Group 05

Sneaker Sales Website
Software Architecture Document

Version 1.0

Revision History

Date	Version	Description	Author
22-Nov-2019	1.0	draft the software architecture document	Nguyễn Phạm Quang Định

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This Software Architecture Document provides an architectural overview of the Sneaker Sales System. The Sneaker Sales System is being developed by us to support online sneakers sales..

1.3 Definitions, Acronyms and Abbreviations

None.

1.4 References

Applicable references are:

- Project proposal.
- Project Plan.
- Vision document.
- User-case document.

2. Architectural Goals and Constraints

The following section outlines key requirements and system constraints that significantly affect the architecture.

2.1 Architecture Representation

This document presents the architecture as a series of views; use case view, logical view, process view and deployment view. There is no separate implementation view described in this document. These are views on an underlying Unified Modeling Language (UML) model developed using Rational Rose.

2.2 Hardware Constraints

- The website must work on both:
    - Mid range computer with a x86_64 CPU.
    - Mid range Smartphone support at least iOS 12 and Android 7.
- Required Internet Connection.
- Device should have 8Gb of RAMs for the best UX.
- All performance and loading requirements, as stipulated in the Vision Document, must be taken into consideration as the architecture is being developed.
- Database requires a midrange server with a x86_64 CPU and 8Gb of RAM.

2.3 Software Constraints

Considering the hardware constraints from the above:

- The website must work on all Operating System that support the newest version of Google Chrome and Mozilla Firefox.
- All UX functions must be available on the web browsers.
- All performance and loading requirements, as stipulated in the Vision Document, must be taken into consideration as the architecture is being developed.

2.4 User Interaction Constraints

- The website must have a friendly UI for searching and buy product for the customer.
- All functions for the admin must be easy to use and hard to make mistake.
- All performance and loading requirements, as stipulated in the Vision Document, must be taken into consideration as the architecture is being developed.

2.5 Security Constraints

- Any and all credit card or other financial transactions must be transmitted in a secured manner.
- The system should support optional encryption of all user data and activity logs.

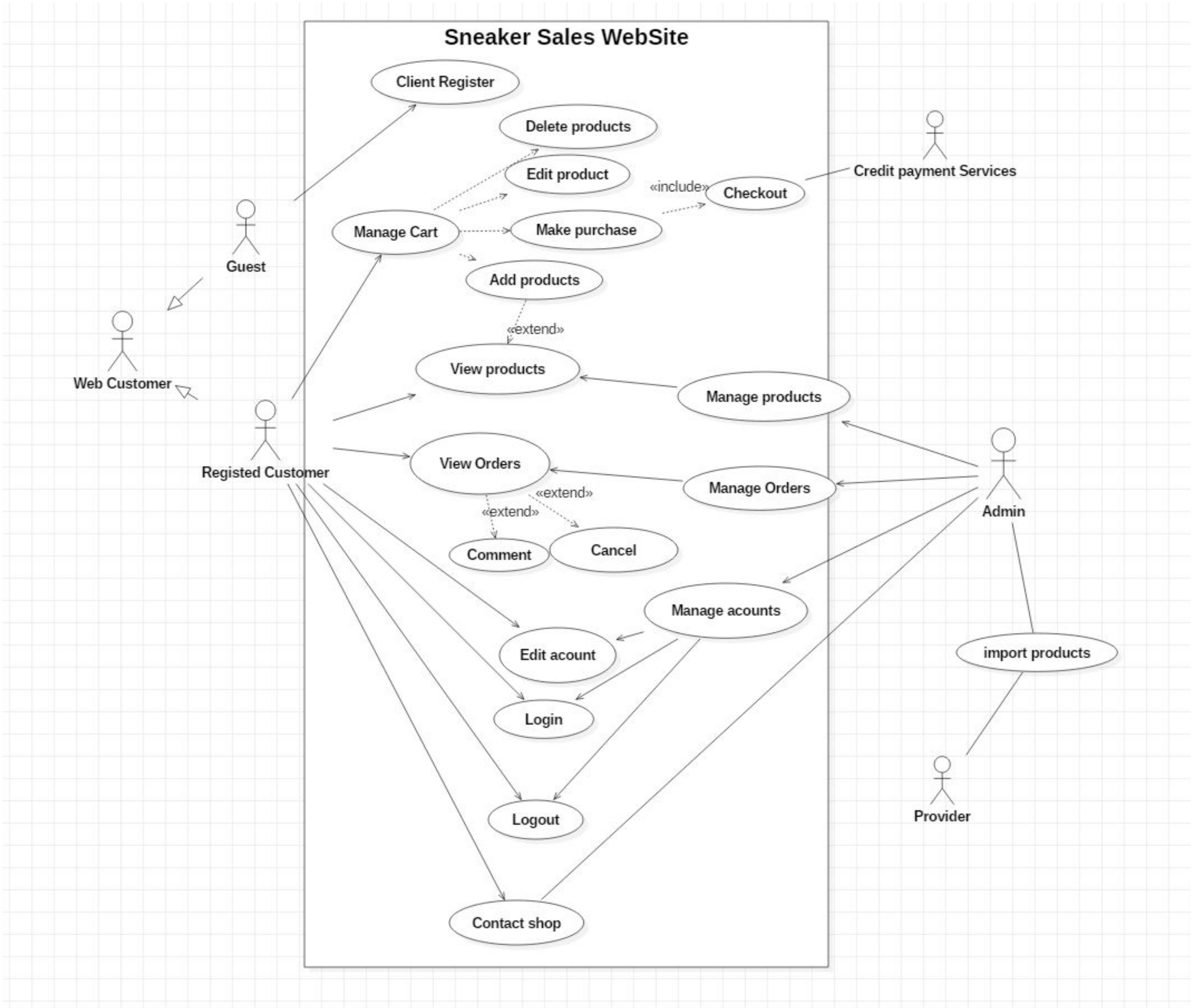
3. Use-Case Model

A description of the use-case view of the software architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

This section includes the use case diagrams that are already modeled and presented in the use-case specification document. The list of use-case are:

- Client Register.
- Login/ Registration.
- Edit account.
- Manage cart.
- Manage products.
- Manage account.
- Contact shop.
- Make purchase.
- Check-out.
- View orders.
- Manage Orders.

3.1 Significant Use Cases



3.1.1 *Client Register.*

This use case describes how a new user can register with SNEAKER sales web since they have not had the account yet. The guess is the actor of this use case.

3.1.2 *Login/ Registration.*

This use case describes how users gain access to the SNEAKER SALE WEB system through the login/registration (account creation) process.

The actors starring this use case are Guest, Registered Customer and Admin.

3.1.3 *Edit account.*

This use case describes how a User can manage his account information with SNEAKER. This included adding, modifying, and delete information from user account.

The user can remove his/her own account.

The actor of this use case is the user(customer, admin).

3.1.4 *Manage cart.*

This use case describes how an actor can manage items in the shopping cart. This includes viewing, adding, modifying and delete items in the cart. The main actor of this use case is the customers.

3.1.5 Manage products.

This use case describes how the administrator of the system can add, modify and delete items from catalog. The admin is the actor of this use case.

3.1.6 Manage account.

This use case describes how the administrator of the system manage the system users. This includes adding, modifying and delete accounts from the system. The admin is the actor of this use case.

3.1.7 Contact shop.

This use case describes how customers can get service and support using the SNEAKER shop system.

This use case occurs when a customer want to get contact with the shop for details of product or make a compliment.

The main actor of this use case is the customers. The supporting personnel is also an actor within use case.

3.1.8 Make purchase.

This use case describes how the User of the system can make payments for the orders.

The customer is the actor of this use case.

3.1.9 Check-out.

This use case describes how Credit payment Services can accept the payment from the SNEAKER webshop system. The actor of this use case is Credit payment services.

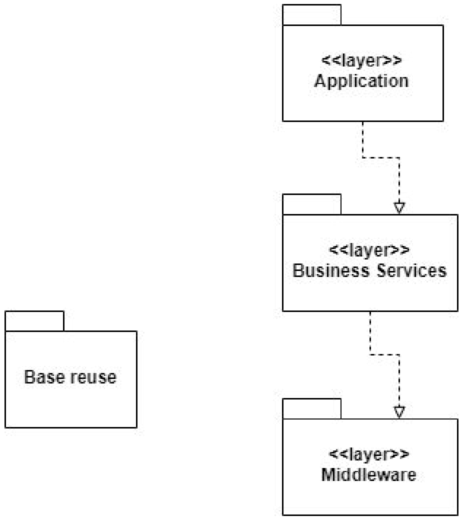
3.1.10 View orders.

This use case describes how an actor can view their order. This includes delivered orders, processing orders and cancelled orders. Users(Customers) is the actor of this use case.

3.1.11 Manage Orders

This use case describes how the administrator of the system manage the orders that are in processing. This includes adding, modifying, removing the orders. The admin is the actor of this use case.

4. Logical View



Overview

A description of the logical view of the architecture. Describes the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.

The logical view of this project is comprised of 4 main packages:

Application
This application layer has all the boundary classes that represent the application screens that the user sees. This layer depends upon the Process Objects layer; that straddles the separation of the client from mid-tier.

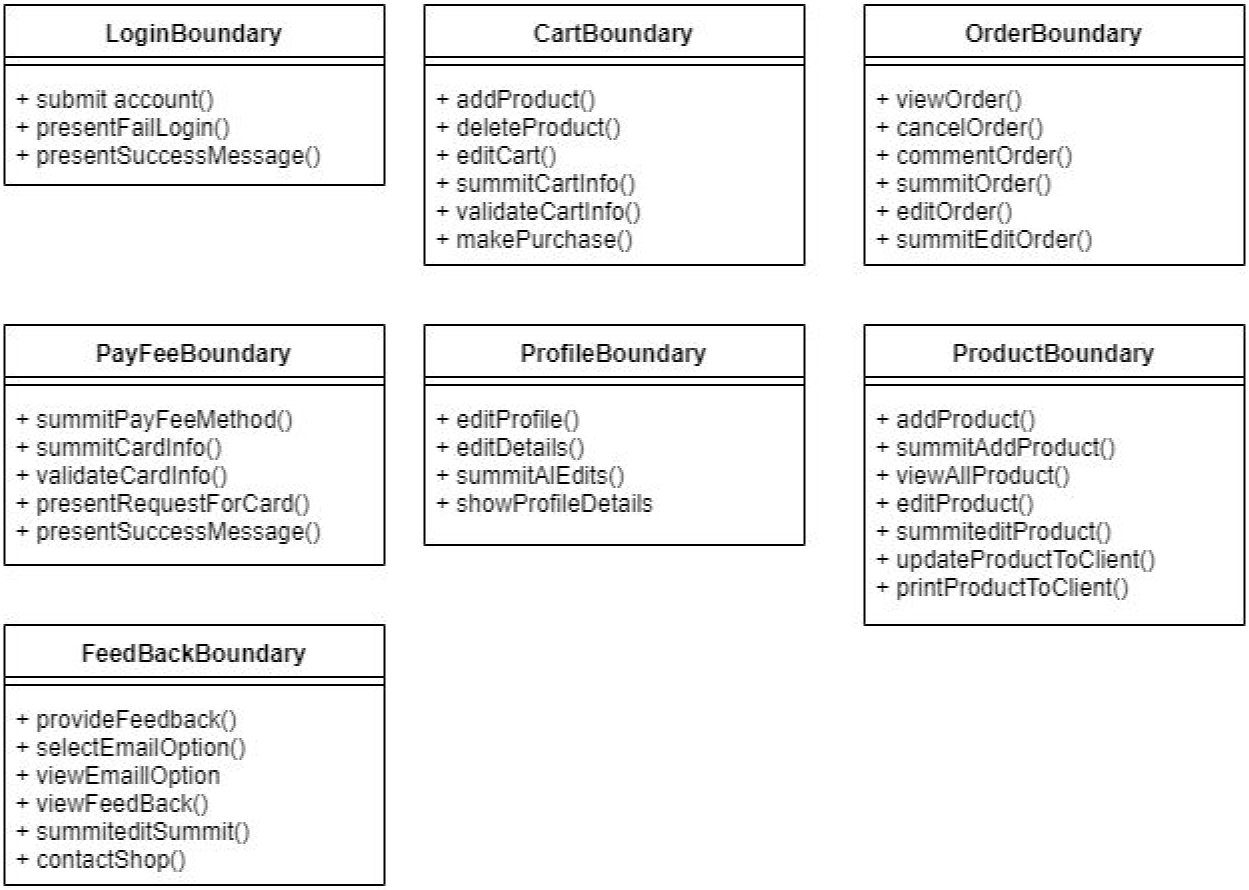
Business Services
The Business Services process layer has all the controller classes that represent the use case managers that drive the application behavior. This layer represents the client-to-mid-tier border. The Business Services layer depends upon the Process Objects layer; that straddles the separation of the client from mid-tier.

Middle layer
The Database layer supports access to Relational DBMS and OODBMS.

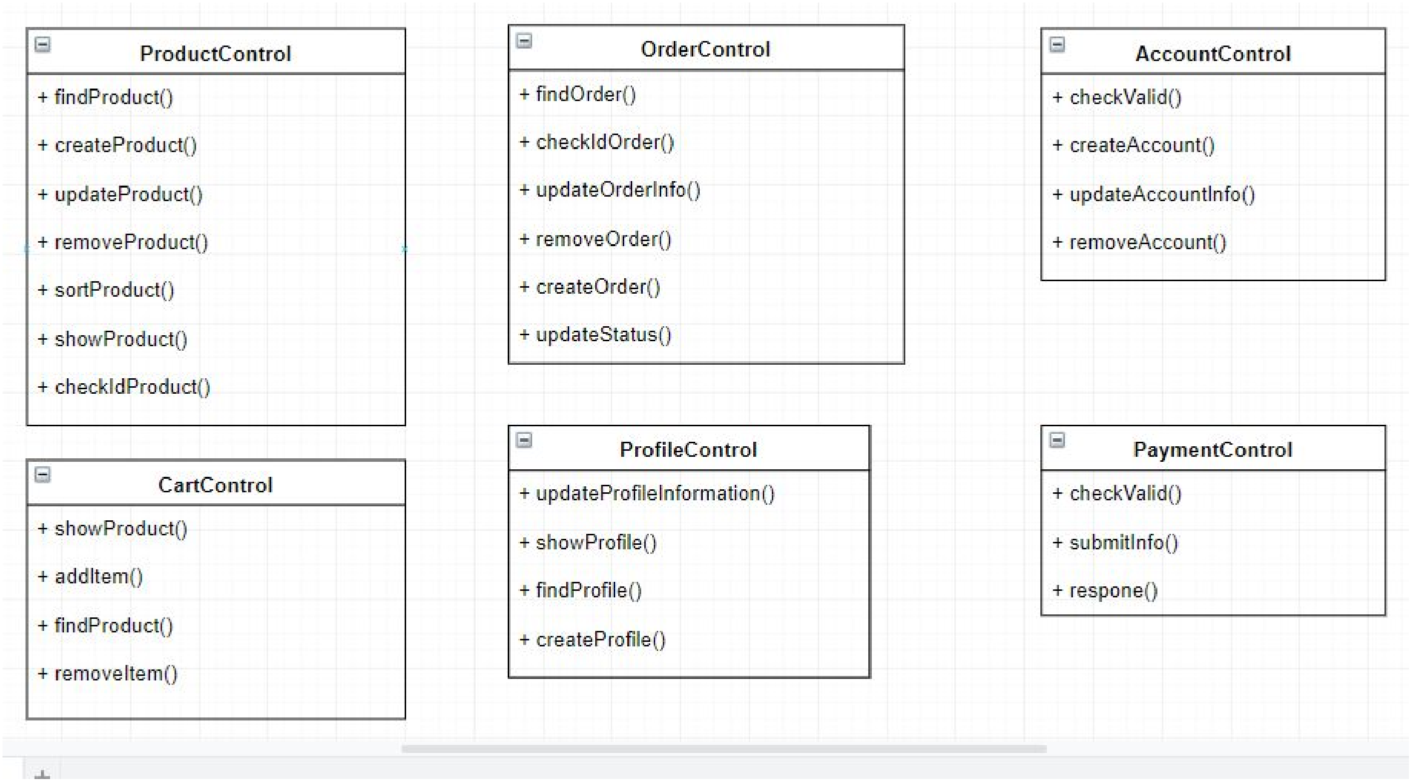
Base Reuse

The Base Reuse package includes classes to support list functions and patterns.

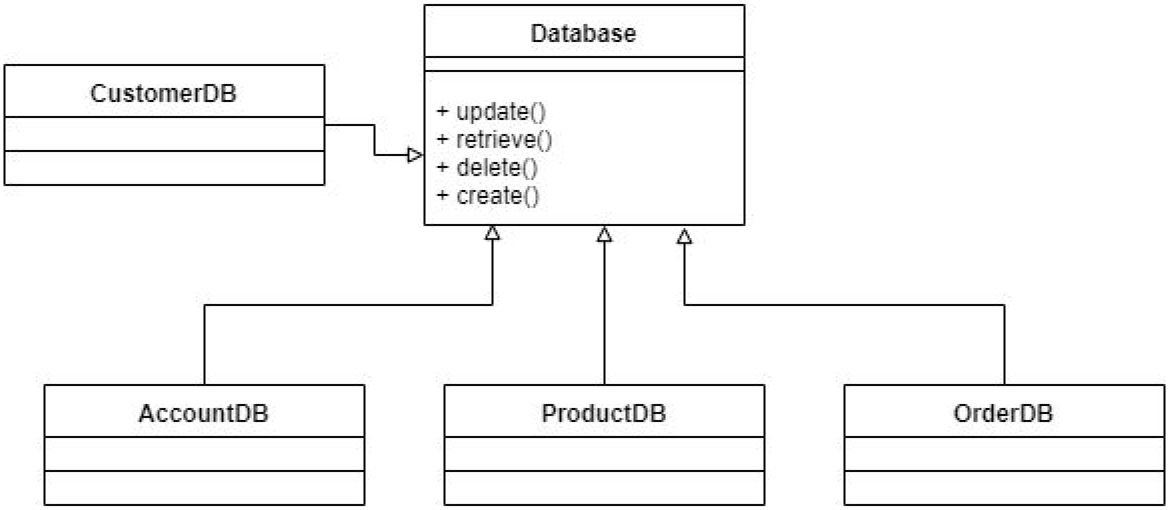
4.1 Application package



4.2 Business services



4.3 Middleware package



5. Deployment

[Leave this section blank for PA3.]

6. Implementation View

[Leave this section blank for PA3.]