### Software Requirements Specification

#### For

**Flower Management System**

**Version 1.0**  
Prepared by:

* Nimil Zubair (22K-4617)
* Alishba Hassan (22K-4333)
* Waniya Badar (22K-4526)

FAST National University of Computer and Emerging Sciences  
**[5th November 2024]**

### Table of Contents

1. **Introduction**  
   1.1 Purpose  
   1.2 Document Conventions  
   1.3 Intended Audience and Reading Suggestions  
   1.4 Product Scope  
   1.5 References
2. **Overall Description**  
   2.1 Product Perspective  
   2.2 Product Functions  
   2.3 User Classes and Characteristics  
   2.4 Operating Environment  
   2.5 Design and Implementation Constraints  
   2.6 User Documentation  
   2.7 Assumptions and Dependencies  
   2.8 Process Model  
   2.9 Project Plan  
   2.10 Feasibility Report  
   2.11 Homogenization Process  
   2.12 Use Case Descriptions
3. **External Interface Requirements**  
   3.1 User Interfaces  
   3.2 Hardware Interfaces  
   3.3 Software Interfaces  
   3.4 Communication Interfaces
4. **System Features**  
   4.1 Flower Inventory Management  
   4.2 Order Placement and Tracking  
   4.3 Customer Engagement and Browsing
5. **Other Nonfunctional Requirements**  
   5.1 Performance Requirements  
   5.2 Safety Requirements  
   5.3 Security Requirements  
   5.4 Software Quality Attributes  
   5.5 Business Rules
6. **Other Requirements**
7. **Appendices**  
   A: Glossary  
   B: Analysis Models  
   C: To Be Determined List

### 1. Introduction

#### 1.1 Purpose

The purpose of this document is to outline the functional and non-functional requirements for the **Flower Management System (FMS)**, a software solution designed to manage a flower shop’s operations efficiently. It serves as a guide for developers, project managers, and stakeholders throughout the system's development and deployment.

#### 1.2 Document Conventions

* Functional requirements are labeled as **REQ-FMS-#**.
* Priorities are marked as High (H), Medium (M), or Low (L).
* Technical terms and acronyms are defined in the glossary.

#### 1.3 Intended Audience and Reading Suggestions

This document is intended for developers, testers, project managers, and stakeholders. Developers and testers will benefit from detailed requirements in Sections 3 and 4. Stakeholders can focus on Sections 1 and 2 for an overview.

#### 1.4 Product Scope

The **FMS** is a web-based platform designed to automate and streamline flower shop operations. It provides functionalities for inventory management, supplier coordination, order placement, customer browsing, and sales analytics.

#### 1.5 References

1. Flower Shop Website Project Proposal, 2024.
2. IEEE Software Requirements Specification Guidelines.

### 2. Overall Description

#### 2.1 Product Perspective

The **FMS** is a standalone web application that integrates front-end and back-end services. It can be extended to support mobile platforms in the future.

#### 2.2 Product Functions

* Manage flower inventory, including stock levels and categories.
* Enable customers to browse and order flowers online.
* Process and track customer orders.
* Coordinate with suppliers for restocking.
* Generate sales and performance reports.

#### 2.3 User Classes and Characteristics

* **Administrators:** Manage the system, inventory, and supplier coordination.
* **Shop Staff:** Handle order fulfillment and customer queries.
* **Customers:** Browse flowers, place orders, and track their purchases.

#### 2.4 Operating Environment

* Compatible with modern web browsers (Chrome, Firefox, Safari, etc.).
* Accessible on desktops, tablets, and smartphones.

#### 2.5 Design and Implementation Constraints

* Must integrate with secure third-party payment gateways.
* Compliance with data protection laws for user data.

#### 2.6 User Documentation

* User manuals for administrators and staff.
* Online tutorials and FAQs for customers.

#### 2.7 Assumptions and Dependencies

* Reliable internet connection for all users.
* Accurate initial inventory data provided by the shop.

### 2.8 Process Model (Strategy for Software Development)

For the development of the Flower Management System (FMS), we will adopt the **Agile methodology**. This approach allows iterative development and frequent reassessments of project progress to meet dynamic requirements effectively.

**Key aspects of our process model include:**

* **Sprint-based development:** Dividing the project into 2-week sprints.
* **Daily stand-ups:** Ensuring clear communication among team members.
* **Backlog prioritization:** Organizing tasks based on urgency and importance, focusing on customer engagement features first.
* **Regular feedback loops:** Gathering feedback from stakeholders at the end of each sprint.
* **Testing integration:** Conducting both manual and automated testing during each iteration to maintain system reliability.

### 2.9 Project Plan

The FMS project plan follows a structured timeline:

1. **Requirement Gathering (Weeks 1-2):** Interviewing stakeholders and documenting software requirements.
2. **Design Phase (Weeks 3-5):** Creating wireframes, use case diagrams, and database schemas.
3. **Development Phase (Weeks 6-10):** Implementing the front-end, back-end, and database, prioritizing flower inventory and order management features.
4. **Testing Phase (Weeks 11-12):** Conducting unit tests, integration tests, and user acceptance testing (UAT).
5. **Deployment and Training (Weeks 13-14):** Deploying the system on live servers and training shop staff on its usage.
6. **Post-launch Support (Ongoing):** Addressing bugs and implementing minor updates.

### 2.10 Feasibility Report

**Technical Feasibility:**

* **Tools and Technologies:** The project utilizes widely adopted tools like MySQL, XAMPP, and modern web development frameworks, ensuring a robust system architecture.
* **Skillset:** The development team is experienced in web and database development, reducing technical risks.

**Economic Feasibility:**

* **Cost Estimate:** The initial development and deployment cost is moderate, with most expenses in developer hours and server hosting. Long-term savings are anticipated through automation.

**Operational Feasibility:**

* The system will significantly enhance flower shop operations by automating inventory management and customer engagement, improving efficiency and customer satisfaction.

### 2.11 Homogenization Process

The homogenization process in the context of the FMS ensures consistent data formats and operations across different system modules:

1. **Database Schema Standardization:** Defining uniform database structures for flower inventory, orders, and customer records.
2. **Data Validation:** Implementing strict validation checks for all user inputs to maintain data integrity.
3. **API Integration:** Standardizing API calls to maintain seamless communication between the front-end and back-end.
4. **User Interface Consistency:** Using a single design language to ensure a cohesive user experience.

### 2.12 Use Case Descriptions in Textual Format

#### Use Case: ****Add Flower to Inventory****

* **Actor:** Administrator
* **Preconditions:** The user must be logged into the administrator account.
* **Basic Flow:**
  1. The administrator navigates to the "Add Flower" section.
  2. Enters details such as flower name, category, stock level, and price.
  3. Submits the form.
  4. The system validates and saves the details into the database.
  5. A success message is displayed.
* **Post conditions:** The new flower is added to the inventory list.
* **Alternate Flows:** If validation fails, the system displays error messages.

#### Use Case: ****Place Customer Order****

* **Actor:** Customer
* **Preconditions:** The customer must be logged in.
* **Basic Flow:**
  1. The customer browses the catalog and selects flowers.
  2. Adds items to the cart.
  3. Proceeds to checkout, enters shipping details, and confirms payment.
  4. The system processes the order and sends a confirmation email.
* **Postconditions:** The order is logged in the database for processing.
* **Alternate Flows:** If payment fails, the system notifies the user and retains the cart.

### 3. External Interface Requirements

#### 3.1 User Interfaces

* **Dashboard:** A comprehensive overview of inventory, orders, and sales.
* **Customer Portal:** Allows browsing, searching, and ordering flowers.
* **Supplier Module:** Facilitates order requests and updates.

#### 3.2 Hardware Interfaces

* Supports standard devices like PCs, tablets, and smartphones.

#### 3.3 Software Interfaces

* Integrates with MYSQL , XAMP, for database operations.
* API-based communication with payment gateways.

#### 3.4 Communication Interfaces

* HTTPS for secure data transmission.
* Email notifications for order updates.

### 4. System Features

#### 4.1 Flower Inventory Management

**Description and Priority:** Enables tracking of flower categories. Priority: High.

**Functional Requirements:**

* REQ-FMS-1.1: The system shall allow administrators to add, update, and delete flower records.

#### 4.2 Order Placement and Tracking

**Description and Priority:** Allows customers to place. Priority: High.

**Functional Requirements:**

* REQ-FMS-2.1: Customers can browse the catalog and add items to their cart.
* REQ-FMS-2.2: The system shall send order confirmation emails.

#### 4.3 Customer Engagement and Browsing

**Description and Priority:** Enhances customer experience with an intuitive interface. Priority: High.

**Functional Requirements:**

* REQ-FMS-4.1: Customers can filter flowers by type, price, or occasion.

### 5. Other Nonfunctional Requirements

#### 5.1 Performance Requirements

* The system shall support at least 100 simultaneous users without performance degradation.

#### 5.2 Safety Requirements

* Data backups will be scheduled daily to prevent data loss.

#### 5.3 Security Requirements

* All sensitive user data, including payment details, shall be encrypted.

#### 5.4 Software Quality Attributes

* Usability: The system will prioritize ease of navigation and search functionality.
* Maintainability: Modular code design for future feature expansion.

#### 5.5 Business Rules

* Only registered suppliers can fulfill restocking requests.

### 6. Other Requirements

* Support for multiple languages in the future.

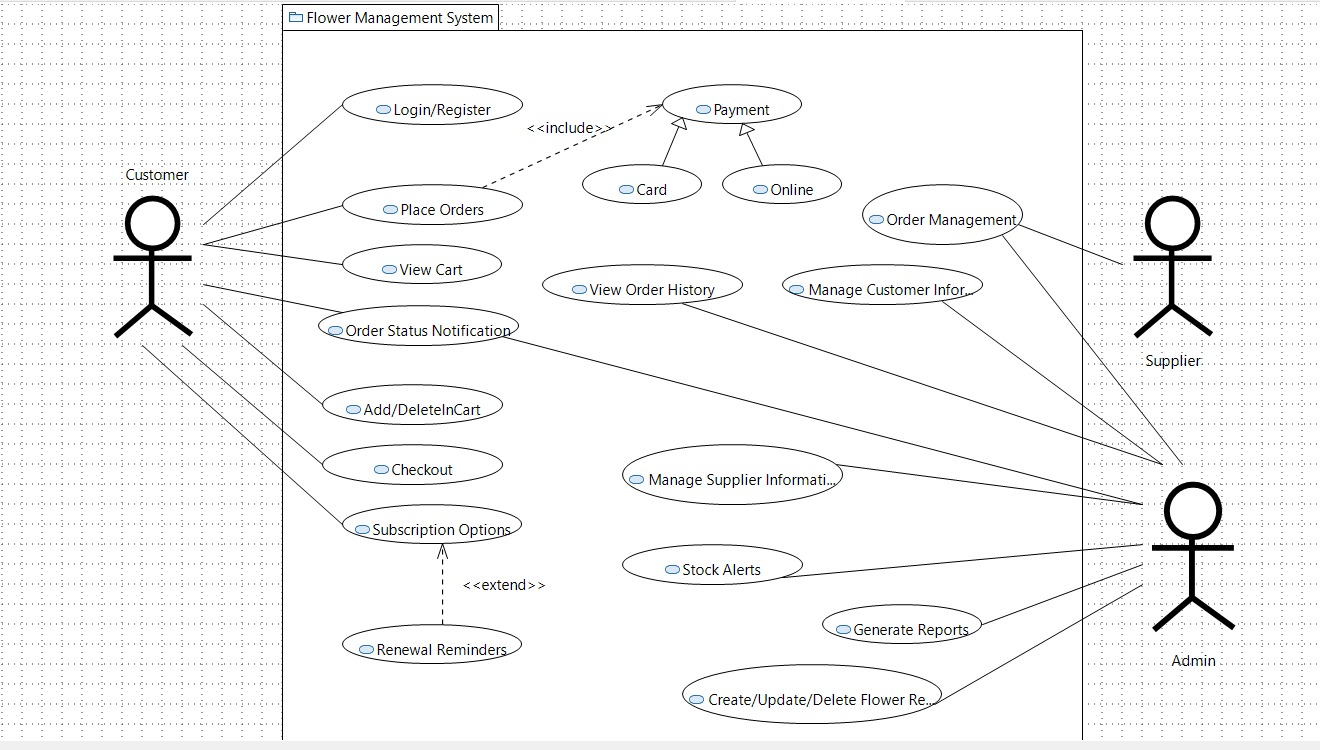
### Appendices

**Appendix A: Glossary**

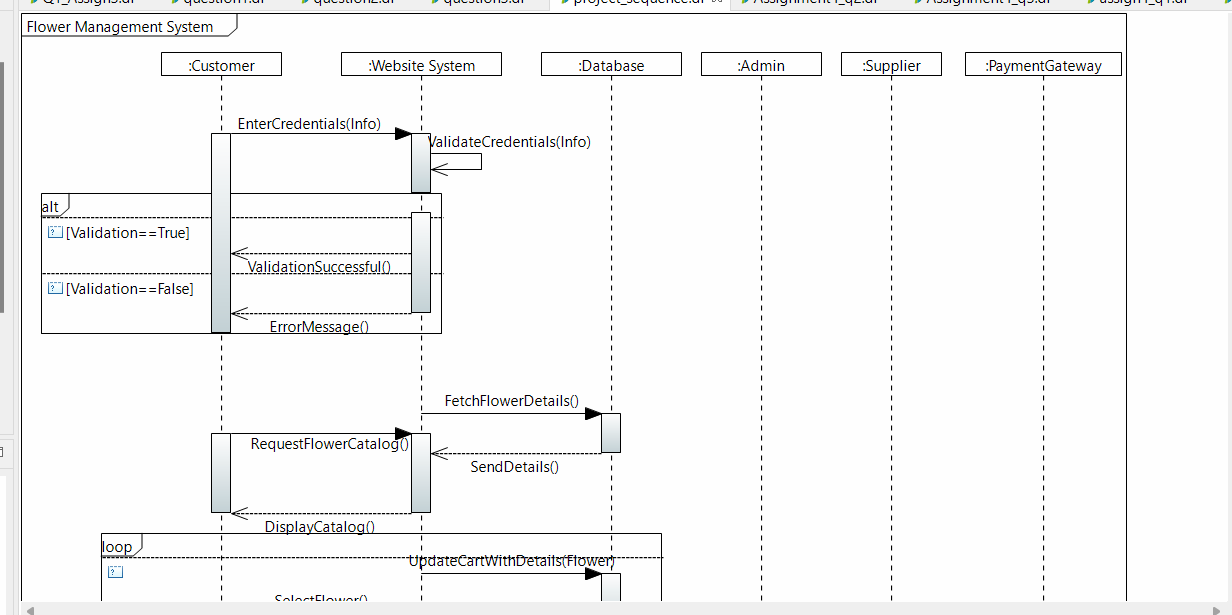
* **FMS:** Flower Management System

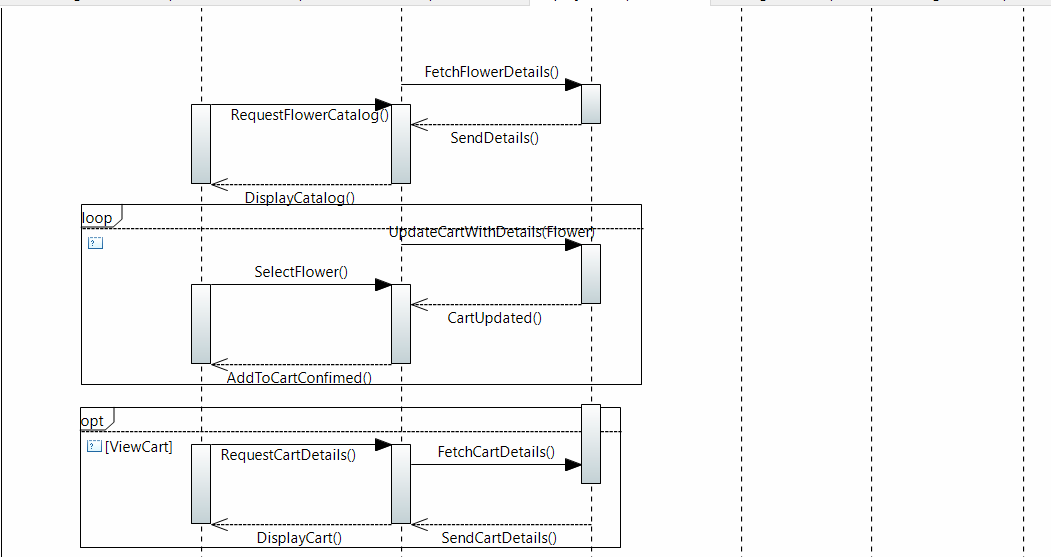
**Appendix B: Analysis Models**

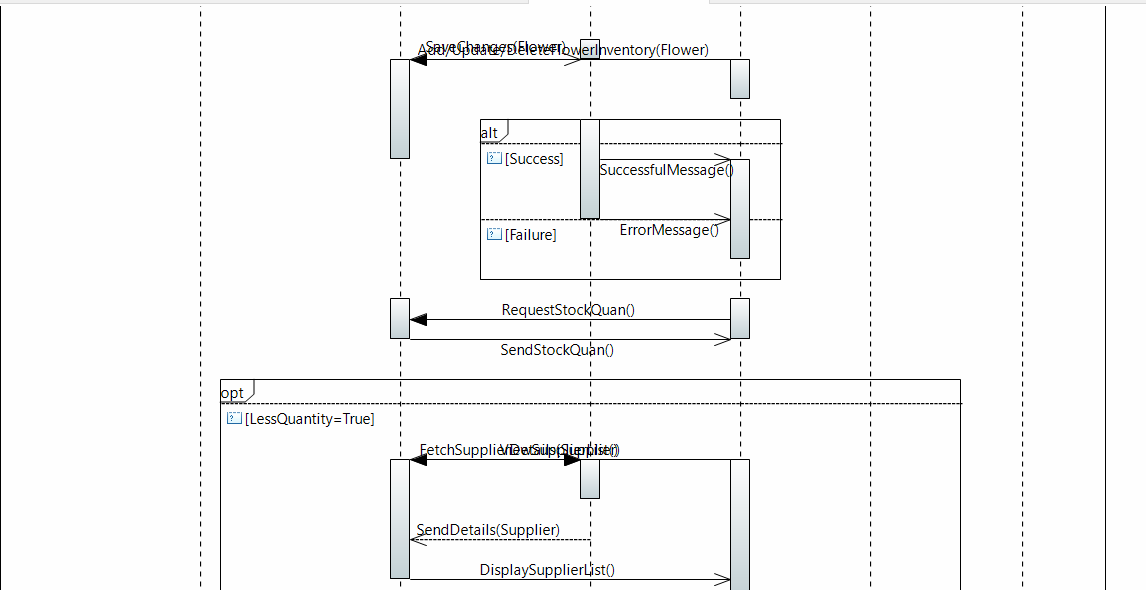
* **Use Case diagram:**

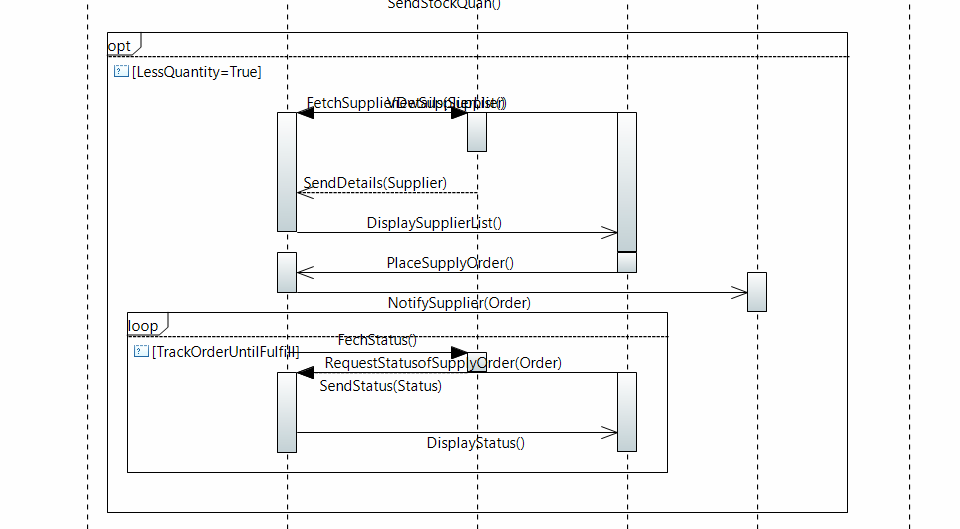


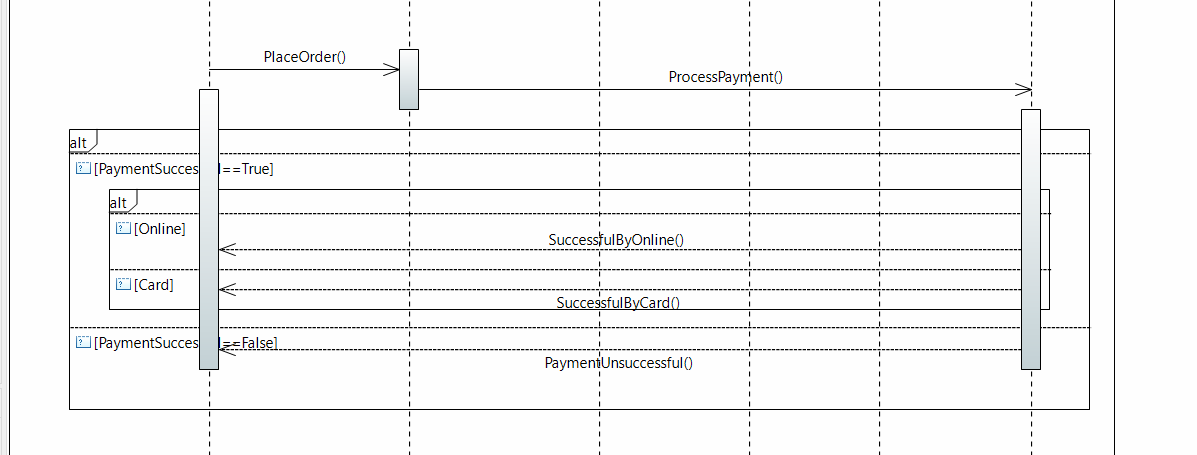
* **Sequence diagram:**

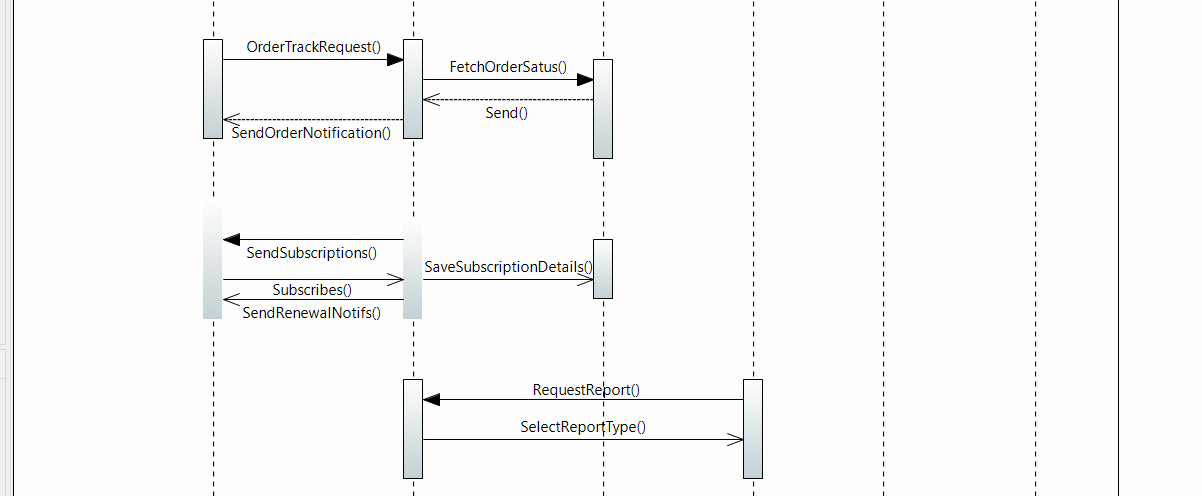
****



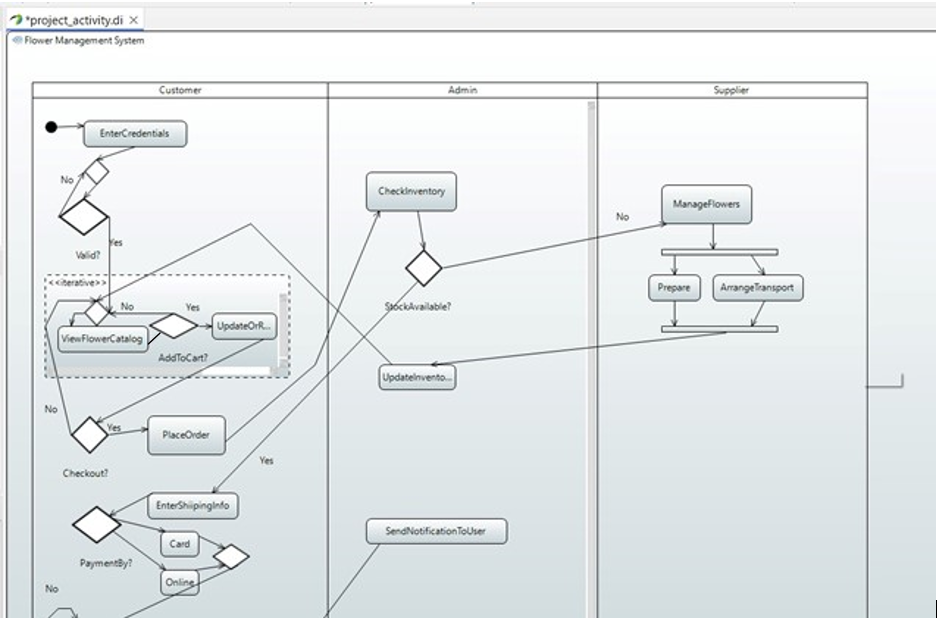


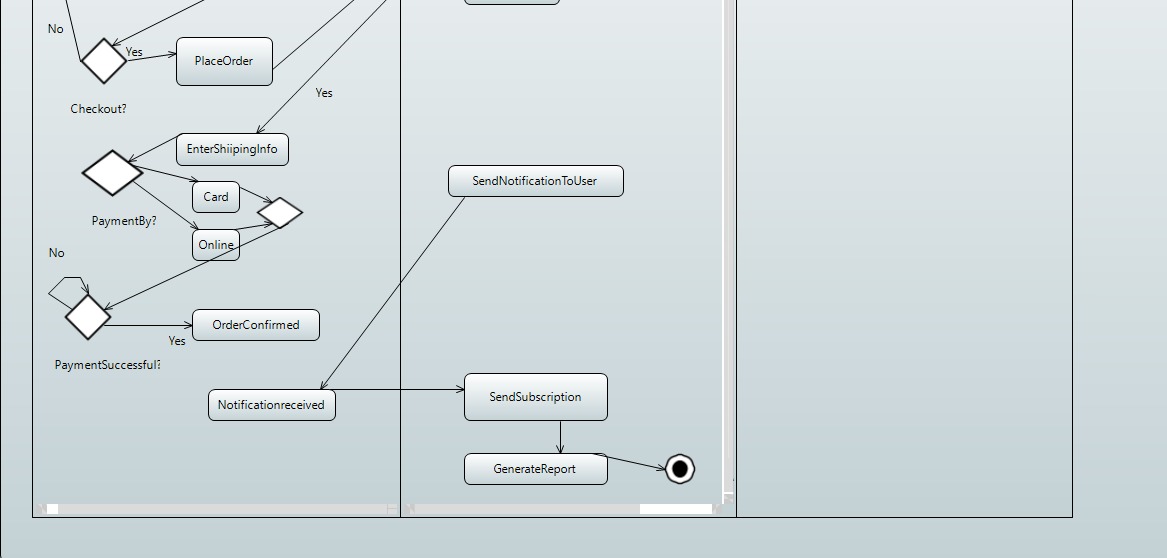


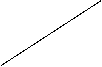




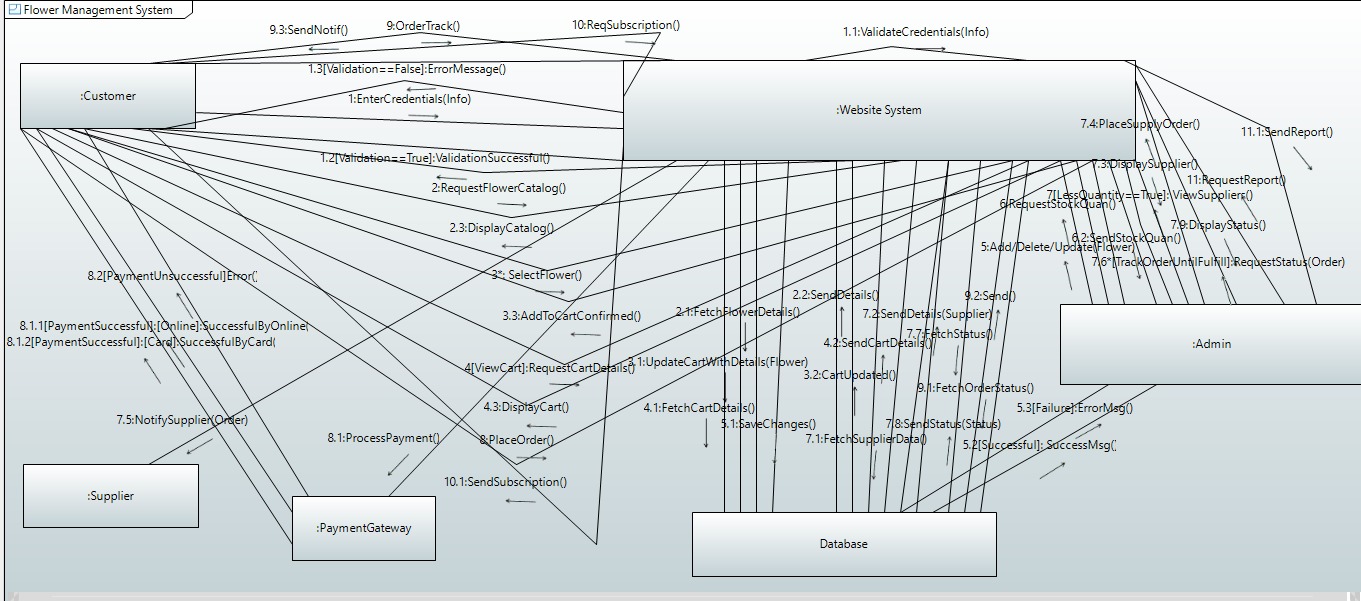
* **Activity diagram:**



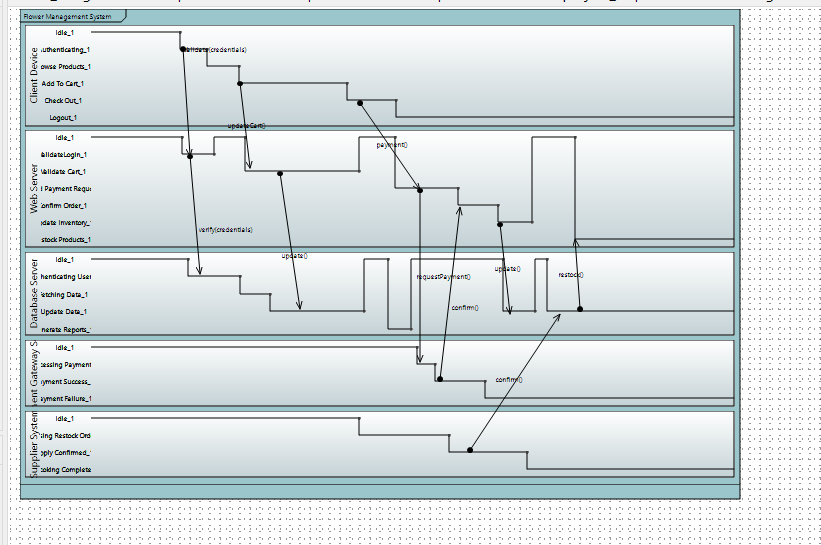




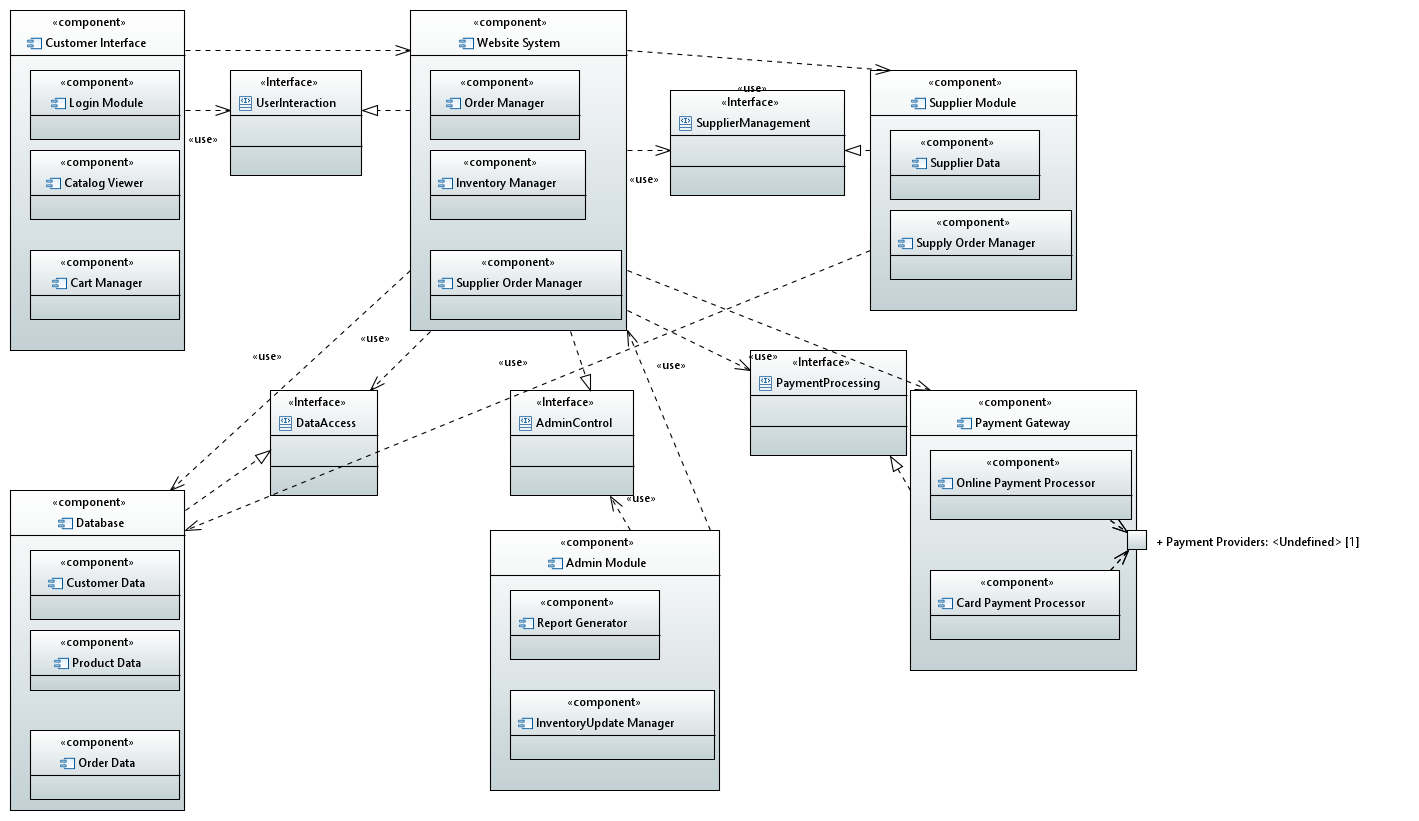
* **Collaboration diagram:**



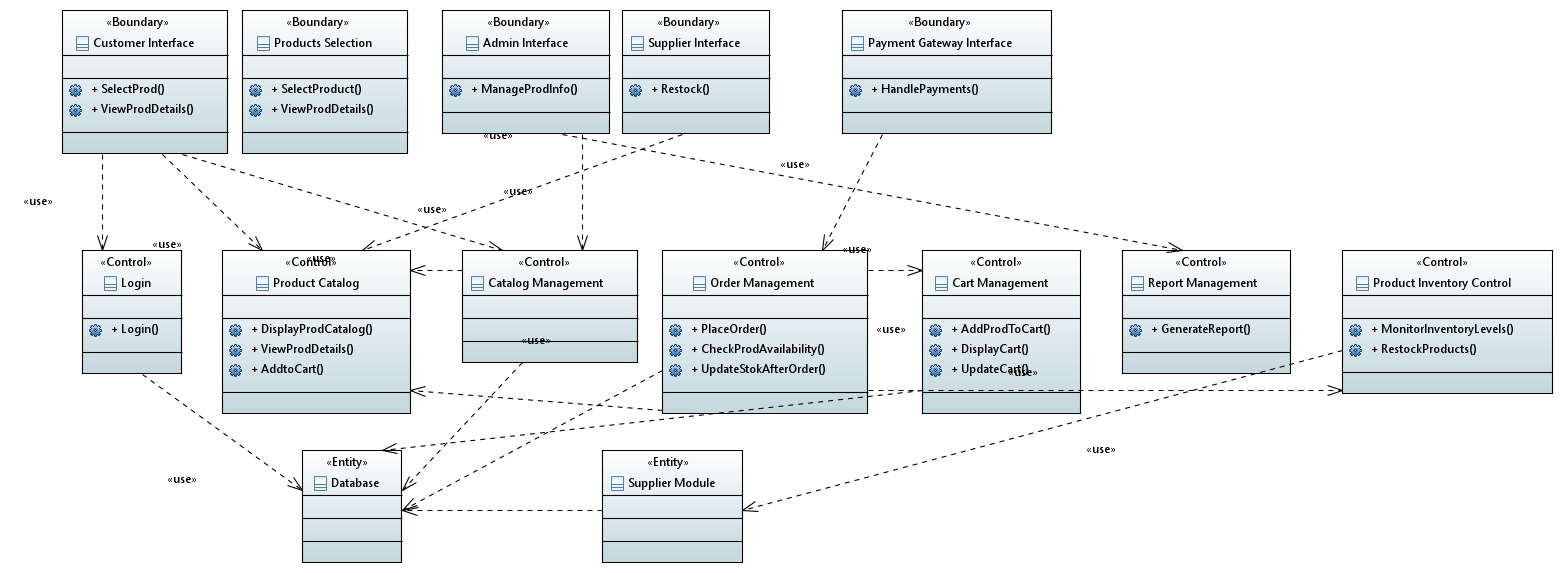
* **Timing diagram:**



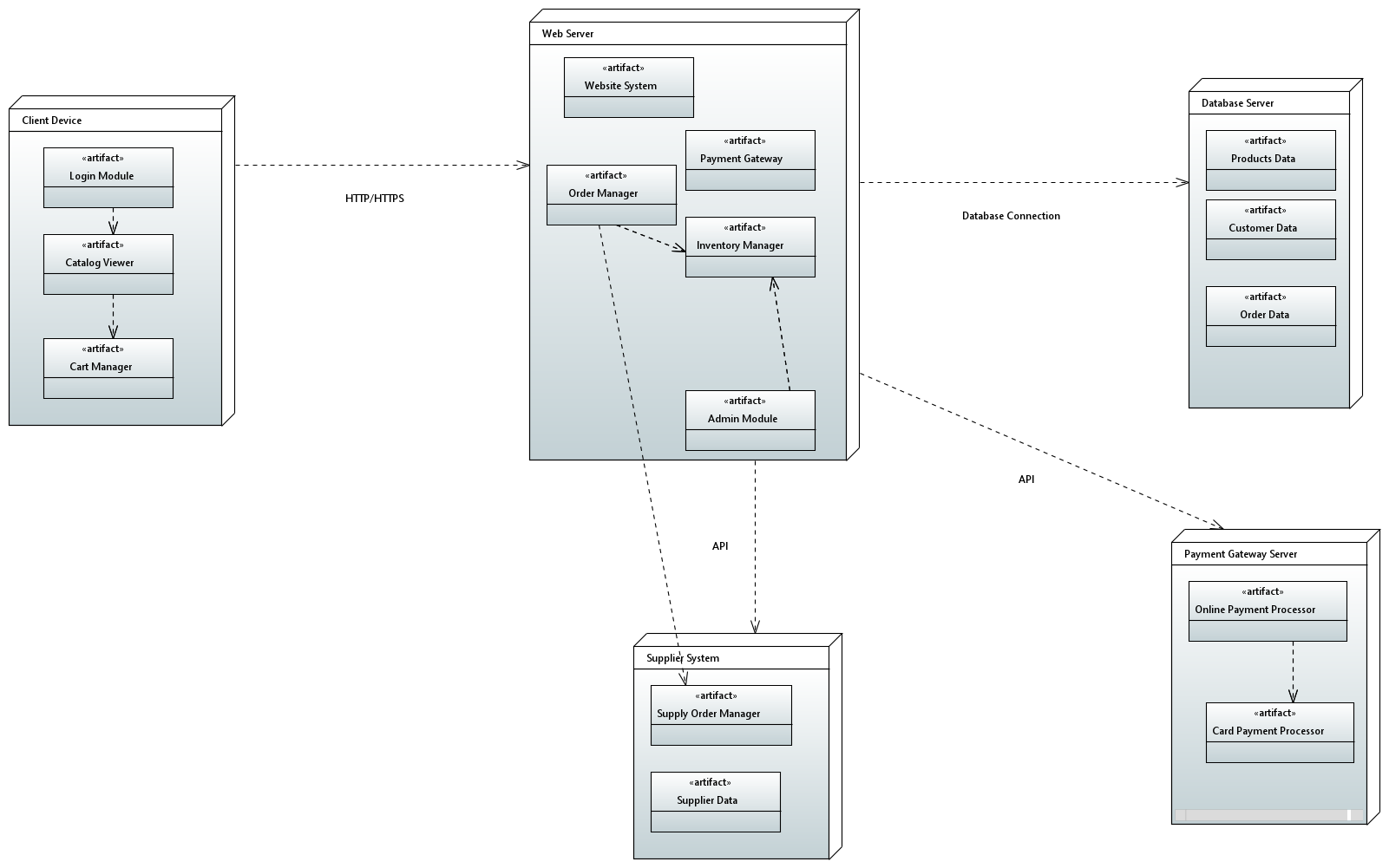
* **Component diagram:**



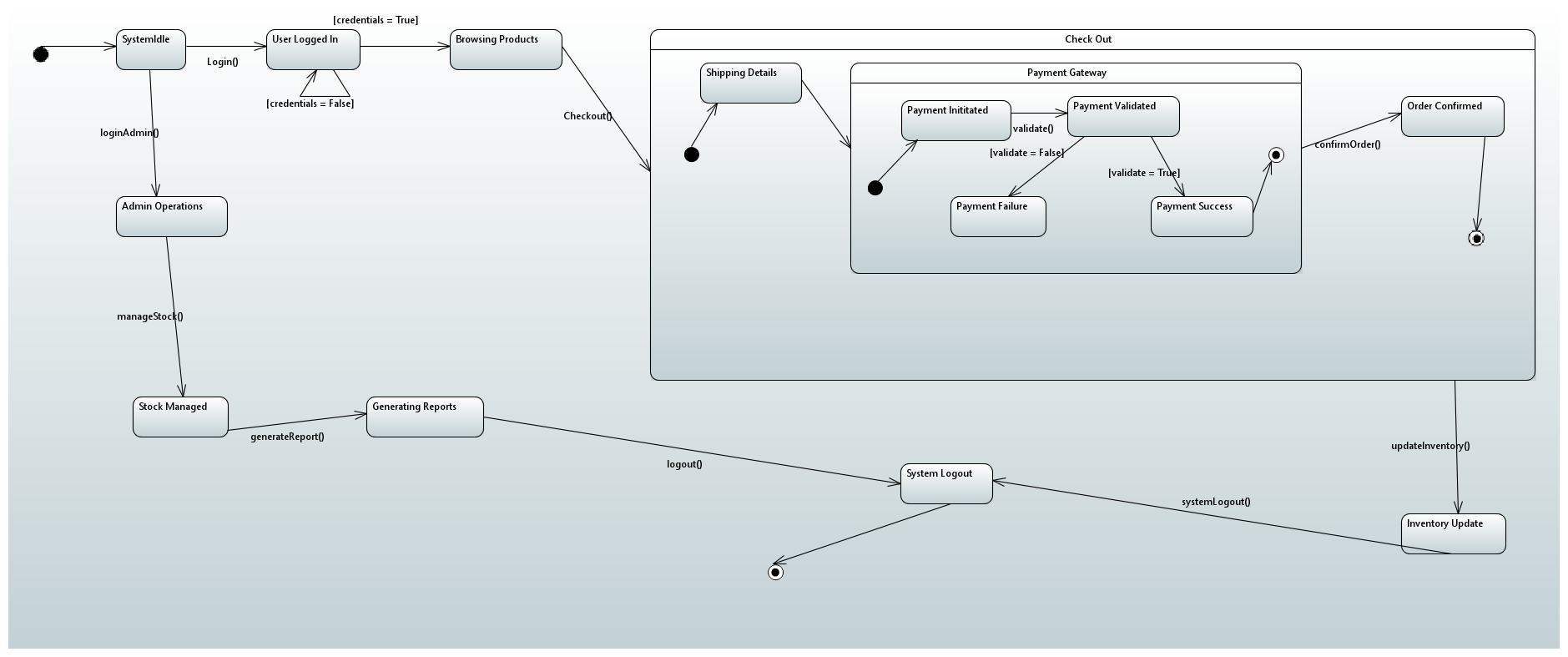
* **ECB diagram:**



* **Deployment diagram:**



* **State diagram:**



**Appendix C: To Be Determined List**

* Integration with social media platforms for marketing.