifying and mitigating bias in algorithms and artificial intelligence systems.

8. System Documentation

The documentation for the system shall be developed and maintained by the project team throughout the software lifecycle, covering both front-end and back-end functionalities in parallel with implementation. It shall include setup guides, user instructions, process descriptions, and troubleshooting workflows to assist all stakeholders in effectively interacting with the platform.

Role-specific documentation shall be prepared for each user group—customers, couriers, restaurant owners, and administrators—clearly outlining the intended usage, key features, and access permissions relevant to their roles. This documentation will be designed with accessibility and usability in mind to accommodate users with varying levels of technical knowledge.

To ensure smooth usage, online help resources will be embedded directly into the platform interface through contextual help buttons, tooltips, and a central "Help Center." These will provide just-in-time assistance for users on tasks such as:

- Placing and managing orders (for customers)
- Handling deliveries and updating statuses (for couriers)
- Managing restaurant profiles, menus, and order flow (for restaurant owners)
- Utilizing monitoring and moderation tools (for administrators)

Upon successful registration, users shall receive a welcome email containing platform policies, usage instructions, and links to the relevant documentation for their role. For admins, additional secure manuals will be maintained to cover more advanced system functionalities and will be revised regularly as features evolve.

The system will also support user self-service through FAQs and may integrate a chatbot in future iterations to enhance support availability. To improve clarity during uncommon scenarios, the system shall display dynamic error messages and contextual guidance for incidents such as failed payments or order processing delays.

In the event of critical errors or service interruptions, users will be informed through system banners, emails, or in-app alerts. All such incidents will be logged by the system and periodically reviewed by the development team to ensure stability and responsiveness.

Legal agreements, terms of service, and licensing notices shall be presented during the registration process and remain accessible via a permanent footer link on the platform. Users must accept these terms to complete their registration and begin using the system.

9. Traceability Table

Works/ Team Members	**Barış Yıldız**	**Said Çetin**	**Abdussamet Tekin**	**Muzaffer Berke Savaş**	**Mehmet Oğuz Kocadere**
Use Cases Document		X			
Use Case Diagram		X			

Critical Use Case Activity Diagram	X				
GUI for Application Pages	X	X	X		X
Test Case Document				X	
Software Requirements Document	X	X	X	X	X
ER Diagram			X		
Total Effort	24 hours	24 hours	24 hours	24 hours	24 hours

10. Prompts

(1) https://you.com/search?q=I+am+providing+you+with+some+of+the+gui+that+we+developed.+we+are+developing+a+food+delivery+system....&cid=c1_78e18b30-bfb2-45bb-97b6-829273f2346d&tbm=youchat&chatMode=custom

(2) https://chat.deepseek.com/a/chat/s/e7ce64ba-a8eb-4f34-97dc-67da04860faa

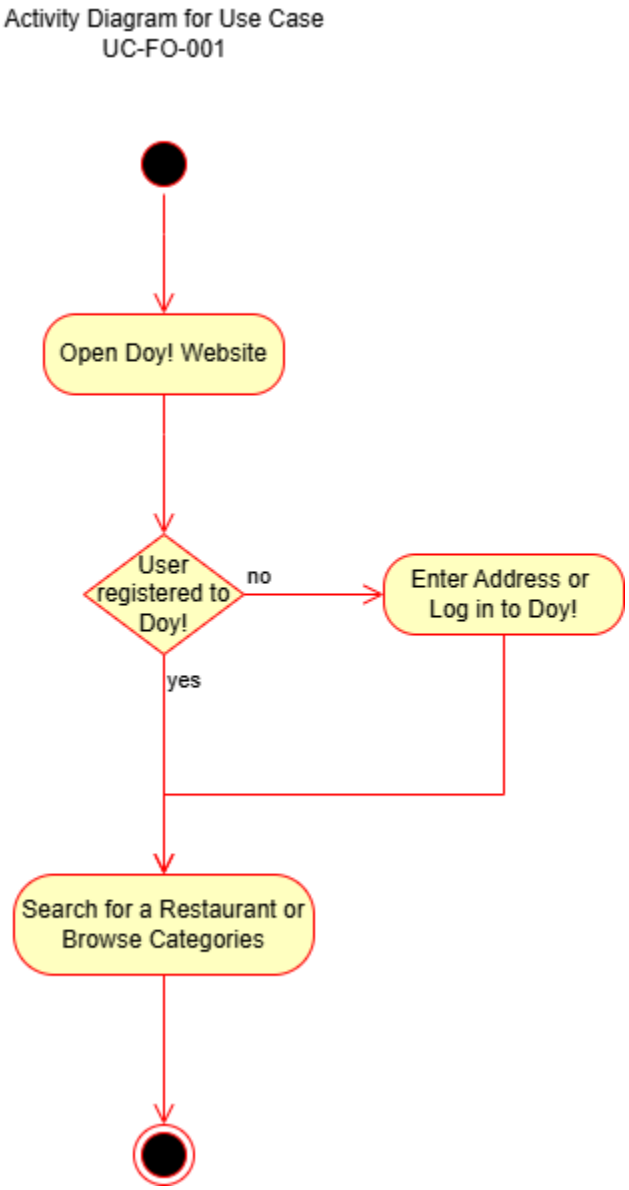
(3) https://grok.com/share/bGVnYWN5_8dde1474-9995-4c71-b358-692860f2633f

SRSgpt.txt

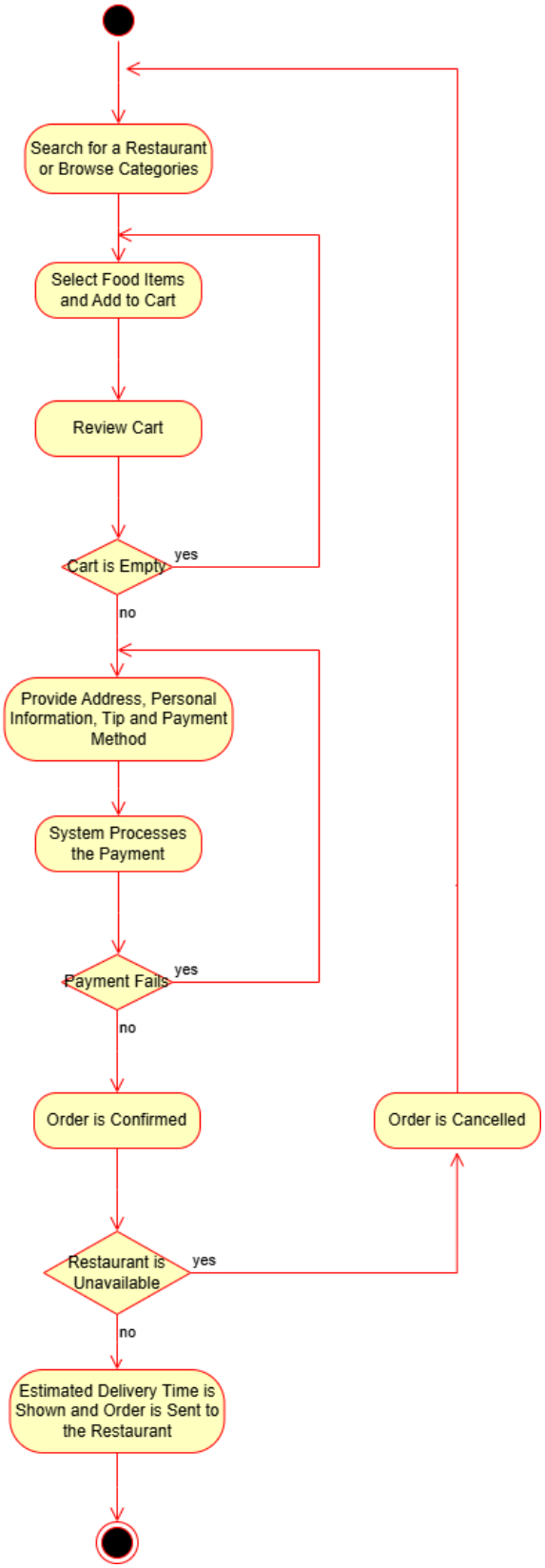
(4)

11. Appendix

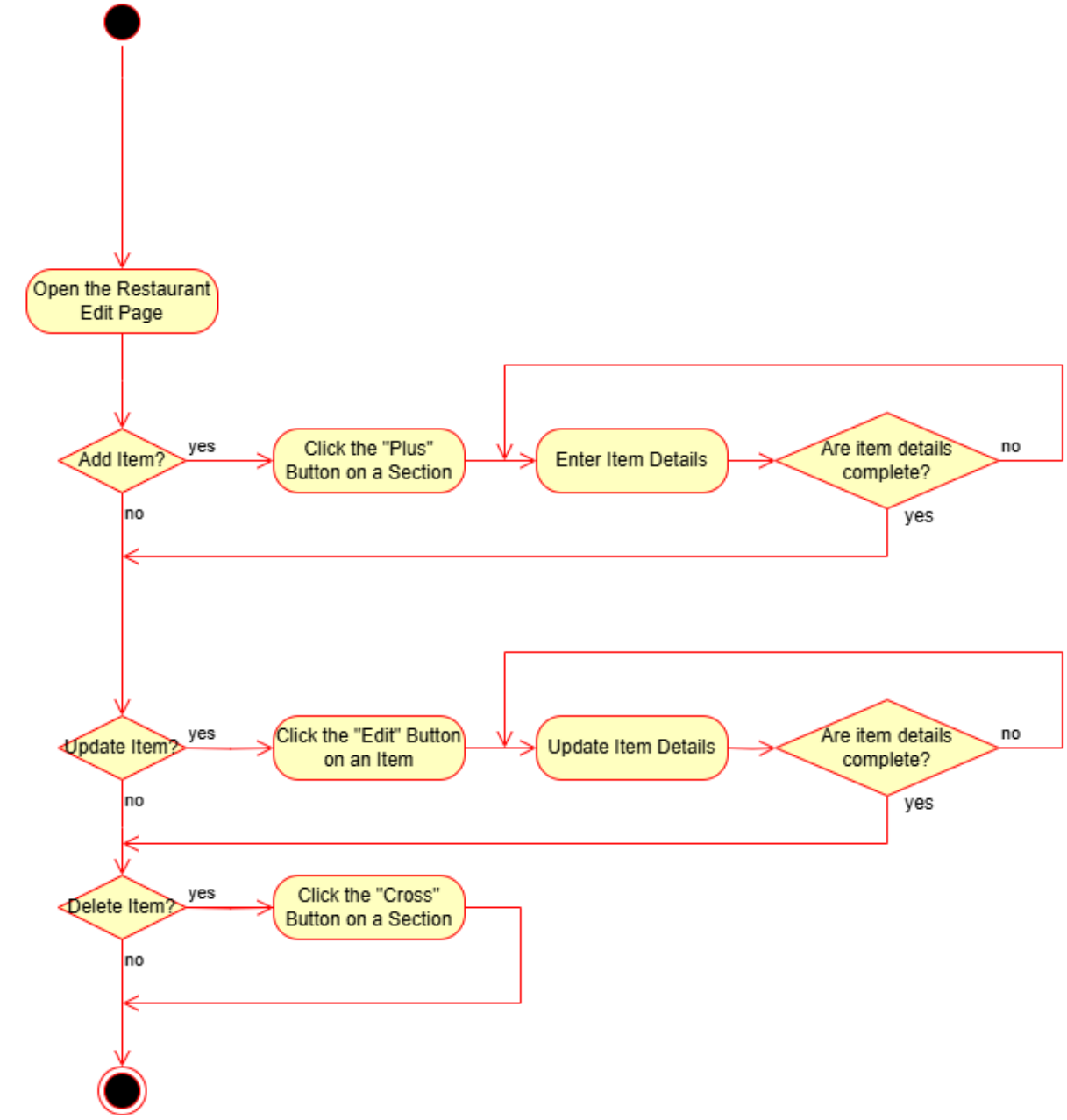
11.1. Activity Diagrams



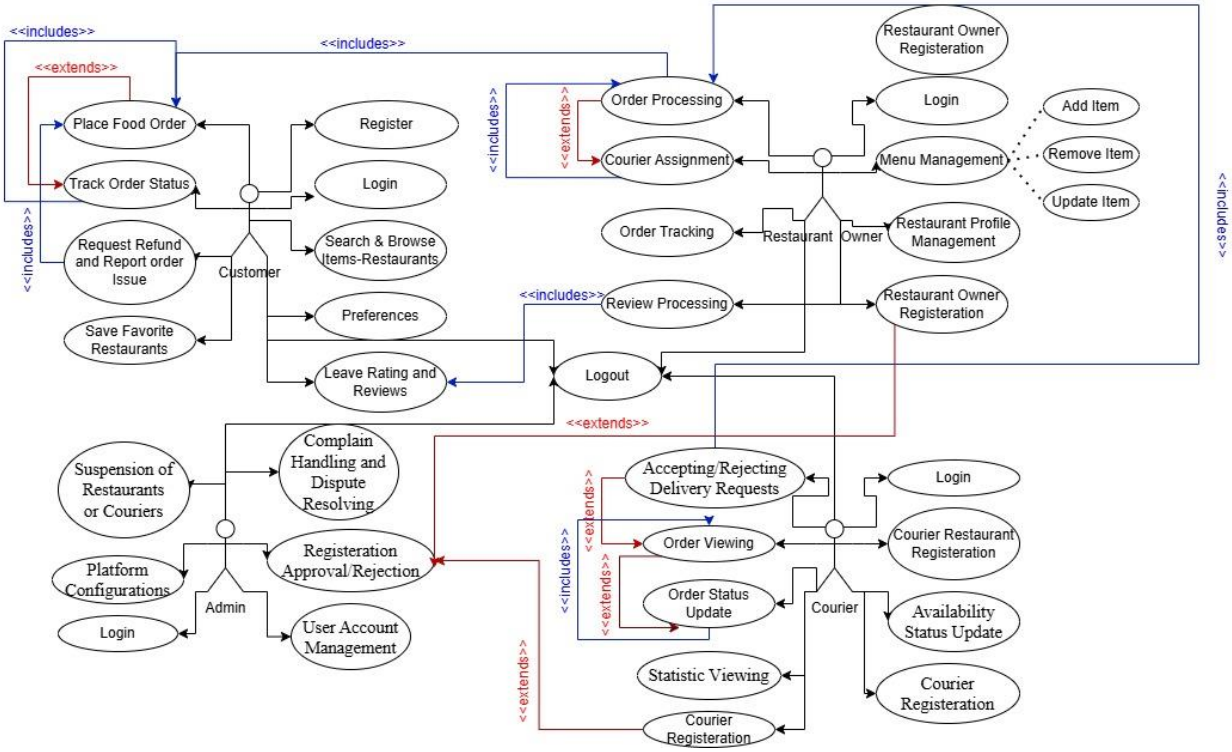
Activity Diagram for Use Case
UC-FO-002



Activity Diagram for Use Case
UC-FO-010



11.2. Use Case Diagram



11.3. ER Diagram