

Software Engineering

B7IS127



07 August 2023

10608881

KASSIM NOMAAN SAYAD

**Contents**

[**Requirements specification** 2](#_Toc142174734)

[**Evidence of Research and Analysis for Life-Cycle Model:** 4](#_Toc142174735)

[**System Architecture/Design Specification:** 5](#_Toc142174736)

[**UML Diagrams:** 6](#_Toc142174737)

# **Requirements specification**

**Functional Requirements:**

1. **User Registration and Authentication:**
   * **Users should be able to create accounts and log in securely.**
   * **User roles should include customers, restaurant owners, delivery staff, and administrators.**
2. **Restaurant Management:**
   * **Restaurant owners should be able to register their restaurants and manage their menu items.**
   * **Each menu item should have attributes like name, description, price, and category.**
3. **Order Placement and Tracking:**
   * **Customers should be able to browse restaurants and their menus.**
   * **Customers should be able to place orders and customize them (e.g., toppings, quantity).**
   * **The system must support real-time order tracking for customers and delivery staff.**
4. **Payment Processing:**
   * **The system should support various payment methods for customers.**
   * **Secure payment processing should be implemented to protect user data.**
5. **Delivery Management:**
   * **Delivery staff should receive real-time order notifications and delivery details.**
   * **Delivery staff should update order statuses (e.g., picked up, delivered) as they progress.**
6. **Reviews and Ratings:**
   * **Customers should be able to rate and leave reviews for restaurants and their items.**
   * **The system should display average ratings and reviews for each restaurant.**

**Non-Functional Requirements:**

1. **Performance:**
   * **The system must handle a high number of concurrent users and orders efficiently.**
   * **Response times for critical operations (e.g., order placement, payment processing) should be low.**
2. **Security:**
   * **User data should be securely stored and encrypted.**
   * **The system should have measures to prevent common web vulnerabilities (e.g., SQL injection, cross-site scripting).**
3. **Scalability:**
   * **The architecture should be scalable to accommodate future growth and increased user load.**

**Stakeholders:**

1. **Customers: Who will be placing orders and using the application.**
2. **Restaurant Owners: Who will be managing their menus and accepting orders.**
3. **Delivery Staff: Who will be responsible for delivering the orders.**
4. **Administrators: Who will oversee the entire system and manage restaurants and users.**

**Constraints:**

1. **Budgetary Constraints: The development budget for the system is limited.**
2. **Time Constraints: The system needs to be developed and deployed within a specific timeframe.**
3. **Technology Constraints: The system must be compatible with existing restaurant POS systems and payment gateways.**

# **Evidence of Research and Analysis for Life-Cycle Model:**

**After thorough research and analysis of the food ordering software development requirements, it is evident that an Agile development life-cycle model, particularly the Scrum framework, is the most appropriate choice. Agile is well-suited for this project due to its following advantages:**

* **Iterative Development: The food ordering software involves continuous feedback and evolving requirements. Agile's iterative approach allows the development team to adapt quickly to changes, ensuring that customer feedback is incorporated in each sprint.**
* **Customer Collaboration: Agile emphasizes continuous collaboration with stakeholders. For a customer-centric application like food ordering software, involving customers throughout the development process is crucial to meeting their needs effectively.**
* **Quick Time-to-Market: The food industry is highly competitive, and early delivery of core functionalities is important. Agile's incremental delivery allows for the software to be deployed in usable increments, providing early value to end-users.**
* **Risk Management: The uncertain nature of software development can lead to unforeseen challenges. Agile mitigates these risks by breaking down the project into smaller, manageable iterations, reducing the impact of potential issues.**
* **Continuous Improvement: Quality is essential in the food ordering software. Agile promotes continuous improvement through regular testing and feedback, resulting in a high-quality product.**

**Considering these factors, Agile, particularly the Scrum framework, provides the flexibility, adaptability, and customer focus needed for the successful development of the food ordering software.**

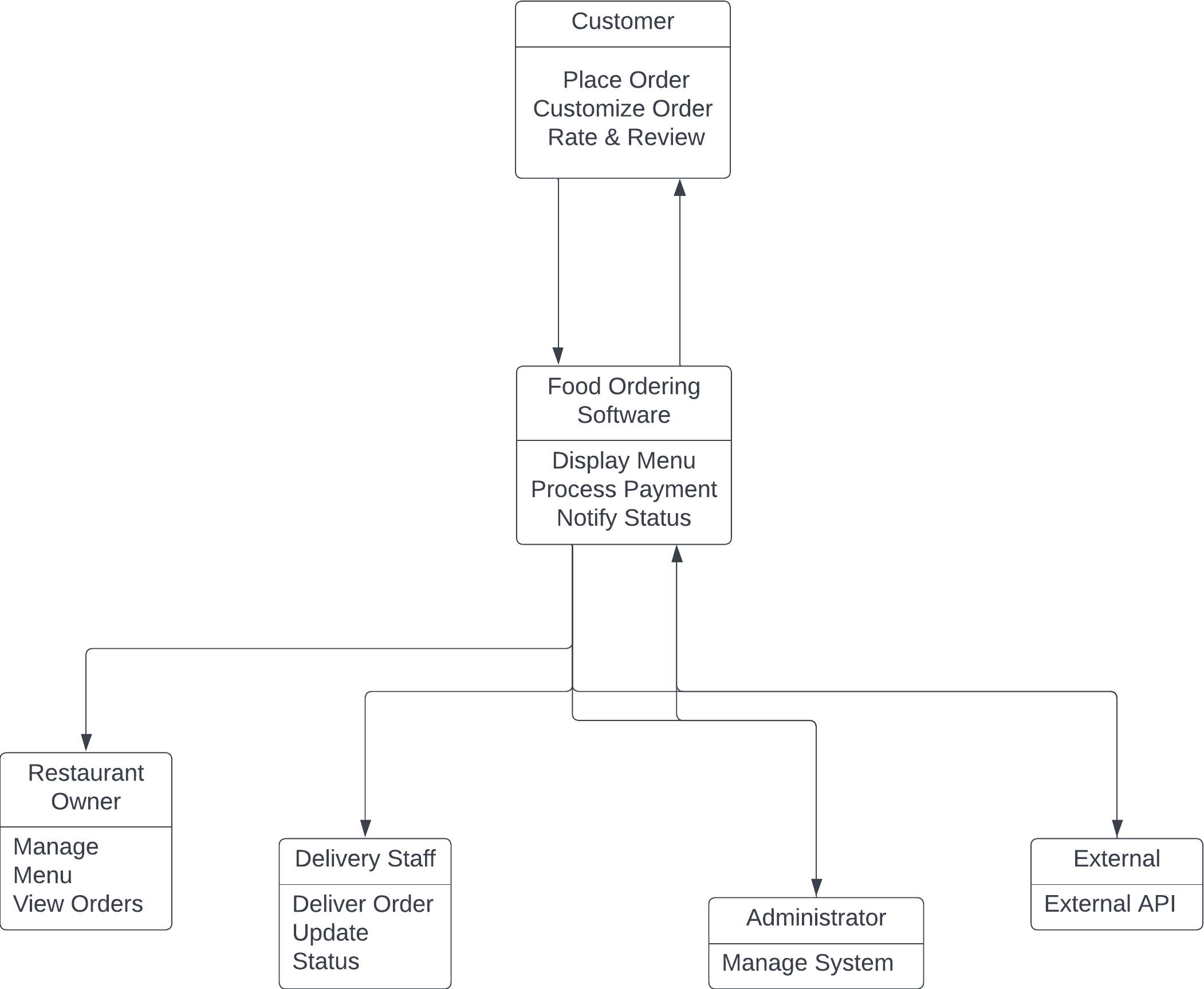
# **System Architecture/Design Specification:**

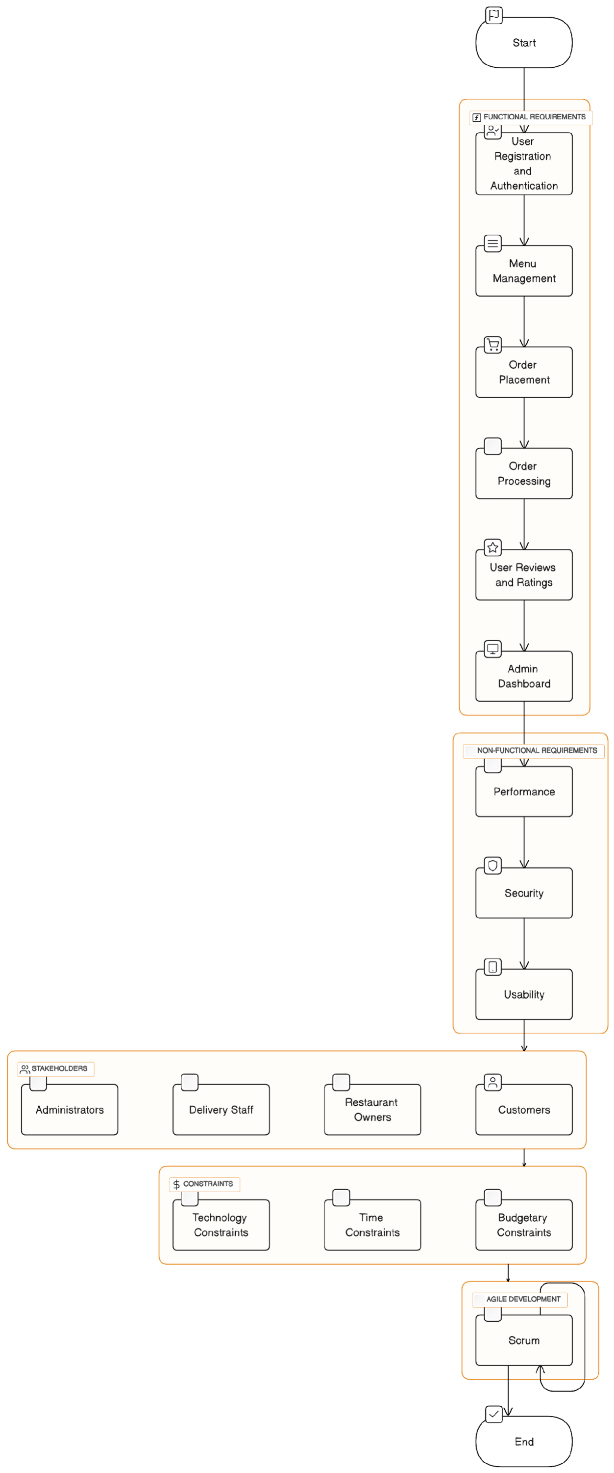
**The system architecture of the food ordering software will follow a microservices architecture, which allows for a modular and scalable system. The architecture will consist of the following main components:**

1. **User Management Microservice: Responsible for user registration, authentication, and managing user roles.**
2. **Restaurant Management Microservice: Handles restaurant registration, menu management, and reviews/ratings.**
3. **Order Management Microservice: Manages order placement, tracking, and payment processing.**
4. **Delivery Management Microservice: Facilitates delivery staff operations, order notifications, and status updates.**
5. **Review/Rating Microservice: Handles customer reviews and restaurant ratings.**

# **UML Diagrams:**

1. **Use Case Diagram:**

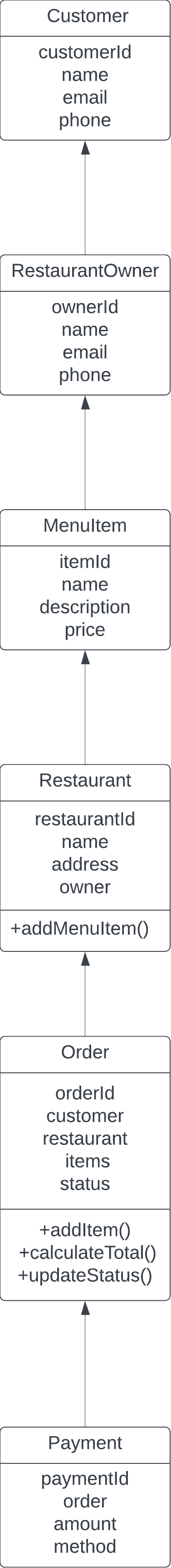




1. **Flow Chart:**
2. **Sequence Diagram:**

**A screenshot of a computer

Description automatically generated**

1. **Class Diagram**