****

**Detailed Design Document**

**DDD**

# Introduction

# Mid-Level Design

# Low-Level Design

# Sequence Diagrams

# 5. Design Rationale

Detailed Design Document (DDD)

# 1. Introduction

The Shoe Review system is designed to facilitate the management of user reviews, wishlists, orders, and shoe inventory. This document provides a comprehensive design, detailing the system's structure, its main components, and their interactions. The goal is to offer a robust, scalable, and user-friendly platform for shoe enthusiasts.

# 2. Mid-Level Design

This section describes the key classes in the system, their attributes, methods, and relationships.



## Class Descriptions

### User

\*\*Description\*\*: Represents a system user who can write reviews, place orders, and manage wishlists.

\*\*Attributes\*\*:

* - userId: Integer
* - name: String
* - email: String
* - password: String

\*\*Methods\*\*:

* - register()
* - login()
* - updateProfile()

### Wishlist

\*\*Description\*\*: Represents a collection of shoes that a user saves for future reference.

\*\*Attributes\*\*:

* - wishlistId: Integer
* - userId: Integer
* - name: String

\*\*Methods\*\*:

* - addItem(shoeId: Integer)
* - removeItem(shoeId: Integer)

### Order

\*\*Description\*\*: Represents a purchase containing one or more items.

\*\*Attributes\*\*:

* - orderId: Integer
* - userId: Integer
* - orderDate: Date
* - totalPrice: Double

\*\*Methods\*\*:

* - placeOrder()
* - calculateTotal()

### Shoe

\*\*Description\*\*: Represents a shoe product available for purchase or wishlist.

\*\*Attributes\*\*:

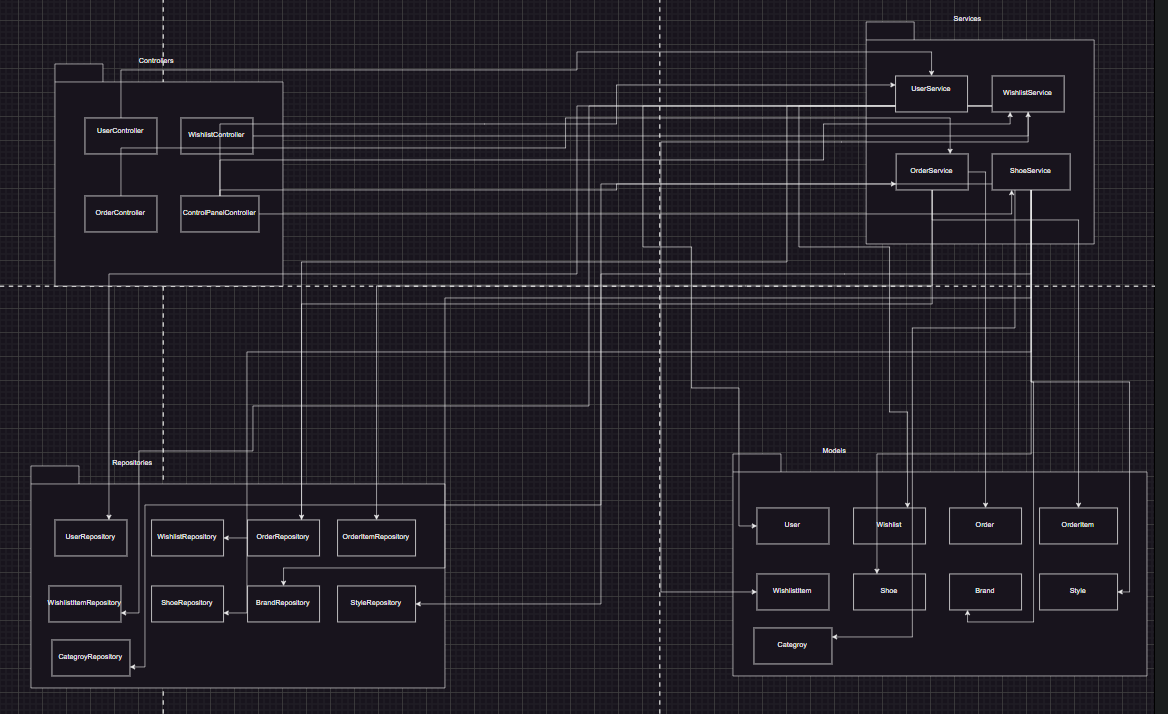
* - shoeId: Integer
* - name: String
* - brandId: Integer
* - styleId: Integer
* - categoryId: Integer
* - price: Double
* - stock: Integer
* - description: String

\*\*Methods\*\*:

* - updateStock(quantity: Integer)
* - applyDiscount(percentage: Double)

# 3. Low-Level Design

The Low-Level Design organizes the system into logical packages based on the Model-View-Controller (MVC) architecture. The Package Diagram below illustrates the organization and dependencies between these packages.



## Package Responsibilities

### Controllers Package

Manages user requests and delegates tasks to the Services layer.

Components:

* - UserController
* - WishlistController
* - OrderController
* - ControlPanelController

### Services Package

Implements business logic and processes data received from Controllers.

Components:

* - UserService
* - WishlistService
* - OrderService
* - ShoeService

### Repositories Package

Interacts with the database to perform CRUD operations.

Components:

* - UserRepository
* - WishlistRepository
* - WishlistItemRepository
* - OrderRepository
* - OrderItemRepository
* - ShoeRepository
* - BrandRepository
* - StyleRepository
* - CategoryRepository

### Models Package

Represents the core entities in the system.

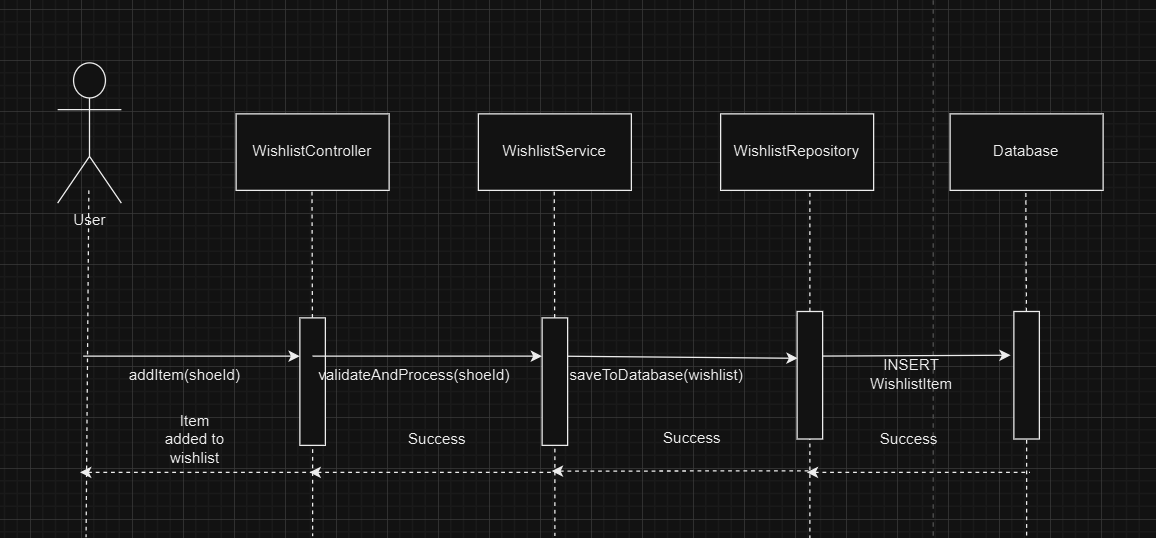
Components:

* - User
* - Wishlist
* - WishlistItem
* - Order
* - OrderItem
* - Shoe
* - Brand
* - Style
* - Category

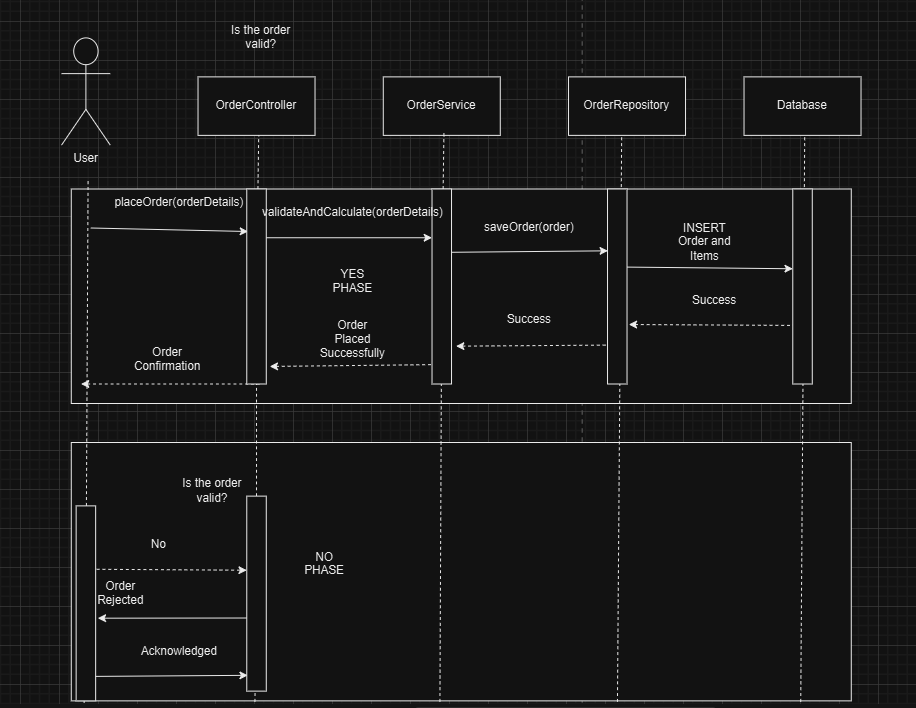
# 4. Sequence Diagrams

Sequence diagrams illustrate the interactions between system components during key workflows.

Sequence Diagram WishList



Sequence Diagram OrderService



# 5. Design Rationale

The design of the Shoe Review system was guided by the following principles and considerations:

- Separation of concerns through the MVC architecture, ensuring maintainability and scalability.

- Flexibility in managing user preferences with distinct Wishlist and WishlistItem classes.

- Aggregation and composition relationships between entities for accurate data representation.