E-Shopper Project

CS-425 Software Engineering Team 4:
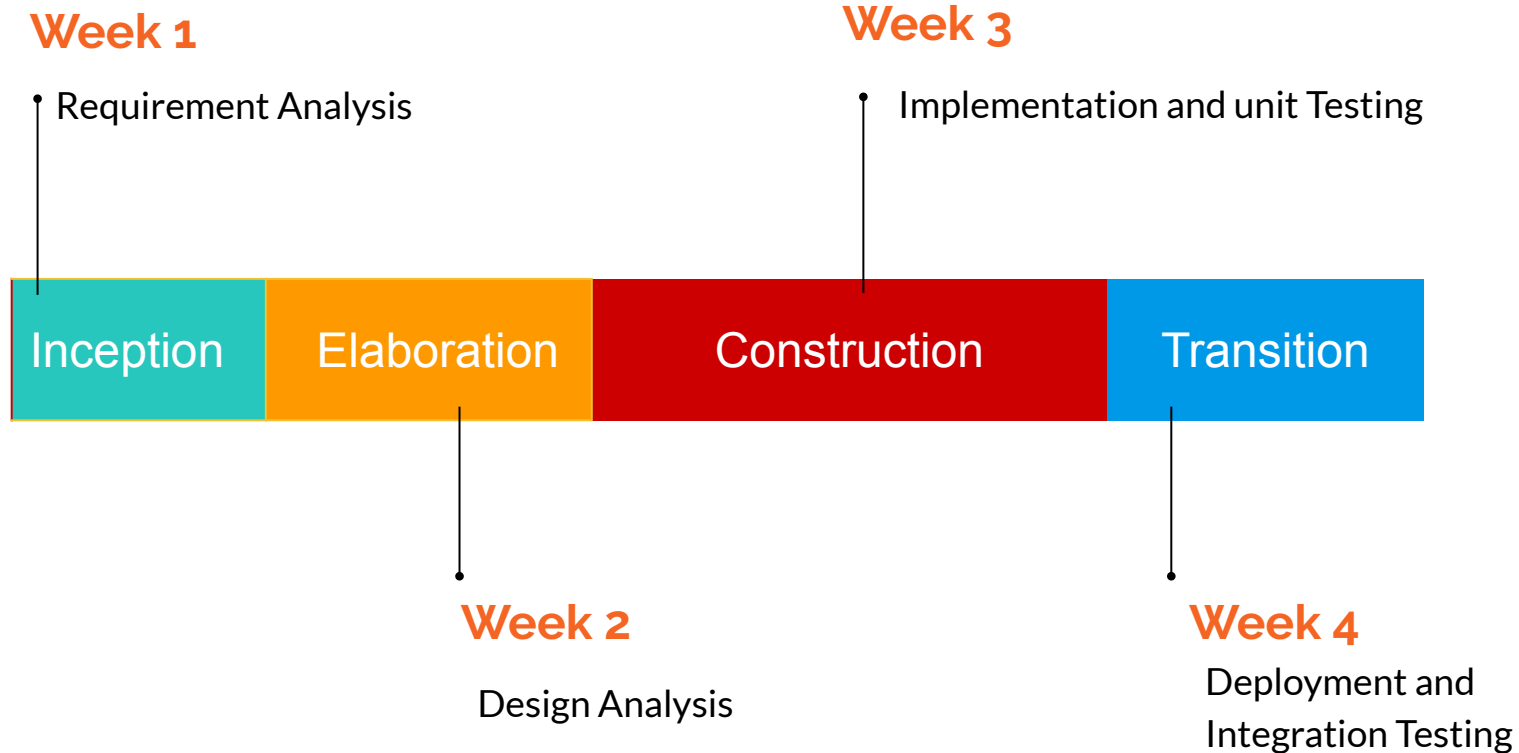Adeleke Oyebode Adeola   610516
Amit Bhattarai   610514
Ny Aina Michael Andriantsoa   610574
Lamhot Siagian   610517

Outline

- ➜ Timeline
- ➜ Software Development Process Methodology
- ➜ Key Abstraction
- ➜ System Architecture Diagram
- ➜ Use Case Diagram
- ➜ Use Case Description
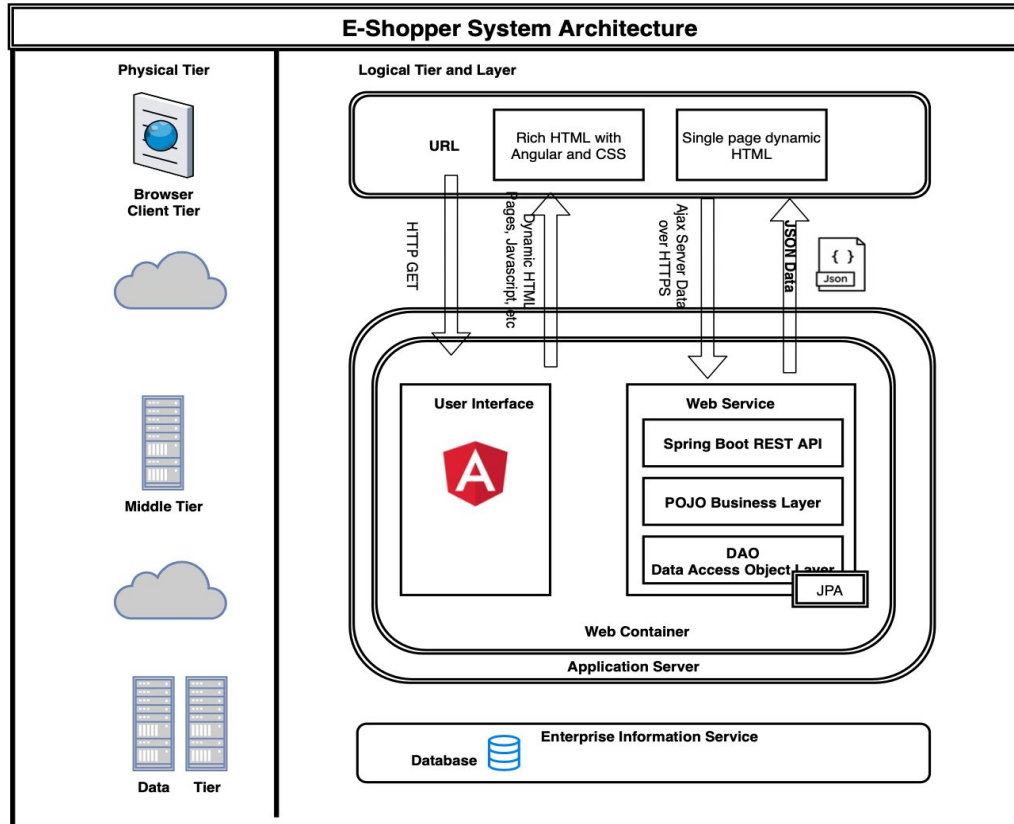- ➜ Sequence Diagram
- ➜ Collaboration and VOPC

Timeline

Week 1

Requirement Analysis

Week 3

Implementation and unit Testing

Inception	Elaboration	Construction	Transition

Week 2

Design Analysis

Week 4

Deployment and Integration Testing

Software Development Process :Agile

Using Trello

Daily update

Iterative Process

System Architecture Diagram

E-Shopper System Architecture

Physical Tier

Logical Tier and Layer

URL — Rich HTML with Angular and CSS / Single page dynamic HTML

Browser Client Tier

HTTP GET / Dynamic HTML Pages, Javascript, etc / Ajax Server Data over HTTPS / JSON Data — Json

Middle Tier

User Interface / Web Service
- Spring Boot REST API
- POJO Business Layer
- DAO Data Access Object Layer — JPA

Web Container

Application Server

Data Tier

Enterprise Information Service

Database

Technologies

- Angular
- Mysql
- Spring Boot
- Bootstrap

5

Use Case Diagram

Use Case Description

| Use Case Number: 8		
Name	Place Order	
Brief description	This use case provides a functionality for checkout after adding product to cart	
Actors	Buyer	
Preconditions		
Buyer must login to system		
Flows of Events:		
1. Basic Flows		
1.0 Place Order		
Step	**User Actions**	**System Response**
1	Buyer view product detail	1.1. System displaying product detail
2	Buyer input quantity and click "Add to Cart" button	2.1. System checking stock 2.2. System store to database
3	Buyer click "Checkout" button	3.1. System verify quantity before reducing stock 3.2. Check availability (Status) of the product 3.3. Reduce product quantity and update in database 3.4. System display success message, order successfully 3.5. System will be rendered to order detail
Postconditions		
Product stock will be reduced and render to order details.		
Business Rules		
Stock can't be minus Product Order Quantity should be validated Non available product can't be bought (products status)		

Key Abstraction

➜ **Buyer**

➜ **Seller / Merchant**

➜ **Product**

➜ **Category**

➜ **Order**

➜ **Order Item**

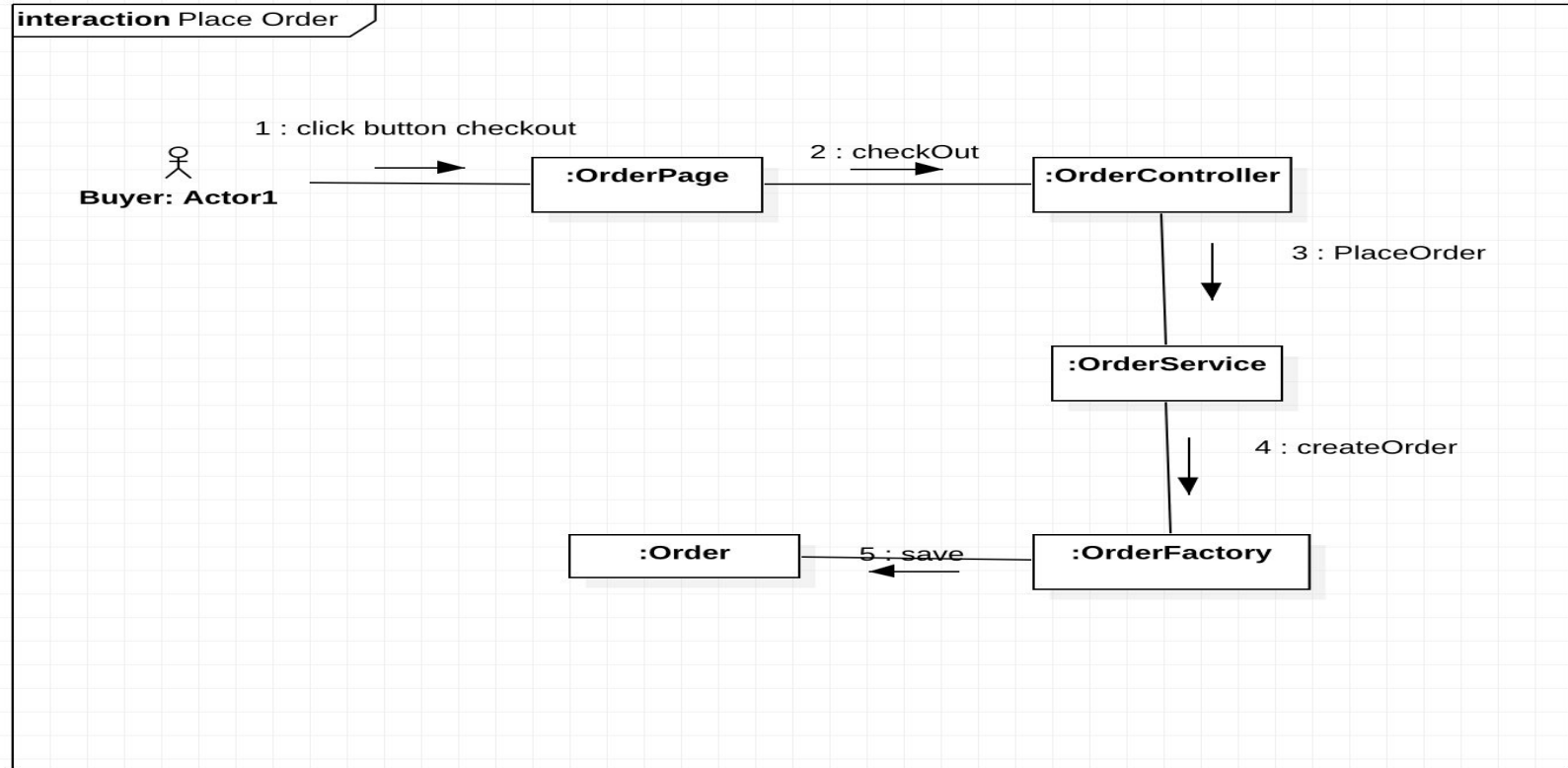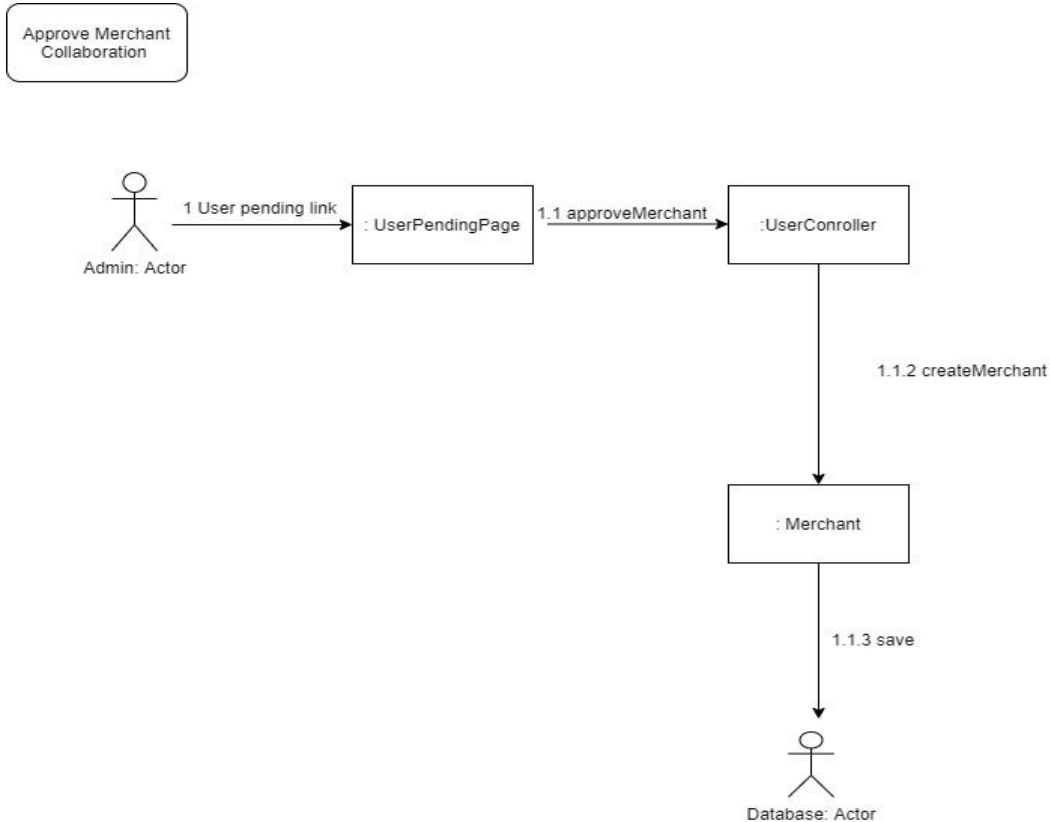# Place Order Sequence
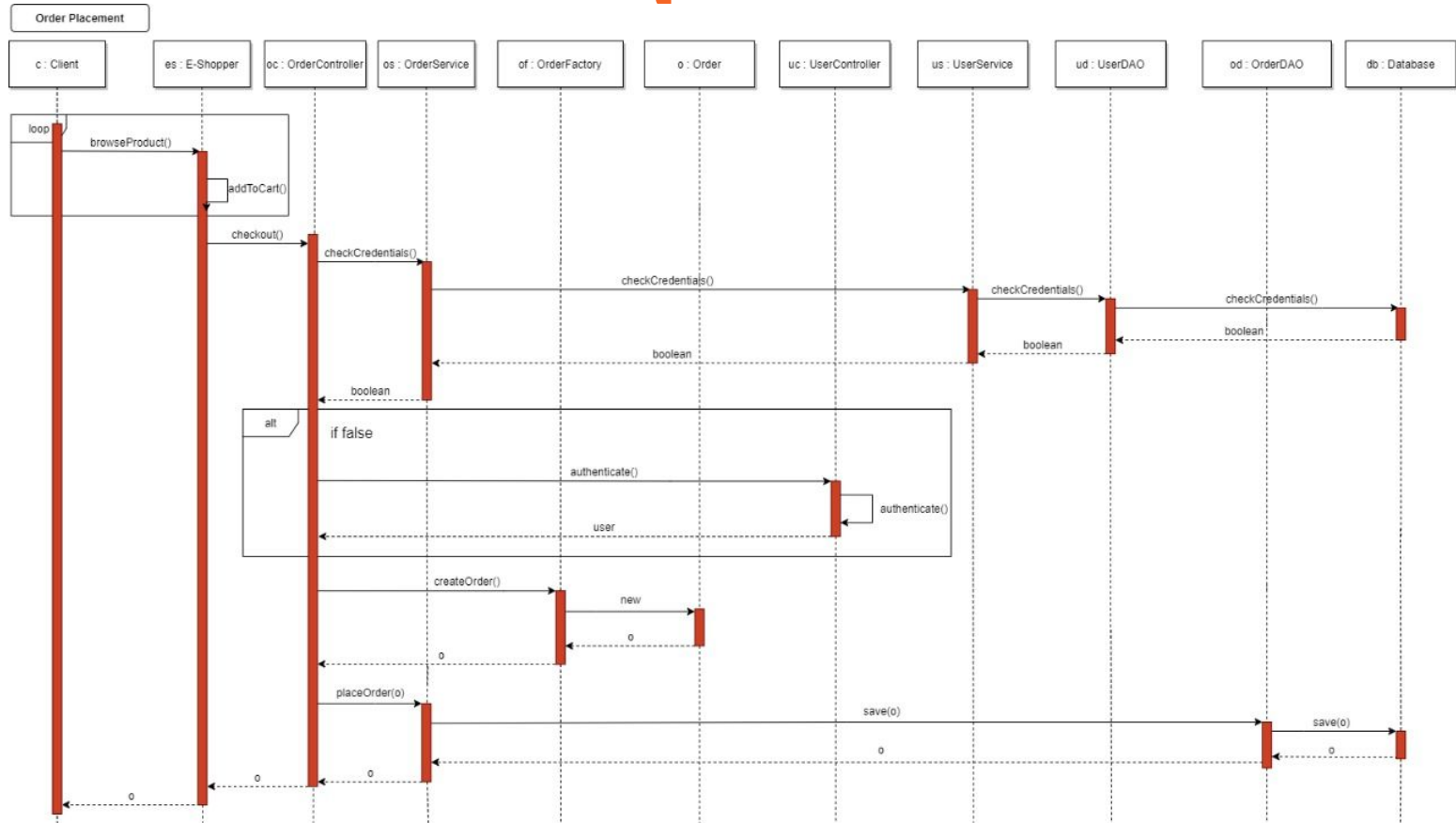
Merchant Approval Sequence
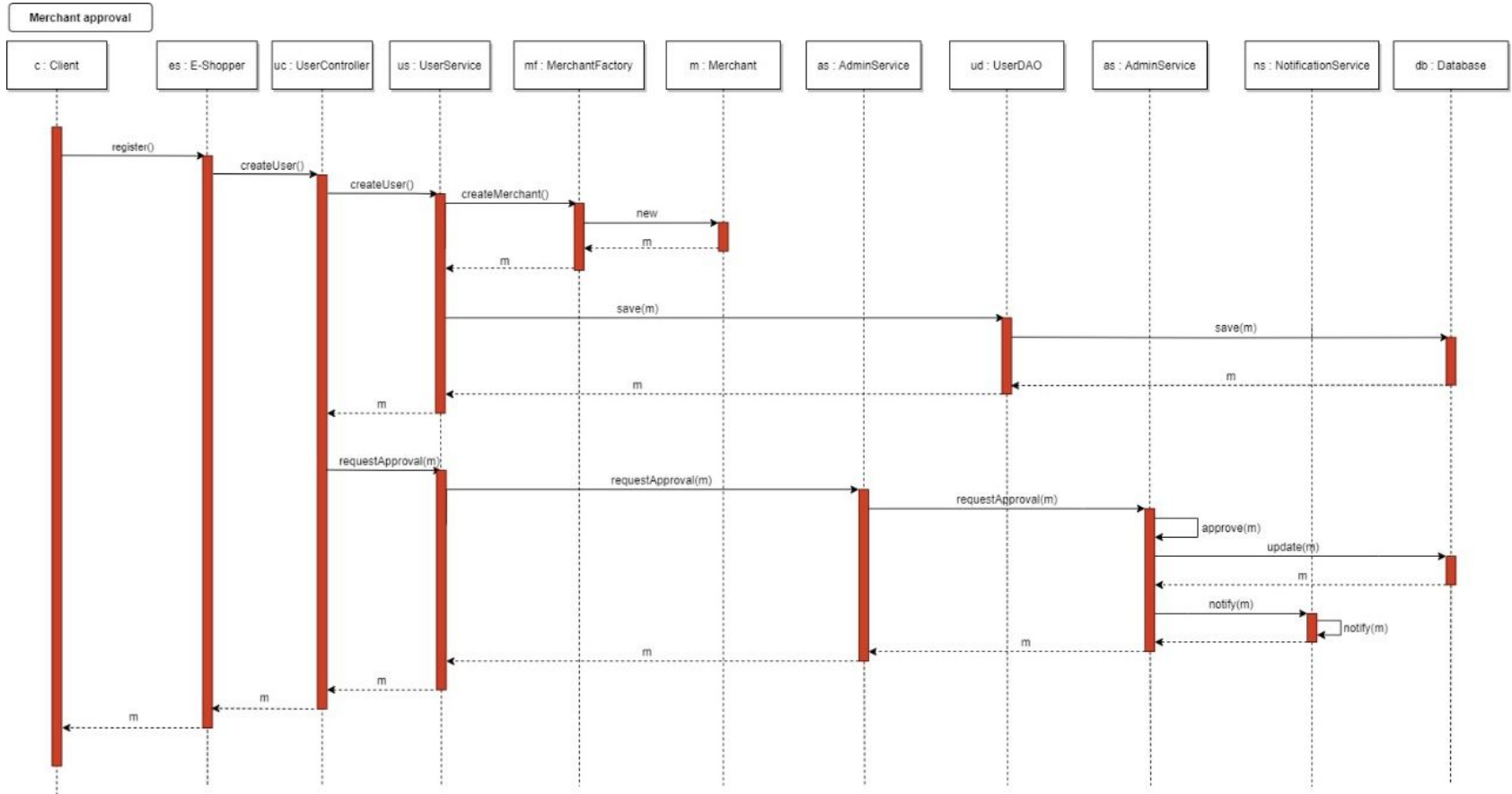
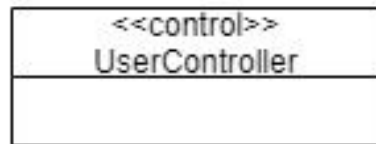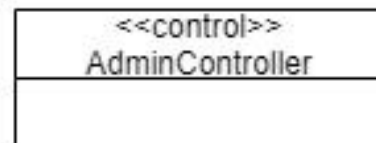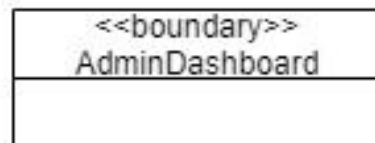# Collaboration Diagram Place Order

Collaboration Merchant Approval

VOPC Merchant Approval

VOPC Diagram
Merchant Approval

<<boundary>>
UserListForm

<<boundary>>
AdminDashboard

<<control>>
AdminController

<<control>>
UserController

<<entity>>
Merchant
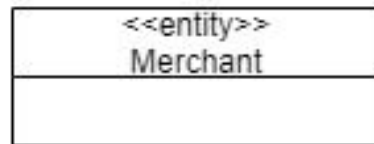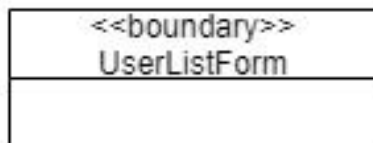
VOPC Place Order

interaction VOPC Place Order

Place Order VOPC

«boundary»
OrderPage

«control»
OrderController

«control»
OrderService

«control»
OrderFactory

«control»
OderDAO

«entity»
Order

Thank You!