

**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

**Capstone Project Document**

**House Decor System**

|  |  |
| --- | --- |
| **Nhóm số** | |
| **Group members** | Cao Đức Sơn Ngọc (Team Leader)  Lê Đại An – SE61479 |
| **Supervisor** | Thân Thị Ngọc Vân |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** |  |

**-Ho Chi Minh City, *12 August 2017*-**

***This page is intentionally left blank***

**ACKNOWLEDGEMENT**

We wish to thank various people for their contribution to this project: Our teachers for their advice and participation in the final review, our friends for their valuable technical support.

Special thanks should be given to Teacher Thân Thị Ngọc Vân, our supervisor for his professional guidance and the useful, constructive recommendations throughout the course of this project.

**Table of Contents**

[**Table of Contents** 4](#_Toc490259111)

[**List of Tables**](#_Toc490259112) 6

[List of Figures](#_Toc490259113) 7

[Definitions, Acronyms, and Abbreviations](#_Toc490259114) 8

[A. Report No. 1 Introduction](#_Toc490259115) 9

[1. Project Information](#_Toc490259116) 9

[2. Introduction](#_Toc490259117) 9

[3. Current Situation 9](#_Toc490259118)

[4. Problem Definition 9](#_Toc490259119)

[5. Proposed Solution 9](#_Toc490259120)

[5.1 Technologies 9](#_Toc490259121)

[5.2 Feature functions 10](#_Toc490259121)

[5.3 Advantages and disadvantages 1](#_Toc490259122)0

[6. Functional Requirements 10](#_Toc490259123)

[7. Role and Responsibility 1](#_Toc490259124)1

[B. Report No.2 Software Project Management Plan 1](#_Toc490259125)2

[1. Problem Definition 1](#_Toc490259126)2

[1.1 Name of this Capstone Project 1](#_Toc490259127)2

[1.2 Problem Abstract 1](#_Toc490259128)2

[1.3 Project Overview 1](#_Toc490259129)2

[1.4 Boundaries of the system 1](#_Toc490259129)3

[1.5 Future Plans 1](#_Toc490259129)4

[1.6 Development Environment 1](#_Toc490259129)4

[2. Project organization 1](#_Toc490259130)5

[2.1 Software Process Model 1](#_Toc490259131)5

[2.2 Roles and responsibilities 1](#_Toc490259132)5

[2.3 Tools and Technique 1](#_Toc490259133)6

[3. Project Management Plan 1](#_Toc490259134)7

[3.1 Software development life cycle 1](#_Toc490259135)7

[3.2 Phase Detail 2](#_Toc490259136)0

[3.3 All Meeting Minutes 2](#_Toc490259137)2

[C. Report No. 3 Software Requirement Specification 2](#_Toc490259139)3

[1. User Requirement Specification 2](#_Toc490259140)3

[1.1 Guest Requirement 2](#_Toc490259141)3

[1.2 User Requirement 2](#_Toc490259142)3

[1.3 Professional Requirement 2](#_Toc490259143)3

[1.4 Admin Requirement 2](#_Toc490259144)4

[1.5 System Requirement 2](#_Toc490259145)4

[2. System Requirement Specification 2](#_Toc490259146)5

[2.1 External Interface Requirement 2](#_Toc490259147)5

[2.2 System Overview Use Case 2](#_Toc490259148)5

[3. Software System Attribute](#_Toc490259150) 27

[3.1 Usability](#_Toc490259151) 27

[3.2 Reliability 27](#_Toc490259152)

[3.3 Availability 27](#_Toc490259153)

[3.4 Security 27](#_Toc490259154)

[3.5 Maintainability 27](#_Toc490259155)

[3.6 Portability 27](#_Toc490259156)

[3.7 Performance 27](#_Toc490259157)

[D. Report No. 4 Software Design Description 28](#_Toc490259159)

[1. Design Overview 28](#_Toc490259160)

[2. System Architectural Design 29](#_Toc490259161)

[2.1 Web application architecture description 29](#_Toc490259162)

[3. Component Diagram 30](#_Toc490259163)

[4. Detailed Description 31](#_Toc490259164)

[4.1 Class Diagram 31](#_Toc490259165)

[4.2 Class Diagram Explanation 32](#_Toc490259166)

[4.3 Interaction Diagram 39](#_Toc490259167)

[5. Database Design 45](#_Toc490259171)

[5.1 Entity relationship diagram (ERD) 45](#_Toc490259172)

[5.2 Data Dictionary 45](#_Toc490259173)

[6. Algorithms 46](#_Toc490259174)

[E. System Implementation & Test 48](#_Toc490259176)

[1. Introduction 48](#_Toc490259177)

[1.1 Overview 48](#_Toc490259178)

[2. Database Relationship Diagram 49](#_Toc490259180)

[2.1 Physical Diagram 49](#_Toc490259181)

[2.2 Data Dictionary 49](#_Toc490259182)

**List of Tables**

[**Table 1: Definitions, Acronyms, and Abbreviations** 9](#_Toc478543610)

[**Table 2: Roles and Responsibilities** 12](#_Toc478543611)

[**Table 3: Hardware Requirement for server** 15](#_Toc478543612)

[**Table 4: Software Requirement** 16](#_Toc478543613)

[**Table 5: Roles and Responsibilities** 17](#_Toc478543614)

[**Table 6: Tool List** 17](#_Toc478543615)

[**Table 7: Technique List** 18](#_Toc478543616)

[**Table 8: Software Development Life Cycle Detail** 20](#_Toc478543617)

[**Table 9: Phase 1: Infrastructure** 21](#_Toc478543618)

[**Table 10: Phase 2: System & Web admin** 22](#_Toc478543619)

[**Table 11: Phase 3: Web service** 23](#_Toc478543620)

[**Table 12: Component Dictionary** 31](#_Toc478543621)

[**Table 13: Class Diagram Dictionary**](#_Toc478543622) 33

[**Table 14: Admin class methods explanation** 33](#_Toc478543623)

[**Table 15: Professional class attributes explanation** 34](#_Toc478543624)

[**Table 16: Professional method methods explanation** 34](#_Toc478543625)

[**Table 17: User class attributes explanation** 34](#_Toc478543626)

[**Table 18: User class methods explanation** 34](#_Toc478543627)

[**Table 19: Role class attributes explanation** 34](#_Toc478543628)

[**Table 20: Role class methods explanation** 34](#_Toc478543629)

[**Table 21: Order class attributes explanation** 35](#_Toc478543630)

[**Table 22: Order class methods explanation** 35](#_Toc478543631)

[**Table 23: Order Detail class attributes explanation** 35](#_Toc478543632)

[**Table 24:** **Order Detail class methods explanation** 35](#_Toc478543633)

[**Table 25:** **City class attributes explanation** 35](#_Toc478543634)

[**Table 26: City class methods explanation** 35](#_Toc478543635)

[**Table 27: Tracking class attributes explanation** 36](#_Toc478543635)

[**Table 28: Tracking class methods explanation** 36](#_Toc478543635)

[**Table 29: Category class attributes explanation** 36](#_Toc478543635)

[**Table 30: Category class methods explanation** 36](#_Toc478543635)

[**Table 31: Photo class attributes explanation** 36](#_Toc478543635)

[**Table 32: Photo class methods explanation** 36](#_Toc478543635)

[**Table 33: Project class attributes explanation** 37](#_Toc478543635)

[**Table 34: Project class methods explanation** 37](#_Toc478543635)

[**Table 35: Idea book class attributes explanation** 37](#_Toc478543635)

[**Table 36: Idea book class methods explanation** 37](#_Toc478543635)

[**Table 37: Product photo class attributes explanation** 37](#_Toc478543635)

[**Table 38: Product photo class methods explanation** 38](#_Toc478543635)

[**Table 39: Type of work class attributes explanation** 38](#_Toc478543635)

[**Table 40: Type of work class methods explanation** 38](#_Toc478543635)

[**Table 41: Seller class attributes explanation** 38](#_Toc478543635)

[**Table 42: Seller class methods explanation** 38](#_Toc478543635)

[**Table 43:** **Promotion class attributes explanation** 38](#_Toc478543635)

[**Table 44: Promotion class methods explanation** 39](#_Toc478543635)

[**Table 45:** **Product class attributes explanation** 39](#_Toc478543635)

[**Table 46: Product class methods explanation** 39](#_Toc478543635)

[**Table 47: Style class attributes explanation** 39](#_Toc478543635)

[**Table 48: Style class methods explanation** 39](#_Toc478543635)

[**Table 49: Promotion Detail class attributes explanation** 40](#_Toc478543635)

[**Table 50: Promotion Detail class methods explanation** 40](#_Toc478543635)

[**Table 51: Idea book photo ref class attributes explanation** 40](#_Toc478543635)

[**Table 52: Idea book photo ref class methods explanation** 40](#_Toc478543635)

[**Table 53: Data dictionary: Describe content of all tables** 55](#_Toc478543635)

**List of Figures**

[Figure 1: Scrum model 17](#_Toc478471179)

[Figure 2: System Overview Use Case 27](#_Toc478471180)

[Figure 3: System Architectural Design 30](#_Toc478471189)

[Figure 4: Web Service Architecture 31](#_Toc478471189)

[Figure 5: Web Service Architecture 32](#_Toc478471189)

[Figure 6: Entity Relationship Diagram (ERD) 46](#_Toc478471189)

[Figure 7: <Database Relation Diagram> Physical Diagram 49](#_Toc478471189)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| **GUI** | Graphic User Interface |

**Table 1: Definitions, Acronyms, and Abbreviation**

**Report 1 – Introduction**

1. **Project Information**

* Project name: **Online furniture retailer**
* Abbreviation: **HouseDecor**
* Product Type: **Web Application**
* Start Date: **May 10, 2017**
* End Date:

1. **Introduction**

House is very important to everyone. In order to have a house as your wish, you will search for home services. So far, people select home service providers often through referrals of friends or relatives. This leaves the user with limited choice and more difficult in finding reliable service providers. On the service provider side, they are also limited in introducing their package services or products.

As a result, we built a system that helps people find the home service provider easier and the service provider will have more customers.

1. **Current Situation**

* The supplier of interior and exterior products has many, but the introduction of online products is limited because it costs to build and maintain a website.
* The problems of housing and the way of thinking in Vietnam are always carefully selected and specific authentication (for example: ask people who have just built houses, neighbors or find relatives). Private websites are rarely found unless they are big, reputable companies.
* Customer services usually work in groups including contractors, builders, painters, electricians water. The customer only needs minor repairs, it is difficult to find the painters, electricians, service clean the house…

1. **Problem Definition**

* In Vietnam, there aren’t any home building services website.
* Cost to build and maintain a website is an issue that many suppliers can’t introduce their products and services online.

1. **Proposed Solution**

Our proposed solution is to build a website Online furniture retailer:

- Provide information and sell seller's products.

- Support to search for service providers: interior design, exterior, construction contractor, home repair services.

- It also supports users to upload designs, share ideas, stories related to the exterior and interior.

* 1. **Technologies**

3 tiers:

* Server side:
* Database Design, OOP, MVC, Java technology.
* Client side:
* CSS, JavaScript, jQuery.
* Communication:
* Exchange information and transfer data in effective in networks.
  1. **Feature Functions**

Web application:

* For customer:
* Browse and save beautiful home photos with others.
* Find the right design and construction professionals.
* For supplier:
* Show their services, projects, products and communicate with customers in their posts to find suitable orders.
* Advertise their products, services to customers.
  1. **Advantages and disadvantages**
* **Advantages:**
* Bring new experience about design and construction home.
* Help customer to save time and money.
* Help customer to find their own supplier.
* Help supplier to have more customers, orders.
* **Disadvantages:**
* B2B website is uncommon for customer.
* Supplier provides services or sell products at price equal or lower than market price.

1. **Functional Requirements**

**Admin component:**

* Admin can manage all accounts: activate/deactivate.
* Admin can manage all products, idea books, projects, photos: view/accept/block.
* Admin can search accounts, products, idea books, projects.

**Guest component:**

* Guest can view and share photos, idea books, projects, products and profile of others.
* Guest can manage their cart: add/edit/remove but can’t check out.
* Guest can search/filter products, projects, professionals.
  + - Guest can register account to be Member or Professional

**Member component:**

* Member can edit profile.
* Member can view and share photos, idea books, projects, products and profile of others.
* Member can manage their idea books, photos: add/edit/delete.
* Member can manage their cart: add/edit/delete and check out.

Member can search/filter products, projects, professionals

**Professional component:**

* Professional can edit profile.
* Member can view and share photos, idea books, projects, products and profile of others.
* Professional can manage their idea books, projects, photos: add/edit/delete.
* Professional can manage their cart: add/edit/delete and check out.
* Professional can search/filter products, projects, professionals.
* Professional can register to be Seller

**Seller component:**

* Seller can manage their products: add/edit/delete.
* Seller can update order’s status.
* Seller can view sale-report by day, month.

**System component:**

* System send SMS, email to Seller when Seller have an order.
* System tracking member, professional’s activities.
* System minus quantity of product when the product is ordered.
* System send Advertisement to member, professional.
* System hide/show account’s posts when the account is banned/unbanned.
* System send notification to Admin, Member, Professional, Seller (notify their activities).

1. **Role and Responsibility**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Thân Thị Ngọc Vân | Project Librarian | Supervisor | [Vanttn2@fe.edu.vn](mailto:Vanttn2@fe.edu.vn) |
| 2 | Cao Đức Sơn Ngọc | Developer | Leader | [Ngoccdsse61116@fpt.edu.vn](mailto:Havhse61394@fpt.edu.vn) |
| 3 | Lê Đại An | Developer | Member | [Ledaian41@gmail.com](mailto:Ledaian41@gmail.com) |
| 4 | Võ Trần Hoàng Long | Developer | Member | [Longvth13@gmail.com](mailto:Longvth13@gmail.com) |
| 5 | Trịnh Duy Nhân | Developer | Member | Duynhan102209@gmail.com |

*Table 2: Roles and Responsibilities*

# Report 2 – Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name:** Online Furniture Retailer
* **Vietnamese name:** Website bán hàng trực tuyến đồ nội ngoại thất
* **Abbreviation:** HouseDecor

### Problem Abstract

We build the system to support customer to find home building services and home products with high quality and reasonable price. Our system also support supplier to sell more products and have more customer use services. For that reason, we create a website that achieve their requirements.

In order to satisfy customer’s demand, we help them create idea books, buy products. Customer can also make a review about the suppliers.

For supplier, the system helps suppliers manage their online shop, their projects had built and notify suppliers about orders.

### Project Overview

#### 1.3.1 Current Situation

Below are the problems encountered in this project:

* Customer’s habit: Customers are used to buy home products at shop, showroom or ask family, friends, neighbors about home services when looking forward to building a house. Or search on the search engines, customers will find many results and the results are not grouped.
* Suppliers: Spend much money to marketing. Old-customer's introductions is the only way that suppliers have more customers.

#### 1.3.2 The Proposed System

Website provides following features:

* + - **For Admin:**
* Admin can manage all accounts: activate/deactivate.
* Admin can manage all products, idea books, projects, photos: view/accept/block.
* Admin can search accounts, products, idea books, projects.
  + - **For Guest:**
* Guest can view and share photos, idea books, projects, products and profile of others.
* Guest can manage their cart: add/edit/remove but can’t check out.
* Guest can search/filter products, projects, professionals.
* Guest can register account to be Member or Professional.
  + - **For Member:**
* Member can edit profile.
* Member can view and share photos, idea books, projects, products and profile of others.
* Member can manage their idea books, photos: add/edit/delete.
* Member can manage their cart: add/edit/delete and check out.
* Member can search/filter products, projects, professionals.
  + - **For Professional:**
* Professional can edit profile.
* Member can view and share photos, idea books, projects, products and profile of others.
* Professional can manage their idea books, projects, photos: add/edit/delete.
* Professional can manage their cart: add/edit/delete and check out.
* Professional can search/filter products, projects, professionals.
* Professional can register to be Seller.
  + - **For Seller:**
* Seller can manage their products: add/edit/delete.
* Seller can update order’s status.
* Seller can view sale-report by day, month.
  + - **For System:**
* System send email to Seller when Seller have an order.
* System tracking member, professional’s activities.
* System minus quantity of product when the product is ordered.
* System send promotions to member, professional.
* System hide/show account’s posts when the account is banned/unbanned.
* System send notification to Admin, Member, Professional, Seller (notify their activities).

### Boundaries of the system

* The web application provides a place to browse and save beautiful home photos, projects. The website helps to connect people each other about home services. The website provides following services:
  + Browse and save to idea book beautiful home photos.
  + Find and view detail of photos, products, projects, full stories about designers, home builders, home services, …
  + Find, view, buy other kinds of products (shop by categories: living room, bed room…; Shop by style: Victoria, Chic, Modern…; Shop by indoor, outdoor…
  + Manager user accounts, this function includes:
* Edit profile.
* Idea books: A place to store all photo saved.
* Upload photos: You can share your photos.
* Activity: you can see your comments, shared photos and some discussions about it.
  + Tracking user’s activity to send relative products, relative topic from the website to user’s email. The webmaster hopes that some of the audience will visit the website.
  + Send new promotions to user’s email.
  + other functions: View order info, view buying guides, FAQs, …

### Future Plans

With further research and development, the system can apply the following features:

* Provide online payment feature.
* Upgrade admin component: admin component divides into different roles/levels to manage website.
* Provide a master account and master component for owner business to manages admin component.

### Development Environment

#### Hardware requirements

* **For web application server**

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Window Server 2008 R2 | Window Server 2012 R2 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core |
| **Computer Memory** | 2GB of RAM | 4GB of RAM or more |

Table 3: Hardware Requirement for Server

#### Software requirements

|  |  |  |
| --- | --- | --- |
| **Software** | **Name / Version** | **Description** |
| **Operating system** | Window 10 Professional | Operating system and platform for development |
| **Environment** | JDK version 1.8.0 update 101 | Specification for developing web application |
| **IDE** | NetBeans IDE 8.2 | Use to implement website. |
| **Design Model tool** | StartUML v2.5.1 | Use to create modal and diagrams. |
| **DBMS** | MySQL | Use to create & manage the database for system |
| **Document storage** | GitHub | Use to store document |
| **Store and manage source code** | GitHub | Use to store all source code |
| **Web browser** | Chrome 42 or above | Testing browser |

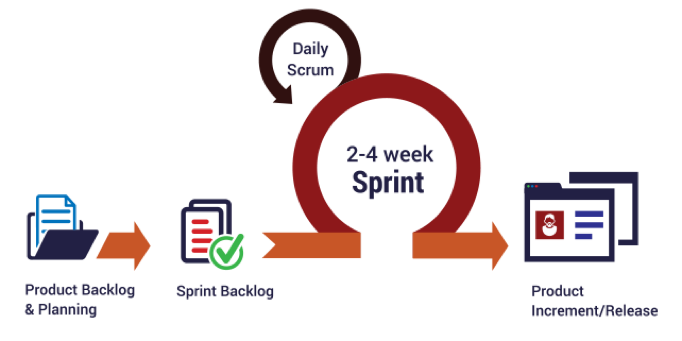
Table 4: Software requirements

## Project organization

### Software Process Model

This project is developed using Scrum Model. This model is appropriate under consideration of 2 following reasons:

* Prototypes are delivered frequently for evaluation.
* Team members can involve more in the development process.

  
Figure 1: Scrum model

Reference: <http://skytechnovation.com/scrum-development-model/>

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Thân Thị Ngọc Vân | Product owner, Project Manager | * Specify user requirements * Control the development process * Provide technical and business analysis support |
| **2** | Cao Đức Sơn Ngọc | Scrum leader, B.A, Developer, Tester | * Manage process * Design database * Clarify requirements * Prepare and combine documents * GUI design * Create test plan * Code * Test |
| **3** | Lê Đại An | Team member,  B.A, Developer,  Tester | * Design database * Clarify requirements * Prepare documents * GUI design * Create test plan * Code * Test |
| **4** | Võ Trần Hoàng Long | Team member,  B.A, Developer,  Tester | * Design database * Clarify requirements * Prepare documents * GUI design * Create test plan * Code * Test |

*Table 5: Roles and responsibilities*

### Tools and Techniques

|  |  |
| --- | --- |
| **Tool** | **Name / Version** |
| **Web server** | Tomcat |
| **Development tool** | NetBeans IDE 8.2 |
| **DBMS** | MySQL |
| **Source control** | GitHub |
| **Modeling tool** | StarUML v5.0.1 |
| **Document tool** | Microsoft Word 2016 |

*Table 6: Tools List*

|  |  |
| --- | --- |
| **Technique** | **Name / version** |
| **Frontend** | HTML, CSS, JavaScript, jQuery, Bootstrap |
| **Backend** | Java EE, Servlet, JSP, JPA, Hibernate |

*Table 7: Techniques List*

## Project Management Plan

### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constrains** | **Risks** |
| **Infrastructure** | - Identify and clarify overall requirements.  - Determine the system architecture.  - Build infrastructure for the project.  - Prepare reports. | - System main structure. | 4 days |  | - Unclear project scope.  - Lack of member share of understand. |
| **Modeling UML** | - Define actors and role of each actor in system.  - Define types and features of each actor.  - Create system’s UML models.  - Prepare reports | - UML models. | 10 days | - Depends on “Infrastructure” | - Unclear project scope.  - Lack of experience. |
| **Database** | - Create ERD describe architecture, database.  - Generate Entity.  - Prepare reports. | - ERD, entity package. | 7 days | - Depends on “Modeling UML” | - Unclear project scope.  - Lack of experience. |
| **Web services** | - Add common library, JPA, JDBC.  - Prepare reports. | - Prototype, common project. | 1 days | - Depends on “Database” |  |
| **Implements** | - Members implement system’s features.  - Test modules.  - Combine modules into system. - Prepare reports. | - Completed system. | 56 days | - Depends on “Web services” | - Unclear system’s features.  - Lack of experience. |
| **Test System** | - Create use-cases and test system based on this.  - Prepare reports. | - Test reports and system with features are fixed bugs. | 5 days | - Depends on “Implements” | - Lack of experience. |
| **Package** | - Build setup-files.  - Deploy server.  - Burn CD.  - Complete reports. | - Reports, CD, setup-files. | 1 days | - Depends on “Test System” |  |

*Table 8: Software development life cycle*

### Phase Detail

* + 1. ***Phase 1: Infrastructure***

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Assessment** | - Determine requirements.  - Create product backlog. | * NgocCDS * AnLD * LongVTH |
| **2. Selection** | - Determine system architecture: Spring Boot.  - Determine software design pattern: Repository & Service.  - Determine all core functions. | * NgocCDS * AnLD * LongVTH |
| **3. Development** | - Create the main structure of project. | * NgocCDS * AnLD * LongVTH |
| **4. Review** | - Review all completed works and presentation.  - Create sprint backlog. | * NgocCDS * AnLD * LongVTH |

*Table 9: Infrastructure*

* + 1. ***Phase 2: System & Web admin***

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Assessment** | - Determine requirements for System and Web app.  - Update product backlog. | * NgocCDS * AnLD * LongVTH |
| **2. Selection** | - Determine all functions according to requirements of System and Web app. | * NgocCDS * AnLD * LongVTH |
| **3. Development** | - Design and build prototype for web UI  - Create conceptual diagram  - Design class diagram  - Design database  - Implement the entire web UI: layouts, detail pages, etc.  - Implement all the functions in controllers.  - Build needed utility classes | * NgocCDS * AnLD * LongVTH |
| **4. Review** | - Review all completed works and presentation.  - Create sprint backlog. | * NgocCDS * AnLD * LongVTH |

*Table 10: System and Web Admin*

* + 1. ***Phase 3: Web service***

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Assessment** | - Determine requirements for Web service.  - Update product backlog. | * NgocCDS * AnLD * LongVTH |
| **2. Selection** | - Determine all functions according to requirements of Web service. | * NgocCDS * AnLD * LongVTH |
| **3. Development** | - Implement all the functions on the web app. | * NgocCDS * AnLD * LongVTH |
| **4. Review** | - Review all completed works and presentation.  - Create sprint backlog. | * NgocCDS * AnLD * LongVTH |

*Table 11: Web service*

### All Meeting Minutes

Meeting minutes are contained in folder “Meeting minutes” in the attached CD.

# Report 3 - Software Requirement Specification

## User Requirement Specification

### Guest Requirement

Guest are those do not login to this system. They have some functions:

* View photo
* View product
* Search product
* Search professional
* View other Profile
* View Project
* View Idea Book
* Sign up
* Sign in
* Manage shopping cart

### Member Requirement

They can use some following functions:

* Sign out
* Edit Profile
* View photo
* View product
* Search product
* Search professional
* View other profile
* View project
* View idea book
* Send an order
* Create idea book
* Edit idea book
* Delete idea book
* Add photo to idea book
* Delete photo in idea book
* Order

### Professional Requirement

Professional, who is professional in house-building service. They provide house services or sell house products (like desk, chain, table, bed…). They can use some following functions:

* Manage projects:
  + - * Create new project.
      * View project detail.
      * Edit project.
      * Delete project.
      * Add photo to project.
      * Delete photo from project.
* Seller register.
* Manage store:

### Admin Requirement

Admin, who is responsible for managing accounts, products, projects, idea books, photos, sellers for the whole system, has the following function:

* Manage accounts:
  + - * View account list.
      * Search accounts.
      * View account’s detail.
      * Deactivate account.
      * Activate account.
* Manage products:
* View products.
* Search products.
* View product’s detail.
* Accept product.
* Block product.
* Manage idea books:
* View idea books.
* Search idea books.
* View idea book’s detail.
* Approve idea books.
* Accept photo in idea book.
* Block photo in idea book.
* Manage projects:
* View projects.
* Search projects.
* View project’s detail.
* Approve projects.
* Accept photo in project.
* Block photo in project.
* Manage sellers:
* View seller’s detail
* Update sale time

### System Requirement

System, which automatically runs some following functions:

* Send order to seller by email
* Add tracking
* Remove tracking
* Update quantity of product
* Send promotion to Customers
* Send notification to User:

• Admin: have new product, image need accept

* User and Professional: their product or image has been proceed.

## System Requirement Specification



### External Interface Requirement

#### User Interface

* The user interface uses Vietnamese language for web application.
* General requirement for graphics user interface is the GUI should be simple, clear, intuitive, and reminiscent.

#### Hardware Interface

#### Software Interface

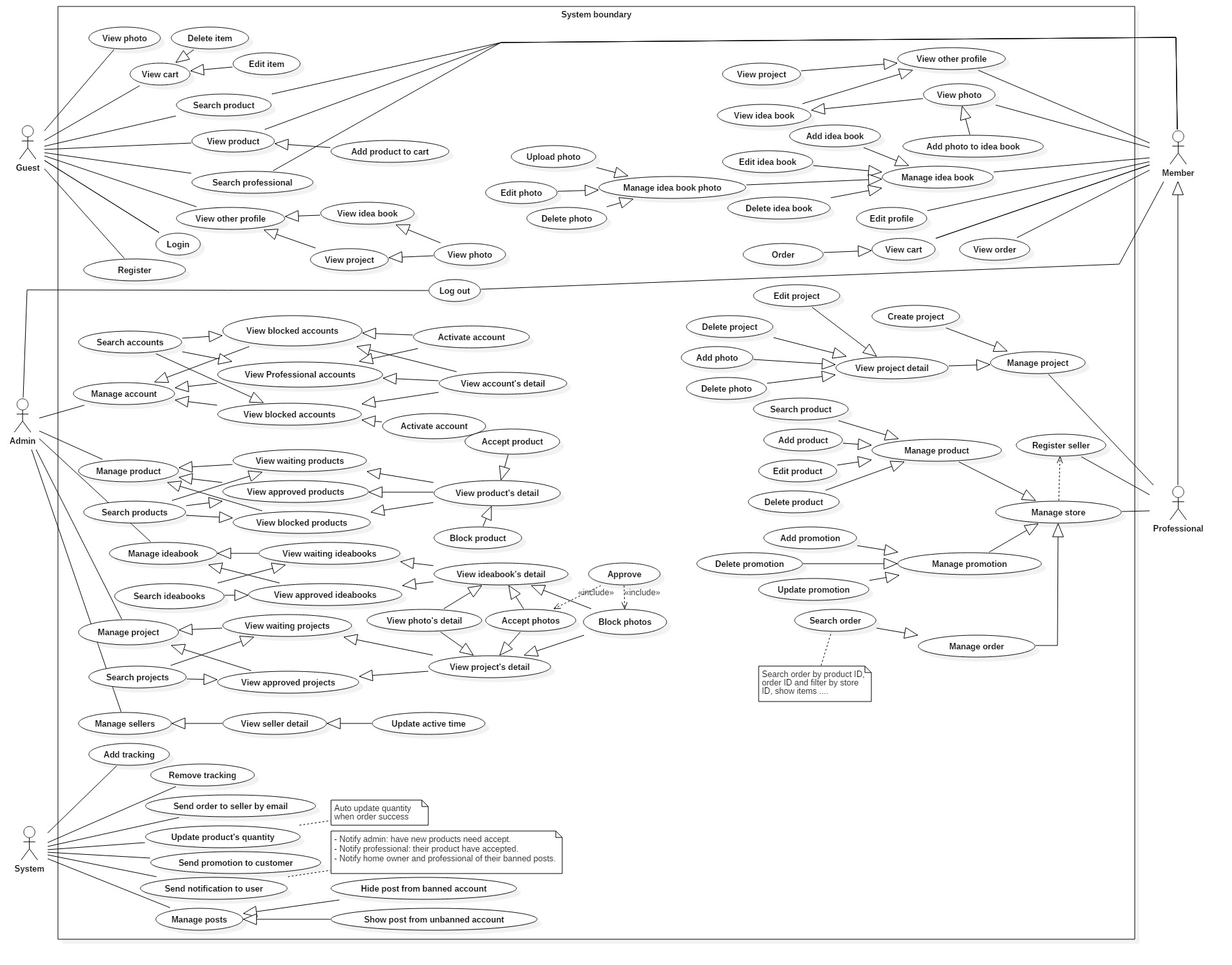
* Web application work with Firefox (v30 or above), Chrome (v14 or above), Internet Explorer (v10 or above).

#### Communication Protocol

* Use HTTP to communicate between the web browser and the web server.

### System Overview Use Case

(Figure 2 : <Use case> System overview)





## Software System Attribute



### Usability

* The Web application for guest/ member/ professional/ seller will take less than 5 minutes to get used.
* The Web application for admin will take 10-30 minutes to get used to the system completely

### Reliability

* The number of sending notification failure is 1 time per 1000 notifications.
* Timer tasks run at configured time with 100% execution rate.

### Availability

* System is available 24 hours per day and 7 days per week.
* System should take at most 5 hours per month for backup or repairing.

### Security

* Privacy: Each role of user has a specific permission to interact with system.
* System always checks authorization and authenticated before doing anything.
* Only admin can grant permission to other roles.

### Maintenance

* The system is divided into separated modules.
* The code is easy to maintain and upgrade.

### Portability

* The web application is running on Windows Server 2008 or above.

### Performance

* Web appication handles the task within 30 seconds.



# Report 4 - Software Design Description

## Design Overview

* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
* Section 1: gives an overall description of the system architecture design.
* Section 2: gives component diagrams that describe the connection and integration of the system.
* Section 3: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 4: gives the interface design description, which includes component interface, web application interface and mobile application design.
* Section 5: describe a fully attributed Entity Relationship Diagram.
* Section 6: describe the algorithms that apply to the system.

## System Architectural Design

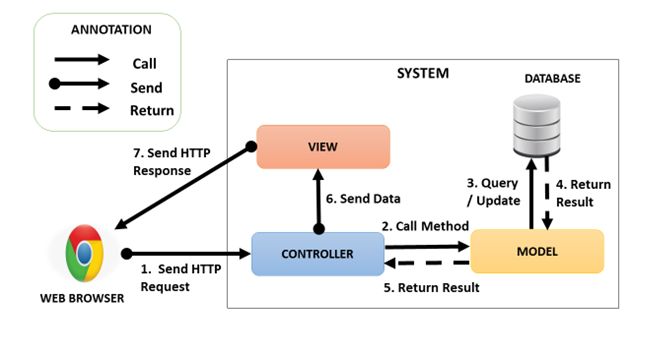


Figure 3: System Architectural Design

### Web Application Architecture Description

In Web Application, we develop the system under MVC architecture style. We choose this architecture for Web application because of following advantages:

* Web application contains Web services with MVC architecture; we can separate business code with Controller and View. In addition, we can use the business code in web service without repeat the code.
* We can organize the code better for maintainability, extensibility, reusability.
* In scope of 4-members team, MVC architecture makes it easier to split the big project into small modules and make it easier to assign each module for members in our team

## Component Diagram

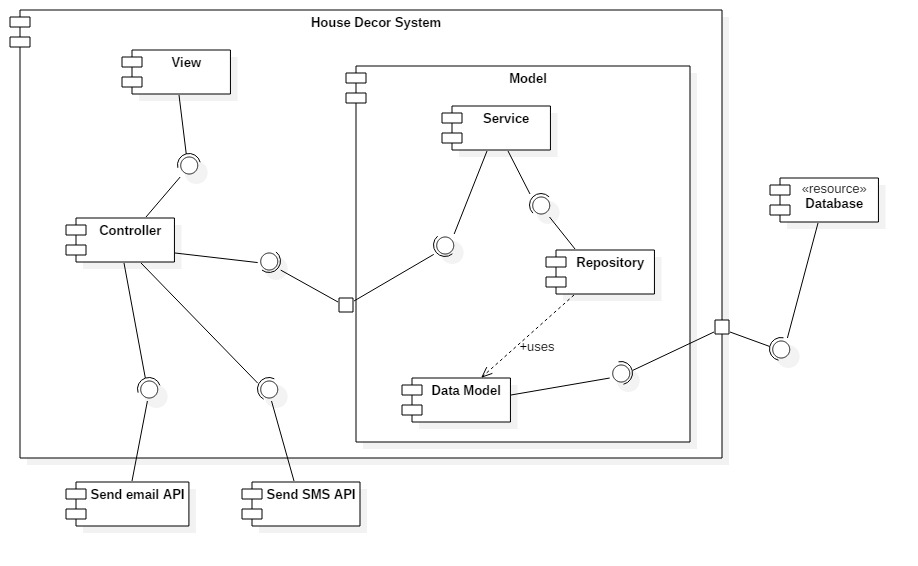


Figure 4: Web Service Architecture

|  |  |
| --- | --- |
| **Name** | **Description** |
| Controller | Common component used to handle request. |
| Service | Common component used to handle system’s business operations. |
| Repository | Component used to handle interaction between the system and database. |
| Data Model | Class entity |
| Database | Support storage and manipulate data. |
| Send email API | Provide API for send email. |
| Send SMS API | Provide API for send SMS. |

Table 12: Component Dictionary

## Detailed Description

### Class Diagram

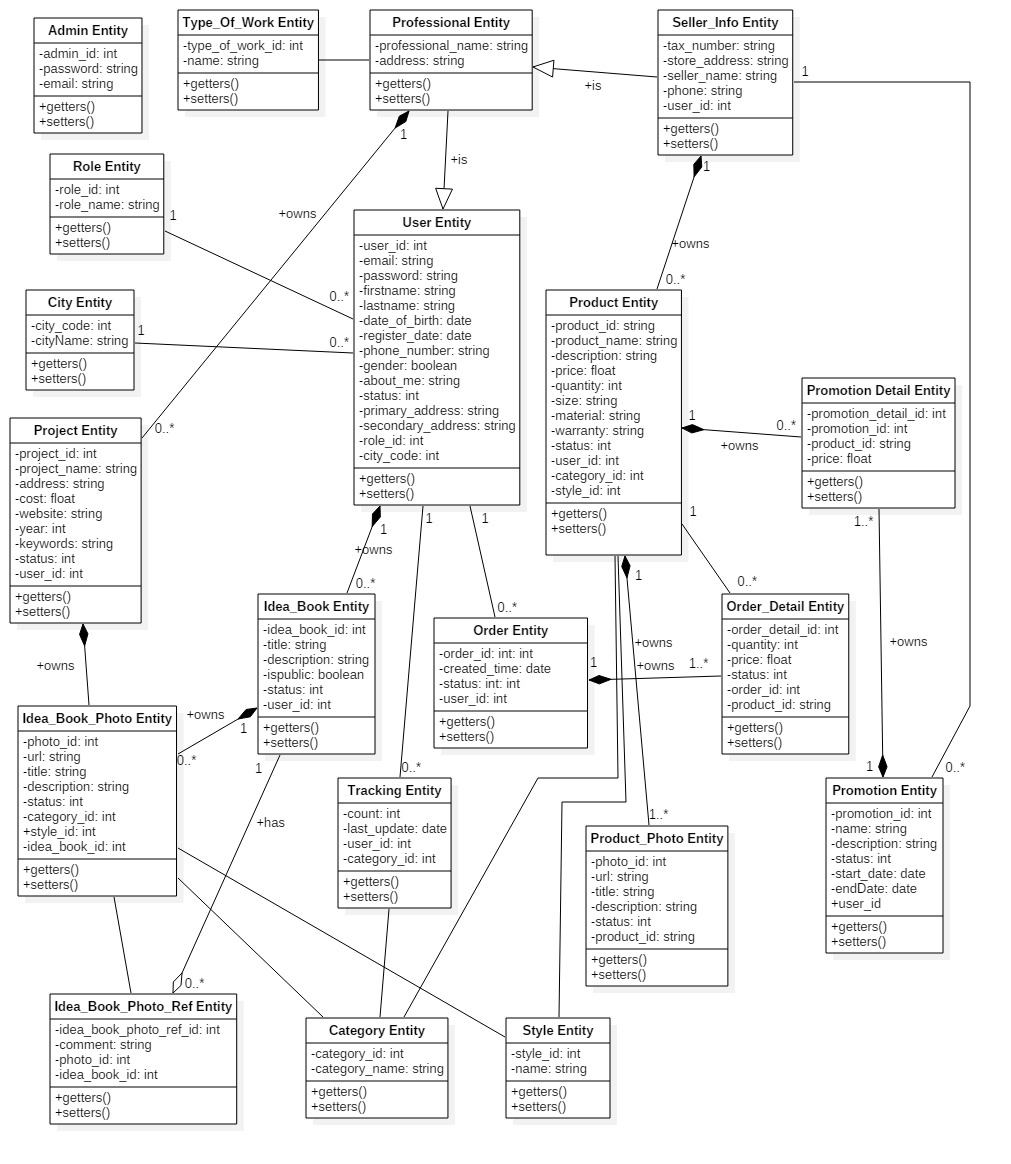


Figure 5: Web Service Architecture

|  |  |  |
| --- | --- | --- |
| **Class name** | **Mapping column with conceptual diagram** | **Description** |
| Admin Entity | Admin | Admin’s accounts in the system. |
| Professional Entity | Account | Professional’s information in the system. |
| User Entity | User | User’s accounts in the system. |
| Role Entity | Role | Roles in the system. |
| Order Entity | Order | Orders in the system. |
| Order Detail Entity | Order Detail | Order’s details in the system. |
| City Entity | City | Cities in the system. |
| Tracking Entity | Tracking | Tracking user’s activities in the system. |
| Category Entity | Category | Categories in the system. Example of category are . |
| Idea\_Book\_Photo Entity | Idea Book Photo | Idea book and project photos in the system. |
| Idea\_Book\_Photo\_Ref Entity | Idea Book Photo Ref | Photos are saved from other idea books, projects in the system. |
| Project Entity | Project | Projects in the system. |
| Idea\_Book Entity | Idea Book | Idea books in the system. |
| Product\_Photo Entity | Product Photo | Product’s photos in the system. |
| Type\_Of\_Work Entity | Type of Work | Types of work in the system. |
| Seller\_Info\_Entity | Seller Info | Seller’s informations in the system. |
| Promotion Entity | Promotion | Promotions in the system. |
| Promotion\_Detail Entity | Promotion Detail | Promotion details in the system. |
| Product Entity | Product | Products in the system. |
| Style Entity | Style | Styles in the system. |

Table 13: Class Diagram Dictionary

### Class Diagram Explanation

#### Admin

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique id of an account. |
| password | String | Private | Password of an account. |
| email | String | Private | Email of an account. |
| Table 13: Admin class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 14: Admin class methods explanation | | | |

#### Professional

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| nameProfessional | String | Private | Unique id of an professional. |
| address | String | Private | Address of an professional. |
| userId | Integer | Private | Id of user. |
| TypeOfWorkId | Integer | Private | Id of TypeOfWork |
| Table 15: Professional class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 16: Professional method methods explanation | | | |

#### User

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | Integer | Private | Unique id of user role. |
| email | String | Private | Email of an user. |
| Password | String | Private | Password of an user. |
| Firstname | String | Private | Firstname of an user. |
| Lastname | String | Private | Lastname of an user. |
| DateOfBirth | Date | Private | Birthdate of an user. |
| RegisterDate | Date | Private | Register date of an user. |
| PhoneNumber | String | Private | Phone number of an user. |
| RoleId | Integer | Private | Id of role. |
| Gender | Integer | Private | Gender of an user. |
| AboutMe | String | Private | Description about user. |
| Status | Integer | Private | Status of an user. |
| PrimaryAddress | String | Private | Primary address of an user. |
| SecondAddress | String | Private | Second address of an user. |
| cityCode | Integer | Private | Code of city |
| Table 17: User class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 18: User class methods explanation | | | |

#### Role

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| RoleId | Integer | Private | Unique id of a role. |
| RoleName | String | Private | Name of a role. |
| Table 19: Role class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 20: Role class methods explanation | | | |

#### Order

**Attributes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | | **Type** | | **Visibility** | | **Description** |
| OrderId | | Integer | | Private | | Unique id of a order. |
| CreatedTime | | Date | | Private | | Created time of an order. |
| Status | Integer | | Private | | Status of an order. | |
| UserId | Integer | | Private | | Id of user. | |
| Table 21: Order class attributes explanation | | | | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 22: Order class methods description | | | |

#### Order Detail

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| OrderDetailId | Integer | Private | Unique id of an OrderDetail. |
| ProductId | Integer | Private | Id of product. |
| Quantity | Integer | Private | Quantity of an product. |
| Price | Float | Private | Price of an product. |
| StoreId | Integer | Private | Id of store. |
| Status | Integer | Private | Status of an OrderDetail. |
| OrderId | Integer | Private | Id of an order. |
| Table 23: Order Detail class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 24: Order Detail class methods explanation | | | |

#### City

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| CityCode | Integer | Private | Unique id of a City. |
| CityName | String | Private | Name of a city. |
| Table 25: City class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 26: City class methods explanation | | | |

#### Tracking

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Count | Integer | Private | Unique id of an author. |
| LastUpdate | Date | Private | Last update date. |
| UserId | Integer | Private | Id of an user. |
| CategoryId | Integer |  | Id of a category |
| Table 27: Tracking class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 28: Tracking class methods explanation | | | |

#### Category

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| CategoryId | Integer | Private | Unique id of a category. |
| CategoryName | String | Private | Name of a category. |
| Table 29: Category class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 30: Category class methods explanation | | | |

#### Idea Book Photo

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| PhotoId | Integer | Private | Unique id of a Photo. |
| Url | String | Private | Url of a photo. |
| Title | String | Private | Title of a photo. |
| Description | String | Private | Description of a photo |
| Status | Integer | Private | Status of a photo. |
| ProjectId | Integer | Private | Id of a project. |
| CategoryId | Integer | Private | Id of a categegory. |
| StyleId | Integer | Private | Id of a style. |
| Table 31: Photo class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 32: Photo class methods explanation | | | |

#### Project

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ProjectId | Integer | Private | Unique id of a Project. |
| ProjectName | String | Private | Name of a project. |
| Address | String | Private | Address of a project. |
| Cost | Float | Private | Cost of a project. |
| Website | String | Private | Website of a project. |
| Year | Integer | Private | Year of a project |
| Keywords | String | Private | Keywords of a project. |
| Status | Integer | Private | Status of a project. |
| UserId | Integer | Private | Id of an user. |
| Table 33: Project class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 34: Project class methods explanation | | | |

#### Idea Book

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| IdeabookId | Integer | Private | Unique id of a idea book. |
| Title | String | Private | Title of an idea book. |
| Description | String | Private | Description of an idea book. |
| IsPublic | Integer | Private | Idea book is public or not. |
| Status | Integer | Private | Status of an idea book. |
| UserId | Integer | Private | Id of an user. |

Table 35: Idea book class attributes explanation

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

Table 36: Idea book class methods explanation

#### Product Photo

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ProductPhotoId | Integer | Private | Unique id of a product’s photo. |
| Url | String | Private | Url of a product’s photo. |
| Title | String | Private | Title of a product’s photo. |
| Description | String | Private | Description of a product’s photo. |
| Status | Integer | Private | Status of a product’s photo. |
| Product\_ProductId | Integer | Private | Id of a product |
| Table 37: Product Photo class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 38: Product Photo method methods explanation | | | |

#### Type of Work

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Private | Unique id of a type of work. |
| Name | String | Private | Name of a type of work. |
| Table 39: Type Of Work class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 40: Type Of Work method methods explanation | | | |

#### Seller

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| TaxNumber | String | Private | Unique id of a seller. |
| StoreAddress | String | Private | Store address. |
| SellerName | String | Private | Name of a seller. |
| StartDate | Date | Private | Date seller open store on system. |
| DueDate | Date | Private | Date seller close store on system. |
| Phone | String | Private | Phone of a seller. |
| UserId | Integer | Private | Id of an user. |
| Table 41: Seller class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 42: Seller method methods explanation | | | |

#### Promotion

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Private | Unique id of a promotion. |
| Name | String | Private | Name of a promotion. |
| Description | String | Private | Description of a promotion. |
| Status | Integer | Private | Status of a promotion. |
| StartDate | Date | Private | Date promotion start on system. |
| EndDate | Date | Private | Date promotion end on system. |
| userId | Integer | Private | Id of a user. |
| Table 43: Promotion class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 44: Promotion method methods explanation | | | |

#### Product

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ProductId | Integer | Private | Unique id of a Product. |
| ProductName | String | Private | Name of a product. |
| Desciption | String | Private | Desciption of a product. |
| Price | Float | Private | Price of a product. |
| Quantity | Integer | Private | Quantity of a product. |
| Size | String | Private | Size of a product. |
| Material | String | Private | Material of a product. |
| Warranty | String | Private | Warranty of a product. |
| Status | Integer | Private | Status of a product. |
| Style\_StyleId | Integer | Private | Id of a style. |
| Category\_CategoryId | Integer | Private | Id of a category. |
| Seller\_UserId | Integer | Private | Id of user is seller. |
| Table 45: Product class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 46: Product method methods explanation | | | |

#### Style

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| StyleId | Integer | Private | Unique id of a Style. |
| Name | String | Private | Name of a style. |
| Table 47: Style class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 48: Style method methods explanation | | | |

#### Promotion Detail

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| PromotionDetailId | Integer | Private | Unique id of a Promotion detail. |
| PromotionId | Integr | Private | Id of a promotion. |
| ProductId | String | Private | Id of a product. |
| Price | Float | Private | Discounted price. |
| Table 49: Promotion Detail class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 50: Promotion Detail method methods explanation | | | |

#### Idea Book Photo Ref

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| IdeabookPhotoRefId | Integer | Private | Unique id of an Idea book photo ref. |
| Comment | String | Private | Comment of a photo. |
| PhotoId | Integer | Private | Id of a photo. |
| IdeabookId | Integer | Private | Id of an idea book. |
| Table 51: Idea Book Photo Ref class attributes explanation | | | |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |
| Table 52: Idea Book Photo Ref method methods explanation | | | |

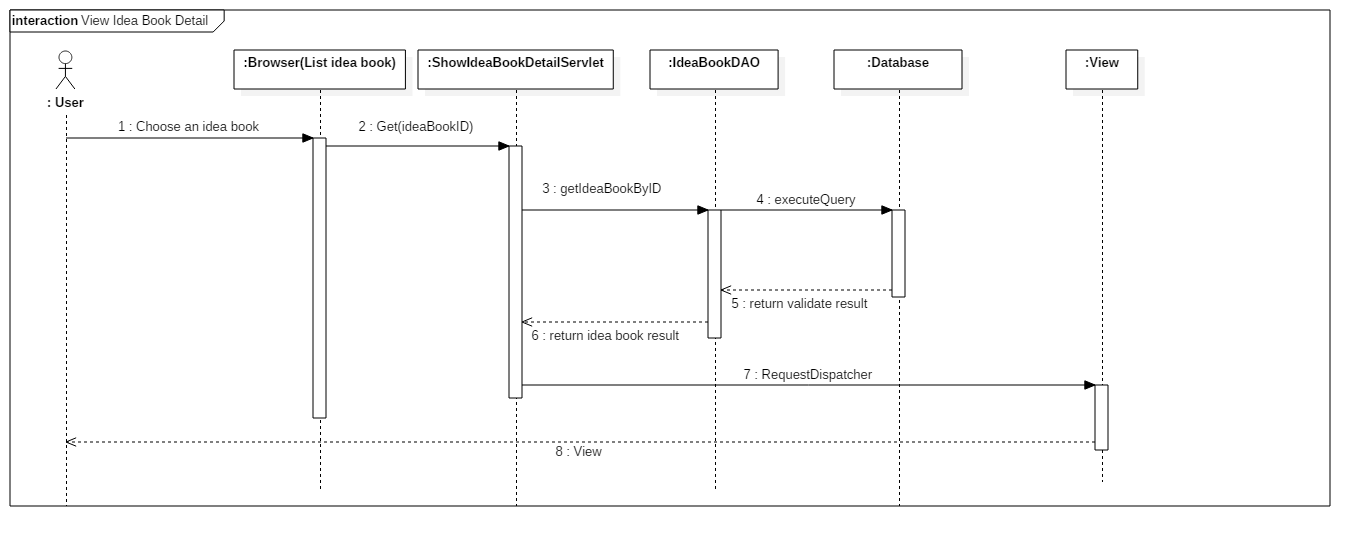
### Interaction Diagram

Web Application

#### Guest

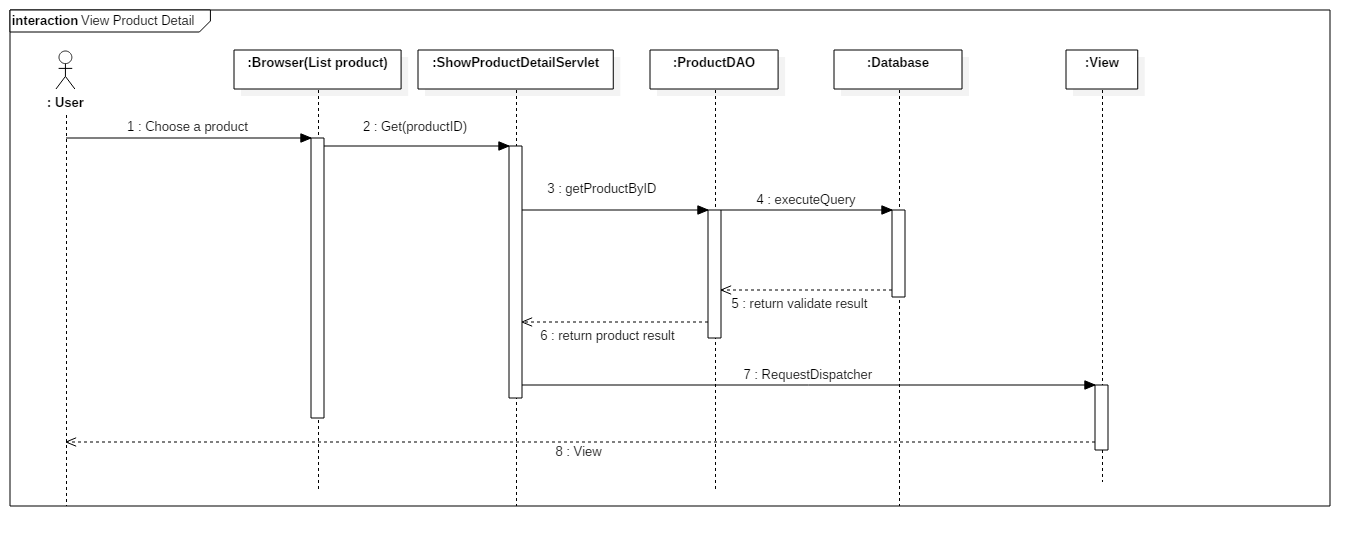
##### < Guest > View Idea Book Detail

Sumary: This diagram shows the process of load idea book return idea book detail.



##### < Guest > View Product Detail

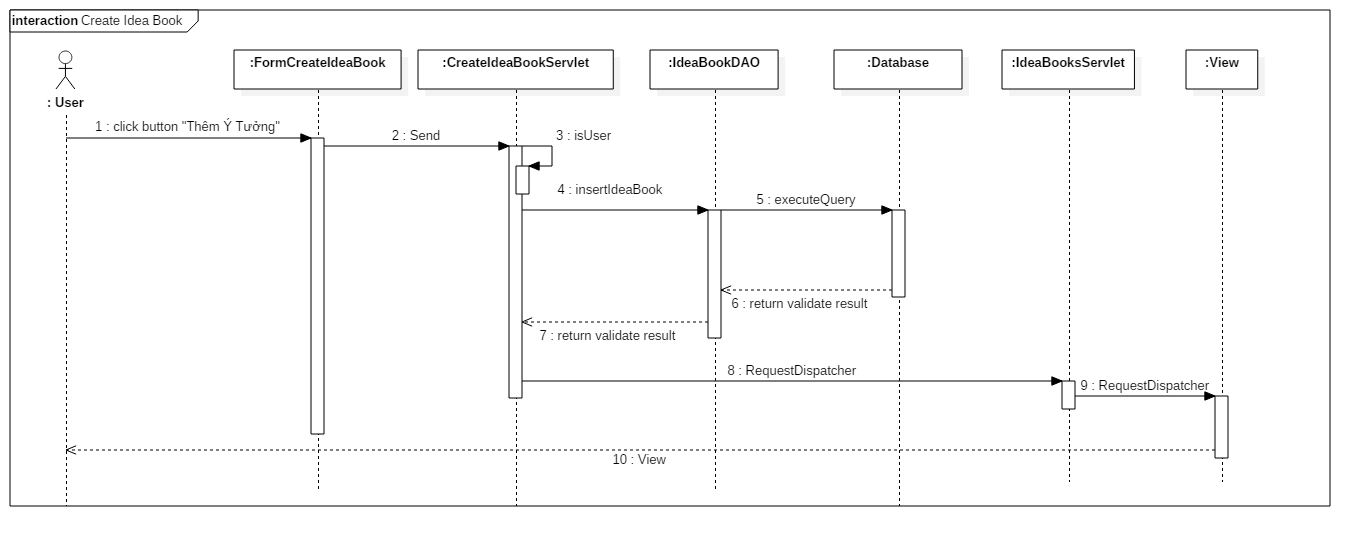
Sumary: This diagram shows the process of load product return product detail.



#### Member

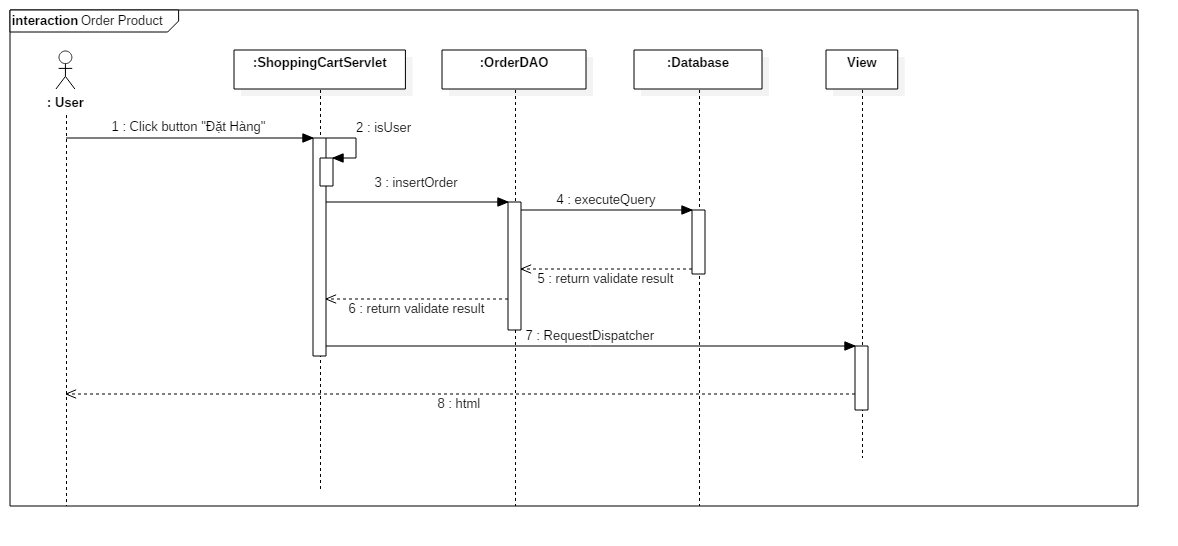
##### < Member > Create Idea Book

Sumary: This diagram shows the process of create idea book by user.



##### < Member > Order Product

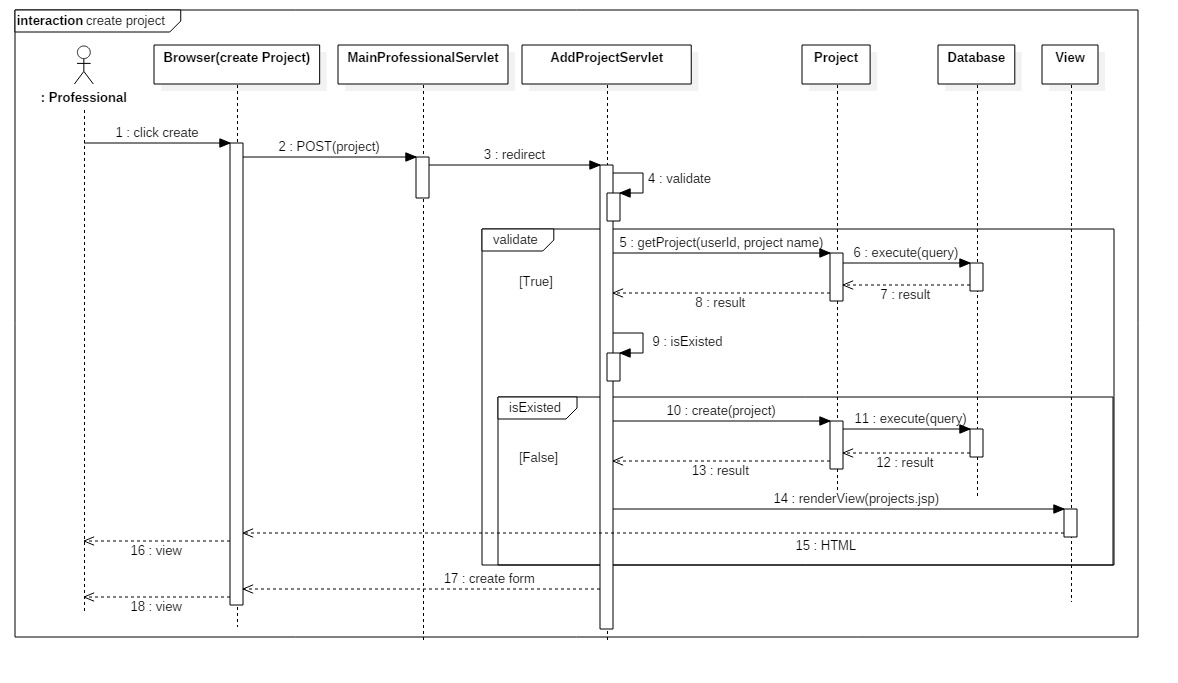
Sumary: This diagram shows the process of order product by user.



#### Professional

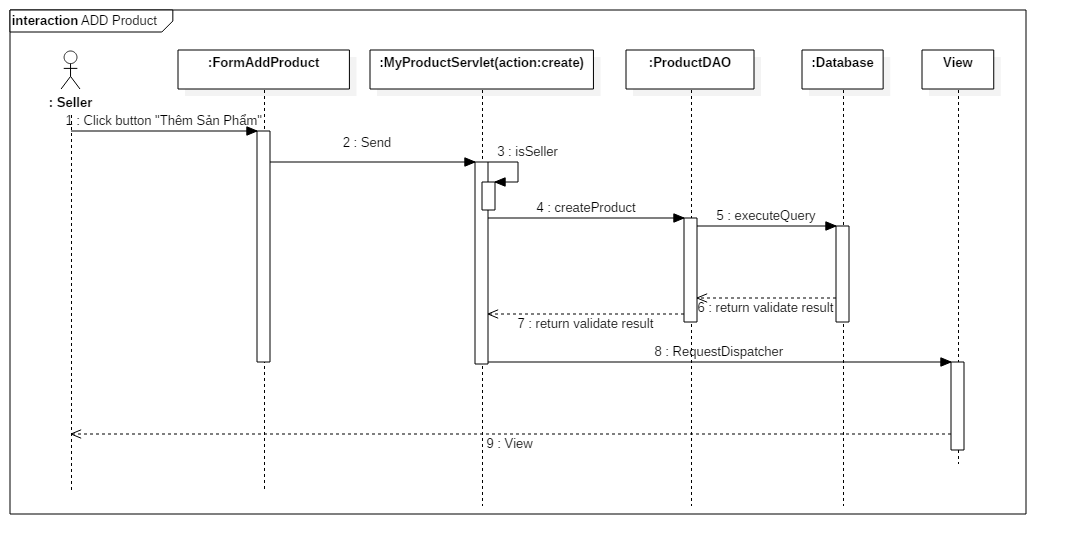
##### < Professional > Create project

Sumary: This diagram shows the process of create project by professional.



##### < Seller > Add Product

Sumary: This diagram shows the process of add product by seller.

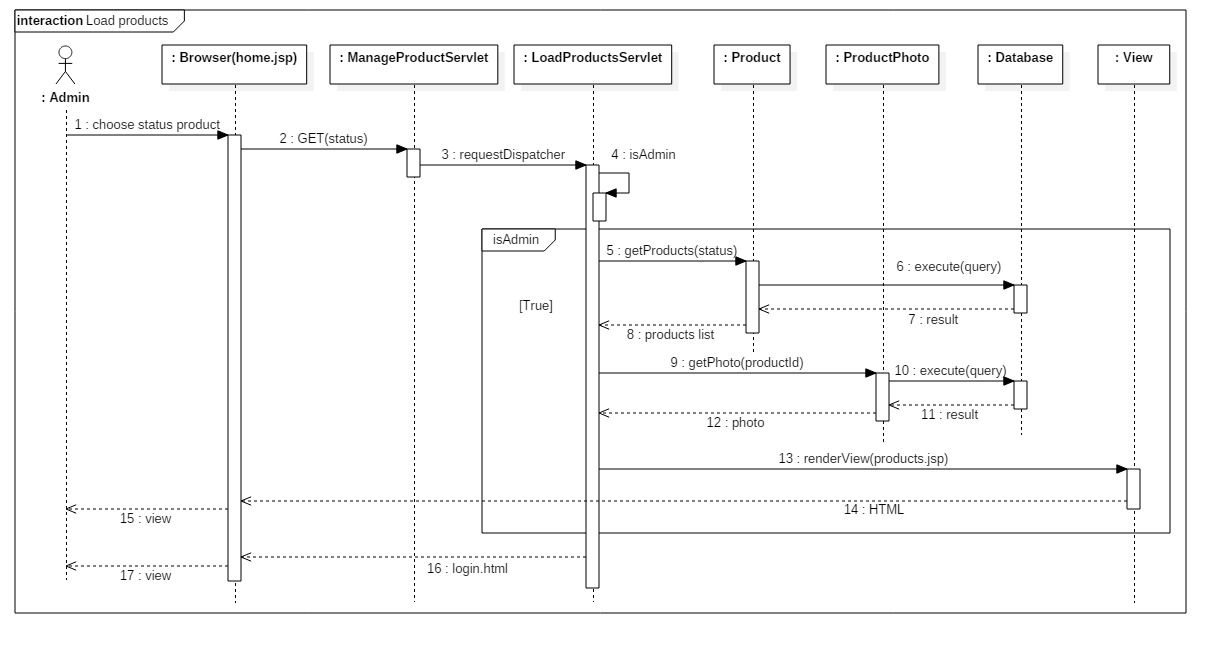


#### Admin

Sumary: The diagrams shows the process of load product list, view details about a product with product’s photos and update product’s status.

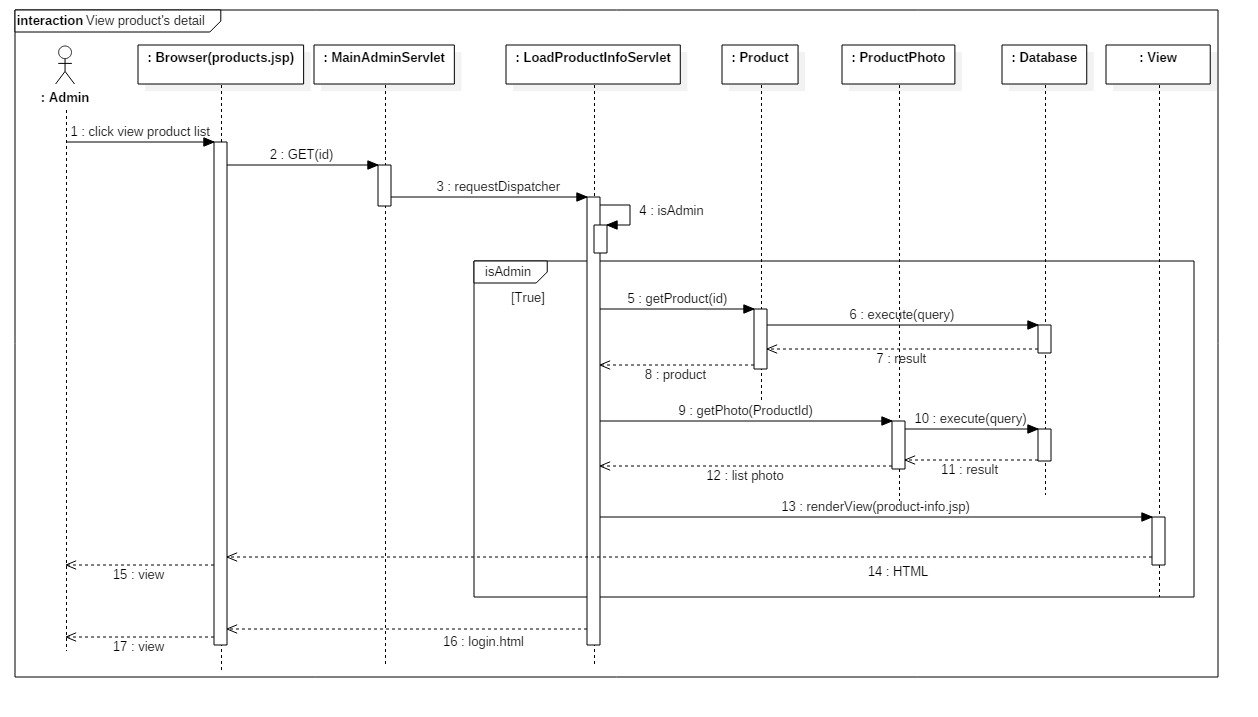
##### < Admin > Load Products

Sumary: This diagram shows the process of load product list return product list by product’s status.



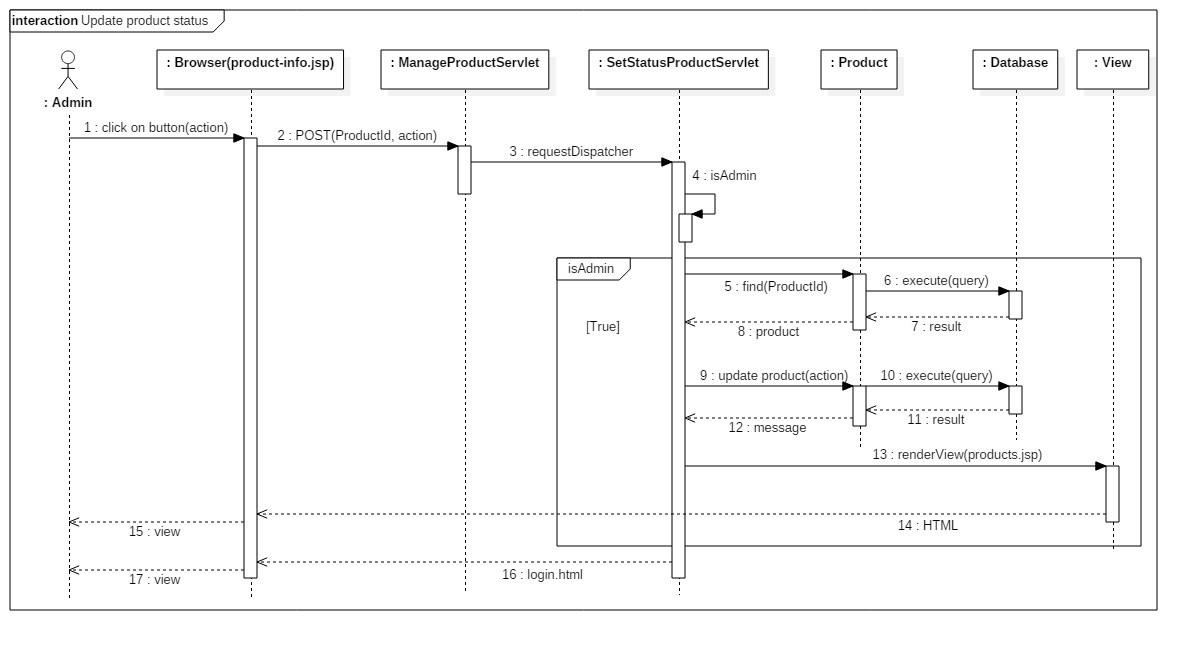
##### < Admin > View Product’s Detail

Sumary: This diagram shows the process of view product’s detail return product’s detail and product’s photos.



##### < Admin > Update Product’s Status

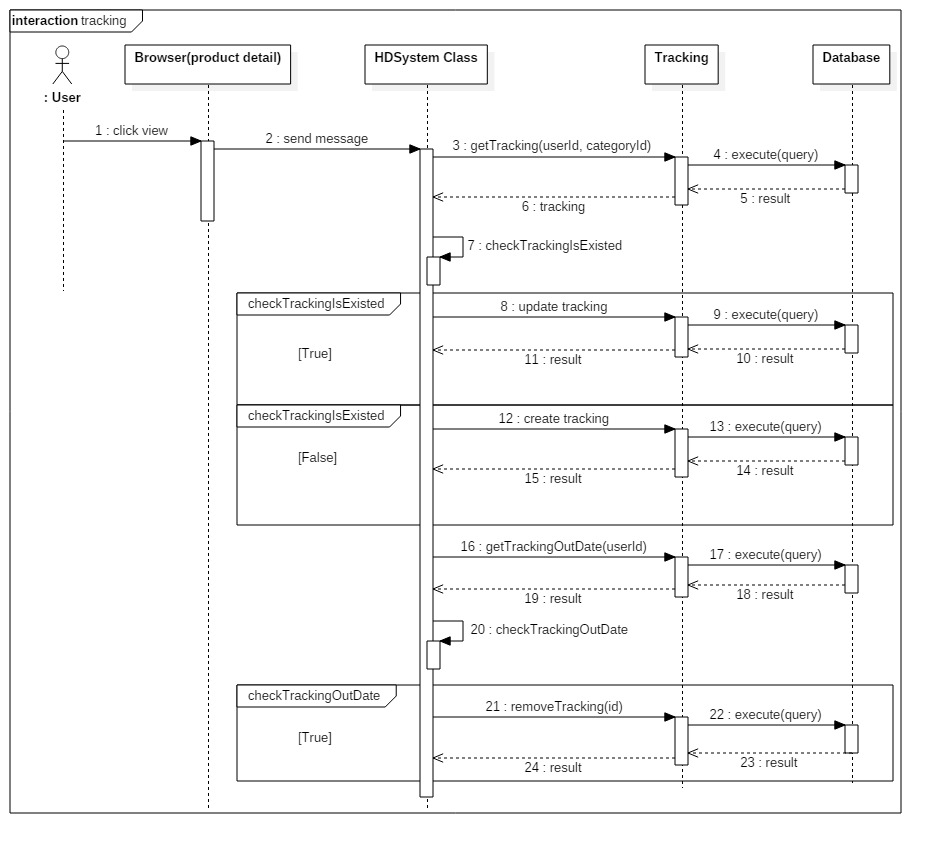
Sumary: This diagram shows the process of update product’s status.



#### System

##### < System > Tracking

Sumary: This diagram shows the process of tracking in system.



## Database Design

### Entity Relationship Diagram (ERD)

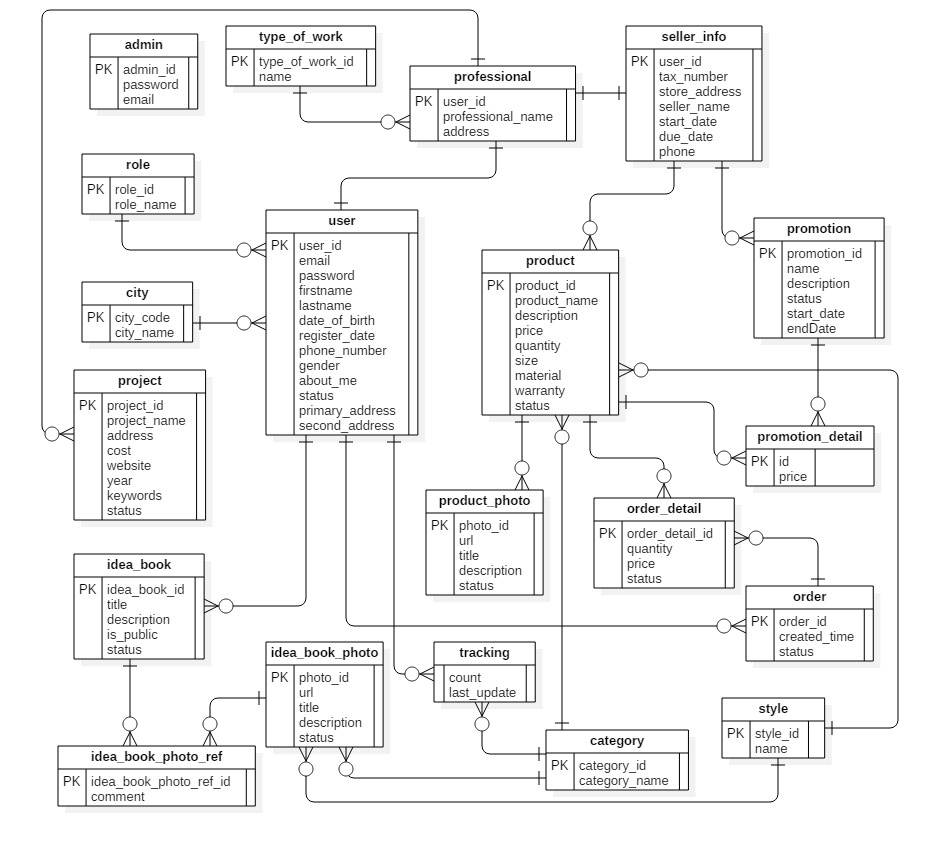


Figure 6: Entity Relationship Diagram (ERD)

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data Dictionary: describe contents of all entities** | |
| **Entity Name** | **Description** |
| User | All user accounts in the system. |
| Professional | All professional accounts in the system. |
| Admin | All admin accounts in the system. |
| Tracking | All activities of users in the system. |
| Idea\_book\_photo\_ref | All accepted idea book/project’s photos in the system. |
| Idea\_book | All idea books in the system. |
| Order\_detail | Detail information of all orders in the system. |
| Idea\_book\_photo | All idea book/project’s photo in the system. |
| Type\_of\_work | All type of work in the system. |
| Role | All role in the system. |
| Promotion | All promotion of products in the system. |
| City | All city in the system. |
| Seller info | All sellers in the system. |
| Product\_photo | All product’s photos in the system. |
| Product | All products in the system. |
| Project | All projects in the system. |
| Order | All orders in the system. |
| Style | All styles in the system. |
| Category | All categories in the system. |
| Promotion\_detail | All promotion details in the system. |

## Algorithms

**Approve an idea book: (parameter: Idea book Id)**

Load all photos of the idea book by idea book id in “Idea Book Photo” table

For all photos in photo list

If photo’s status is waiting for approval then

Output: Error message

End for

End if

End for

Load the idea book by idea book id

Update idea book’s status to approved

**Tracking: (parameters: user Id, category Id)**

Load tracking by user Id and category Id

If tracking is existed then

Update this tracking

Else

Create new tracking

Save new tracking to Database

End if

Load tracking is out of date by user Id

If exist tracking is out of date then

Remove tracking out of date

End if

**Send promotion emails: (parameter: product Id)**

Load product by product Id

Load category Id of product

Use category Id to find users matched with in “Tracking” table

If users are not empty then

Load emails of founded users.

Send promotion email to users email

End if

**Send order to seller: (parameter: order Id)**

Load order by order Id

Load products in “Order detail” table by order

For all products in the order

Load sellers provide product by product Id

End for

For all sellers in list loaded above

For all products in the order

Load products, quantity are matched with seller

End for

Send email to seller

End for.

# Report 5 - System Implementation & Test

## Introduction

### Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of House Decor system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

## Database Relationship Diagram

### Physical Diagram

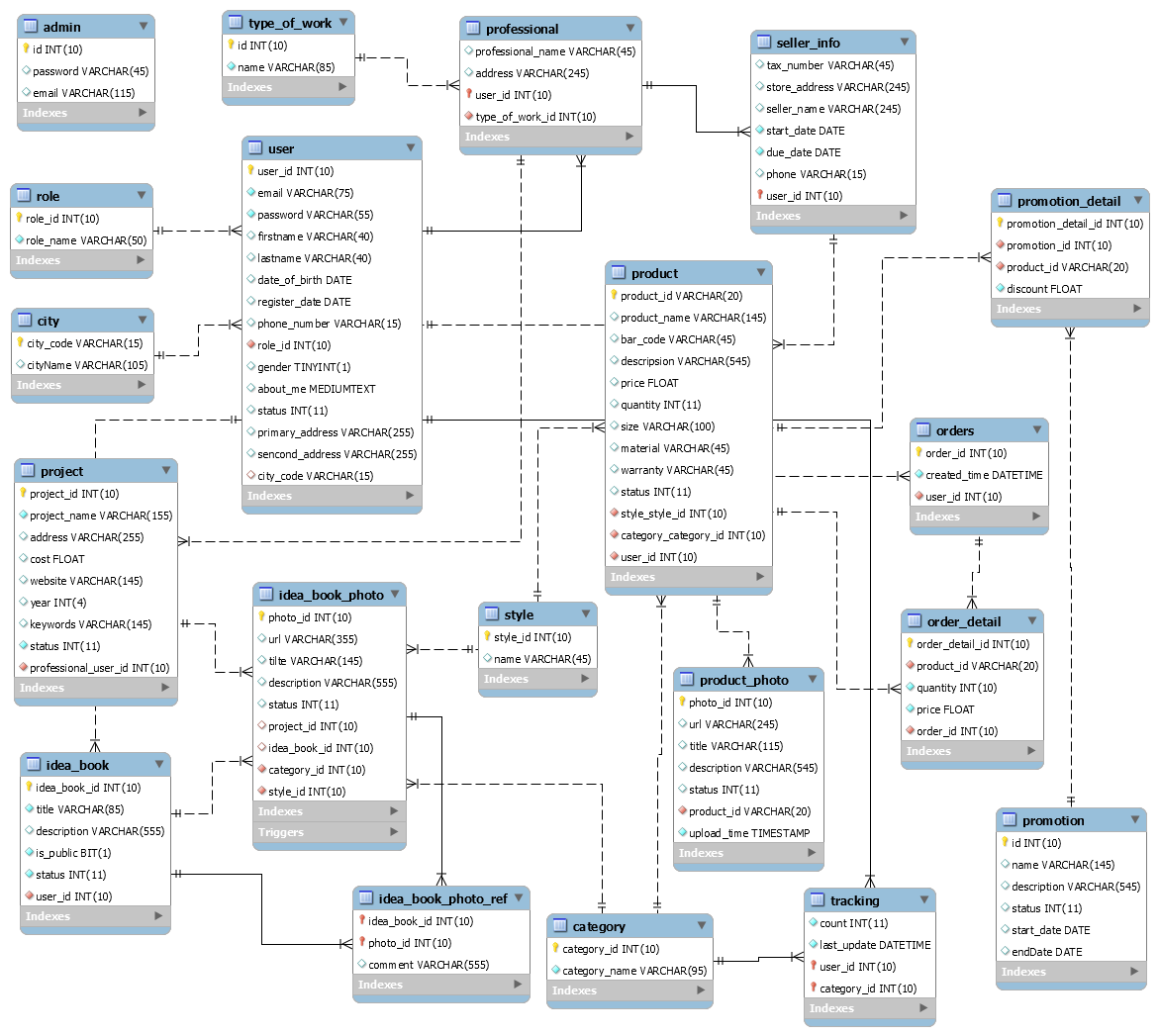


Figure 7: <Database Relation Diagram> Physical Diagram

### Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Data dictionary: describe content of all tables** | | |
| **Table Name** | **Mapping column with Conceptual diagram** | **Description** |
| **Admin** |  | Admin’s accounts in the system. |
| **Professional** | Type\_Of\_Work, Project, Seller\_Info, Project | Professional’s information in the system. |
| **User** | Role, Professional, Idea\_Book, City, Order, Tracking | User’s accounts in the system. |
| **Role** | User | Roles in the system. |
| **Orders** | User, Order\_Detail | Orders in the system. |
| **Order\_Detail** | Order, Product | Order’s details in the system. |
| **City** | User | Cities in the system. |
| **Tracking** | User, Category | Tracking user’s activities in the system. |
| **Category** | Tracking, Product, Idea\_Book\_Photo | Categories in the system. Example of category are . |
| **Idea\_Book\_Photo** | Project, Idea\_Book, Idea\_Book\_Photo\_Ref, Category, Style | Idea book and project photos in the system. |
| **Project** | Idea\_Book\_Photo, Professional | Projects in the system. |
| **Idea\_Book** | Idea\_Book\_Photo, User, Idea\_Book\_Ref | Idea books in the system. |
| **Product\_Photo** | Product | Product’s photos in the system. |
| **Type\_Of\_Work** | Professional | Types of work in the system. |
| **Seller\_Info** | Professional, Product | Seller’s informations in the system. |
| **Promotion** | Promotion\_Detail, Product\_Has\_Promotion | Promotions in the system. |
| **Product** | Product\_Photo, Seller\_Info, Category, Style | Products in the system. |
| **Style** | Product, Idea\_Book\_Photo | Styles in the system. |
| **Promotion\_Detail** | Product, Promotion | Promotion’s detail in the system. |
| **Idea\_Book\_Photo\_Ref** | Idea\_Book, Idea\_Book\_Photo | Photo save from others in the system. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Attributes** | **Description** | **Domain** | **Null** |
| **Admin** | id | Unique id of an account. | Integer | No |
| password | Password of an account. | String | No |
| email | Email of an account. | String | No |
| **Professional** | nameProfessional | Name of an professional. | String | No |
| address | Address of an professional. | String | No |
| userId | Id of user. | Integer | No |
| TypeOfWorkId | Id of TypeOfWork | Integer | No |
| **User** | userId | Unique id of user role. | Integer | No |
| email | Email of an user. | String | No |
| Password | Password of an user. | String | No |
| Firstname | Firstname of an user. | String | No |
| Lastname | Lastname of an user. | String | No |
| DateOfBirth | Birthdate of an user. | Date | No |
| RegisterDate | Register date of an user. | Date | No |
| PhoneNumber | Phone number of an user. | String | No |
| RoleId | Id of role. | Integer | No |
| Gender | Gender of an user. | Integer | No |
| AboutMe | Description about user. | String | Yes |
| Status | Status of an user. | Integer | No |
| PrimaryAddress | Primary address of an user. | String | Yes |
| SecondAddress | Second address of an user. | String | Yes |
| cityCode | Code of city | Integer | No |
| **Role** | RoleId | Unique id of a role. | Integer | No |
| RoleName | Name of a role. | String | No |
| **Order** | OrderId | Unique id of a order. | Integer | No |
| CreatedTime | Created time of an order. | Date | No |
| Status | Status of an order. | Integer | No |
| UserId | Id of user. | Integer | No |
| **Order Detail** | OrderDetailId | Unique id of an OrderDetail. | Integer | No |
| ProductId | Id of product. | Integer | No |
| Quantity | Quantity of an product. | Integer | No |
| Price | Price of an product. | Float | No |
| StoreId | Id of store. | Integer | No |
| Status | Status of an OrderDetail. | Integer | No |
| OrderId | Id of an order. | Integer | No |
| **City** | CityCode | Unique id of a City. | Integer | No |
| CityName | Name of a city. | String | No |
| **Tracking** | Count | Unique id of an author. | Integer | No |
| LastUpdate | Last update date. | Date | No |
| UserId | Id of an user. | Integer | No |
| CategoryId | Id of a category | Integer | No |
| **Category** | CategoryId | Unique id of a category. | Integer | No |
| CategoryName | Name of a category. | String | No |
| **Idea\_Book\_Photo** | PhotoId | Unique id of a Photo. | Integer | No |
| Url | Url of a photo. | String | No |
| Title | Title of a photo. | String | No |
| Description | Description of a photo | String | Yes |
| Status | Status of a photo. | Integer | No |
| ProjectId | Id of a project. | Integer | Yes |
| IdeabookId | Id of an idea book | Integer | Yes |
| CategoryId | Id of a categegory. | Integer | No |
| StyleId | Id of a style. | Integer | No |
| **Project** | ProjectId | Unique id of a Project. | Integer | No |
| ProjectName | Name of a project. | String | No |
| Address | Address of a project. | String | No |
| Cost | Cost of a project. | Float | Yes |
| Website | Website of a project. | String | Yes |
| Year | Year of a project | Intger | Yes |
| Keywords | Keywords of a project. | String | Yes |
| Status | Status of a project. | Integer | No |
| UserId | Id of an user. | Integer | No |
| **Idea book** | IdeabookId | Unique id of a idea book. | Integer | No |
| Title | Title of an idea book. | String | No |
| Description | Description of an idea book. | String | Yes |
| IsPublic | Idea book is public or not. | Integer | No |
| Status | Status of an idea book. | Integer | No |
| UserId | Id of an user. | Integer | No |
| **Product Photo** | ProductPhotoId | Unique id of a product’s photo. | Integer | No |
| Url | Url of a product’s photo. | String | No |
| Title | Title of a product’s photo. | String | No |
| Description | Description of a product’s photo. | String | Yes |
| Status | Status of a product’s photo. | Integer | No |
| ProductId | Id of a product | String | No |
| **Type Of Work** | Id | Unique id of a type of work. | Integer | No |
| Name | Name of a type of work. | String | No |
| **Seller** | TaxNumber | Unique id of a seller. | String | No |
| StoreAddress | Store address. | String | No |
| SellerName | Name of a seller. | String | No |
| StartDate | Date seller open store on system. | Date | No |
| DueDate | Date seller close store on system. | Date | No |
| Phone | Phone of a seller. | String | No |
| UserId | Id of an user. | Integer | No |
| **Promotion** | Id | Unique id of a promotion. | Integer | No |
| Name | Name of a promotion. | String | No |
| Description | Description of a promotion. | String | No |
| Status | Status of a promotion. | Integer | No |
| StartDate | Date promotion start on system. | Date | No |
| EndDate | Date promotion end on system. | Date | No |
| **Product** | ProductId | Unique id of a Product. | String | No |
| ProductName | Name of a product. | String | No |
| Desciption | Desciption of a product. | String | No |
| Price | Price of a product. | Float | No |
| Quantity | Quantity of a product. | Integer | No |
| Size | Size of a product. | String | Yes |
| Material | Material of a product. | String | Yes |
| Warranty | Warranty of a product. | String | Yes |
| Status | Status of a product. | Integer | No |
| CategoryId | Id of category | Integer | No |
| StyleId | Id of style | Integer | No |
|  | Seller\_UserId | Id of user is seller. | Integer | No |
| **Style** | StyleId | Unique id of a Style. | Integer | No |
| Name | Name of a style. | String | No |
| **Promotion Detail** | PromotionDetailId | Unique id of a Promotion detail. | Integer | No |
| PromotionId | Id of promotion | Integer | No |
| Productid | Id of product | String | No |
| Price | Price of product | Float | Yes |
| **Idea\_Book\_Photo\_Ref** | IdeaBookId | Id of idea book | Integer | No |
| PhotoId | Id of photo | Integer | No |
| Comment | Comment | String | Yes |

Table 53: Data dictionary: Describe content of all tables