|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

|  |  |
| --- | --- |
| **FPT UNIVERSITY** | |
| CAPSTONE PROJECT DOCUMENT | |
| **HANGING PRODUCT STORE** | |
| **SE07.B – GROUP 8** | |
| **Group Members** | |  |  |  | | --- | --- | --- | | Quản Tiến Dân | Team Leader | DanQTSE61144 | | Trần Quang Phúc | Team Member | Dropped out | | Nguyễn Huy Hoàng | Team Member | HoangNHSE61007 | | Huỳnh Công Đức | Team Member | DucHCSE61024 | |
| **Supervisor** | Kiều Trọng Khánh |
| **Ext.Supervisor** | N/A |
| **Capstone Project code** | HPS |

- Ho Chi Minh City, May/2015 -

*This page is intentionally left blank*

**

**CAPSTONE PROJECT REGISTER**

Class: Duration time: from 11/05/2015…. To /2015…..

(\*) Profession: <Software Engineer> Specialty: <ES> <IS>

x

(\*) Kinds of person make registers: Lecturer Students

x

1. Register information for supervisor (if have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **E-Mail** | **Title** |
| Supervisor 1 | Kiều Trọng Khánh |  | khanhkt@fpt.edu.vn | Mr. |

2. Register information for students (if have)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Full name** | **Student code** | **Phone** | **E-mail** | **Role in Group** |
| Student 1 |  |  |  |  |  |
| Student 2 |  |  |  |  |  |
| Student 3 |  |  |  |  |  |
| Student 4 |  |  |  |  |  |

3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

English: Hanging Product Store.

Vietnamese: Cửa hàng bán giúp hàng hóa ký gửi

Abbreviation:

- HPS

- Building the application supports the store owner manage the deposit product. The owner will be posted his/her website to trade.

- Building the application provides following services

* **Manage Deposit Product**
* **Manage the transaction**

(\*) 3.2. Main proposal content (including result and product)

1. Theory and practice (document):

* Student should apply the software development process and the UML
* Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
* 3 tiers should be applied
* Server side technique:
  + Database design, OOA, OOD, OOP, MVC, Java or .Net technology, …
* Client side technique
  + HTML5, CSS, JavaScript, JQuery, Ajax, Androids …
* Communication technique
  + Exchange information and transfer data in effective in networks, communicating protocol between mobile device, ...
* Research
  + Principal of Accounting

1. Program:

* Main functions
* **Manage Deposit Product**
* **Manage the transaction**
* Manage the status of the transaction of store owner and good owners
* **...**

1. Other products:

* All of management functions of the system must be implemented to support the operating system in best

4. Other comment (propose all relative thing if have)

* N/A

|  |  |
| --- | --- |
| **Supervisor (If have)**  *(Sign and full name)* | HCM city, date 14/4/2015 …..  **On behalf of Registers**  *(Sign and full name)* |

Table of Contents

[List of Tables 6](#_Toc427485497)

[List of figures 6](#_Toc427485498)

[A. Report No.1 Introduction 6](#_Toc427485499)

[1. Project Information 6](#_Toc427485500)

[2. Introduction 7](#_Toc427485501)

[3. Current Situation 7](#_Toc427485502)

[4. Problem Definition 7](#_Toc427485503)

[5. Proposed Solution 8](#_Toc427485504)

[5.1 Feature functions 8](#_Toc427485505)

[5.2 Advantages and disadvantages 8](#_Toc427485506)

[6. Functional Requirements 9](#_Toc427485507)

[7. Role and Responsibility 9](#_Toc427485508)

[B. Report No.2 Software Project Management Plan 10](#_Toc427485509)

[1. Problem Definition 10](#_Toc427485510)

[1.1 Name of this Capstone Project 10](#_Toc427485511)

[1.2 Problem abstract 10](#_Toc427485512)

[1.3 Project Overview 10](#_Toc427485513)

[C. Report No.3 Software Requirement Specification 11](#_Toc427485514)

[1. User Requirement Specification 11](#_Toc427485515)

[1.1 Guest Requirement 11](#_Toc427485516)

[1.2 Customer Requirement 11](#_Toc427485517)

[1.3 Store Owner Requirement 11](#_Toc427485518)

[1.4 Admin Requirement 12](#_Toc427485519)

[2. System Overview Use Case 13](#_Toc427485520)

[2.1 List of Use Case 14](#_Toc427485521)

[3. Conceptual Diagram 37](#_Toc427485522)

[D. Report No. 4 Software Design Description 38](#_Toc427485523)

[1. System Architectural Design 38](#_Toc427485524)

[1.1 Web Application Architecture Description 38](#_Toc427485525)

[1.2 Mobile Application Architecture Description 38](#_Toc427485526)

[2. Component Diagram 39](#_Toc427485527)

[3. Interaction Diagram 40](#_Toc427485528)

[3.1 Sequence Diagram 40](#_Toc427485529)

[3.2 Activity Diagram 43](#_Toc427485530)

[4. Algorithms 46](#_Toc427485531)

[4.1 Price Suggestion 46](#_Toc427485532)

[4.2 Consignment State 47](#_Toc427485533)

[E. Report No. 5 System Implement & Test Description 49](#_Toc427485534)

[1. Database Relationship Diagram 49](#_Toc427485535)

[1.1 Physical Diagram 49](#_Toc427485536)

# List of Tables

[Table 1: Role and Responsibility 7](#_Toc427481990)

# List of figures

[Figure 1: Overview Use Case 13](#_Toc427485476)

[Figure 2: <Guest> Consign product 14](#_Toc427485477)

[Figure 3: <Guest> Order Product 22](#_Toc427485478)

[Figure 4: <Store Owner> Accept Request 25](#_Toc427485479)

[Figure 5: <Store Owner> Receive Product 27](#_Toc427485480)

[Figure 6: <Store Owner> Publish On Web 29](#_Toc427485481)

[Figure 7: <Store Owner> Pay Consignor 32](#_Toc427485482)

[Figure 8: <System> Search Product Online 35](#_Toc427485483)

[Figure 9: Conceptual Diagram 37](#_Toc427485484)

[Figure 10: System Architectural Design 38](#_Toc427485485)

[Figure 11: Component Diagram 39](#_Toc427485486)

[Figure 12: Consign Product 40](#_Toc427485487)

[Figure 13: Order Product 41](#_Toc427485488)

[Figure 14: Accept Request 41](#_Toc427485489)

[Figure 15: Receive Product 42](#_Toc427485490)

[Figure 16: Publish On Web 43](#_Toc427485491)

[Figure 17: Pay Consignor 43](#_Toc427485492)

[Figure 18: Auto Synchronize 44](#_Toc427485493)

[Figure 19: Accept Product Activity Diagram 45](#_Toc427485494)

[Figure 20: Consignment State 47](#_Toc427485495)

[Figure 21: Physical Database 49](#_Toc427485496)

# Report No.1 Introduction

## Project Information

* **Project Name:** Hanging Product Store
* **Project Code:** HPS
* **Product Type:** Website and Android Application
* **Start Date:** May 11, 2015
* **End Date:** September 3, 2015

## Introduction

Nowadays, because of the shopping hobby, many people are willing to buy product even they just use it once. Therefore, they probably have items hanging in their closet that they never use again. Hence, consignment sales service appears.

Currently, in Viet Nam consignment services becomes more popular. There are some consignment sales services but they require consignor brings items to the store and some use manually process in management (give receipts to consignor for tracking).

From above mentioned reasons, this project is going to build an online system for consignment sales service.

## Current Situation

This kind of business is now becoming more popular in daily life, with variety genres of product, no capital. In other country likes US, TheRealReal is very successful. In Viet Nam, there are some consignment sales services such as LALA-Save&Share, khohanghoa.com… They are running their business and expanding their business very well.

## Problem Definition

Here is their advantages and disadvantages:

* LALA-Save&Share:
* Advantages:
* Provide online shopping method.
* Online support available.
* Support consigned-items tracking.
* Disadvantages:
* Doesn’t support online payment for consignor when consigned items are sold.
* Consignor has to bring theirs item to the store to consign.
* Support only fashion product.
* KhoHangHoa.com:
* Advantages:
* Support many types of product.
* Provide online shopping method.
* Support online payment for consignor when consigned-items are sold.
* Online support available.
* Disadvantages:
* Consignor has to bring theirs item to the store to consign.
* Consignor cannot track their consigned-items status immediately.
* TheRealReal.com:
* Advantages:
* Online support available.
* Provide online shopping method, online payment for both consignment and purchase.
* Support product recommendation by season or fashion trend and wish list.
* Support consigned-items tracking.
* Disadvantages:
* Support only fashion product.
* Consignor has to bring theirs item to the store to consign.

## Proposed Solution

Our system include:

* A web app allow multiple store register to receive consignment request, sell consigned product and manage consigned product. Guest and customer can make consignment request and tracking consigned product status.
* An android app helps store owner to receive product.

In more detail, our system has the following functions:

### Feature functions

* Web Application:
* Admin can create account and manage account.
* Guest can make product consignment request, search product and order product. The system support Guest tracking their product status, receive notification about their product status, payment online method and price suggestion.
* Customer can edit profile, search and review their transaction history.
* Every Store owner can manage their consignment request, raise their products on website and manage order product. The system support Store owner tracking consigned product status, payment online method, price suggestion.
* Mobile Application:
* StoreOwner can use their Android phone to receive or cancel product.

### Advantages and disadvantages

* Advantages:
* Support price suggestion.
* Support online payment method.
* Support notification about consigned product status.
* Support tracking product status.
* Support mobile app to receive or cancel product.
* Disadvantages:
* Support only fashion product.
* Internet connection is required.
* Price suggestion only support for original products.

## Functional Requirements

Function requirements of the system are listed as below:

* Consignment request management
* Receive or refuse product
* Raise products on website
* Order management
* Price suggestion
* Online payment method
* Product status notification
* Product status tracking
* Account management

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| **1** | Kiều Trọng Khánh | Senior Manager | Instructor | [KhanhKT@fpt.edu.vn](mailto:KhanhKT@fpt.edu.vn) |
| **2** | Quản Tiến Dân | Developer | Leader | [DanQTSE61144@fpt.edu.vn](mailto:DanQTSE61144@fpt.edu.vn) |
| **3** | Trần Quang Phúc | Developer | Member | [PhucTQSE60993@fpt.edu.vn](mailto:PhucTQSE60993@fpt.edu.vn) |
| **4** | Nguyễn Huy Hoàng | Developer | Member | [HoangNHSE61007@fpt.edu.vn](mailto:HoangNHSE61007@fpt.edu.vn) |
| **5** | Huỳnh Công Đức | Developer | Member | [DucHCSE61024@fpt.edu.vn](mailto:DucHCSE61024@fpt.edu.vn) |

Table : Role and Responsibility

# Report No.2 Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

Hanging Product Store (HPS)

### Problem abstract

With the increasing of shopping in community, the number of consignment store has started to increase too. A consignor has so many choices to consign his/her product and consignee may receive a lots of consigned product every day. Both consignor and consignee look for a way to perform business transaction online without worries. We propose a solution to solve this problem: a website where consignor and consignee can meet and do consignment business online. For consignor, we provide an easy way to get consignees’ attentions, make contract with them and track consigned product’s status. For consignee, we provide a useful tool to check the consigned products.

### Project Overview

#### Current Situation and Disadvantages

Although there are some shops running this service, most of them are not really effective in helping consignor tracking product’s status, negotiating price between consignor and consignee. When consignor has already consigned product, how could he/she know if consigned product is in store or is being sold? That problem goes to consignees as well. With many consigned products, how could they track each item efficiently? They also need a better way to automatically notify consignors when any actions is applied to consigned products.

#### The Proposed System

##### Web Site

* Admin can manage account
* Guest can make product consignment request.
* Guest can receive notification about their product status.
* Guest can track product status.
* Guest can search product.
* Guest can order product.
* Customer can edit profile
* Customer can review transaction history
* Store owner can receive suggested price for product.
* Store owner can receive or refuse product.
* Store owner can raise product on website.
* Store owner can manage consigned product.
* Store owner can manage order product.
* Store owner can have online payment method to pay for customer.
* Store owner can track consigned product status.
* System support send SMS to notify product’s status
* System support online payment method.
* System support price suggestion.

##### Mobile Application

* Store owner can using mobile to receive or refuse product.

#### Boundaries of System

* People who have computer, laptop or smartphone with internet connection and could run web browser can use this website.
* The complete product of this project includes:
* The website application.
* The mobile application.
* All related documents.
* Language: Vietnamese.

# Report No.3 Software Requirement Specification

## User Requirement Specification

### Guest Requirement

A guest of this system has the following abilities:

* Login
* Register
* Search product
* Consign product
* Order product
* Track consignment status or cancel consignment.

### Customer Requirement

A Customer of this system has the following abilities:

* Search product
* Consign product
* Order product
* Track consignment status or cancel consignment.
* Edit profile.
* View consignment history.

### Store Owner Requirement

A Store Owner of this system has the following abilities:

* View consignment requests.
* Accept or refuse consignment request.
* Receive or refuse product.
* View product status.
* Publish product on web.
* Pay consignor with online payment.
* Edit profile.

### Admin Requirement

An Admin of this system has the following abilities:

* Create user.
* De-active user.
* Active user.
* Update user profile.

## 

## System Overview Use Case

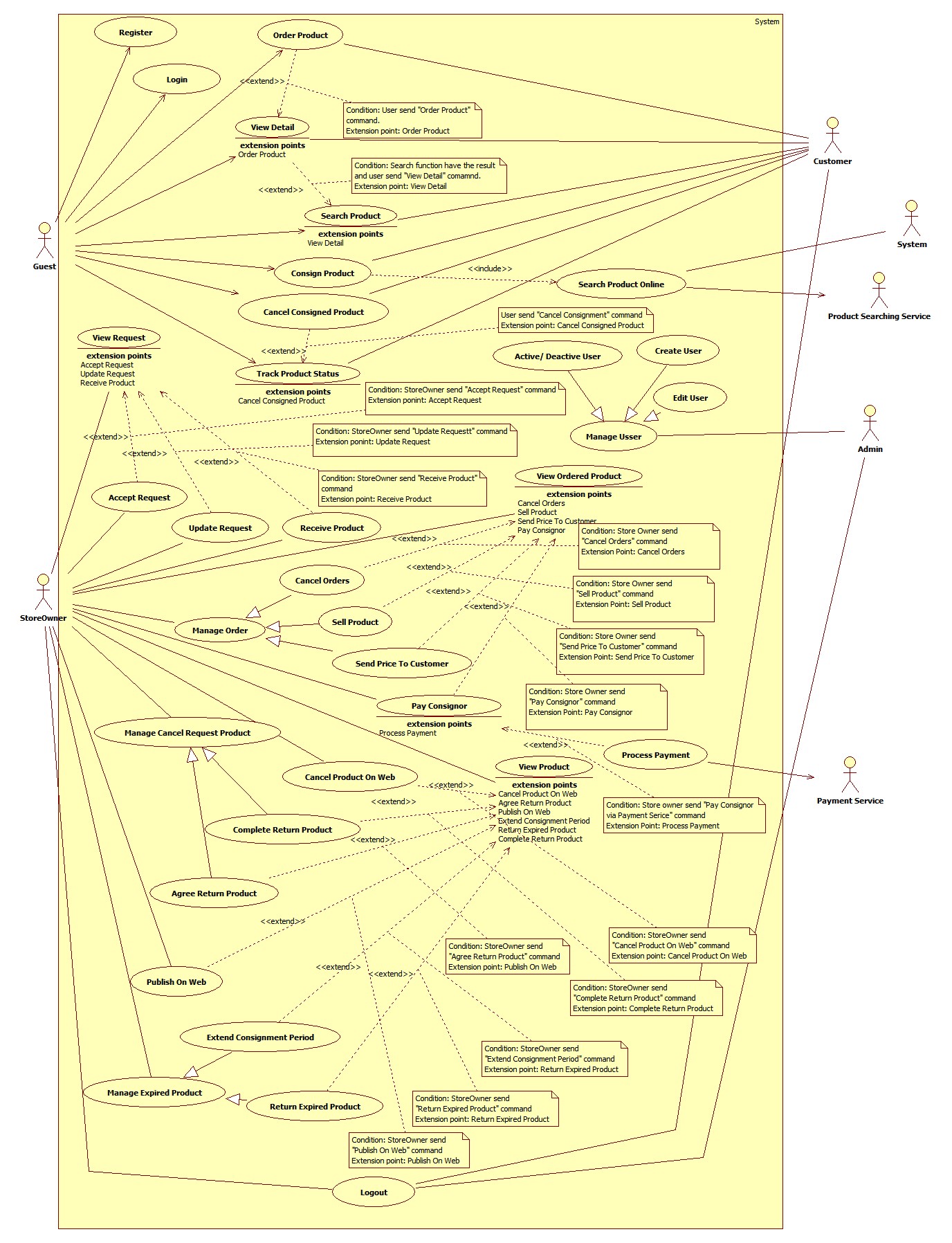


Figure : Overview Use Case

### List of Use Case

#### <Guest> Consign product

**Use Case Diagram**

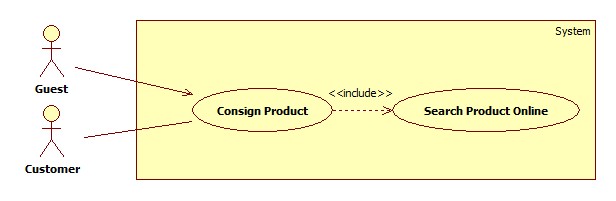


Figure : <Guest> Consign product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS001** | | | |
| **Use Case No.** | **HPS001** | Use Case Version | 2.0 |
| **Use Case Name** | Consign Product | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | May 23, 2015 | Priority | High |
| **Actor**:   * Guest * Customer   **Summary**:   * This use case allows user to send consigning-product request.   **Goal**:   * User sends consigning requests successfully.   **Triggers**:   * User sends command to consign product.   **Preconditions**: N/A  **Post Conditions**:   * **Success**: Show success message. * **Fail:** Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User sends command to consign product. | System requires user to input following information of product:   * “Chọn cách ký gửi”   + User chooses the way to check their information with 2 options: “Tên sản phẩm” and “Mã sản phẩm”.   + “Tên sản phẩm” is chosen by default. * “Tên sản phẩm”:   + Free text input.   + Min-length: 5 characters.   + Max-length: 100 characters.   + Required. * “Mã số”:   + Free text input.   + Required. * “Loại”:   + Option (example: Giày, Áo, Quần, Thắt lưng…).   + Required. * “Hãng”:   + Free text input.   + Max-length: 50 characters. * “Độ Mới”:   + Free text input.   + Format: number 1 -> 99   + Required. * “Mô tả”: free text input.   Systems also shows following component:   * “Bước kế”: send command completed inputting values.   [Exception: 1, 2, 3, 4] | | 2 | User inputs and sends product’s information to system.  [Alternative 1, 2] | System calls “Search Product Online” function to get a list of products with provided product’s information. System then show this list to user with following information:   * “Tên”: label. * “Hình ảnh”: link. * “Chọn”: user chooses one of listed products. System will default choose first product.   Systems also shows following component:   * “Không phải sản phẩm của bạn”: send command to finish choosing product without product chosen. * “Trở về”: send command to go back to providing-product’s information step.   “Bước kế”: send command to finish choosing product with a product chosen. | | 3 | User chooses a product and inform system.  [Alternative 3, 4, 5, 6] | System get chosen product and get its price as basic price. System gets all suitable stores for product based on category, calculates suggested price of each store then shows it to user with following information:   * “Tên”: label. * “Điện thoại”: label * “Địa chỉ”: label. * “Giá”: label. * “Chọn”: user chooses one of listed stores. System will default choose first store.   When user has already choose a store, system will require user to input:   * “Giá mong muốn”:   + Free text input:   + Format: number   + Required.   Systems also shows following component:   * “Trở về”: Send command to go back to product-choosing step. * “Bước kế”: Send command to finish choosing store.   [Exception 5] | | 4 | User chooses a store and informs system.  [Alternative 7, 8] | System requires user to provide personal information listed below:   * “Tên”:   + Free text input.   + Min-length: 5 characters.   + Max-length: 50 characters.   + Required. * “Địa chỉ”: free text input. * “Điện thoại”:   + Free text input.   + Min-length: 9 characters.   + Max-length: 11 characters.   + Number format.   + Required. * Email:   + Free text input.   + Email format.   + Regular expression: “^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$". * “Cách thoan toán”: user chooses one of 2 options: “Tiền mặt” and “Tài khoản paypal”. * “Mã tài khoản”: free text input. * “Hình ảnh”:   + A large box for product’s image.   + Required. * “Ngày hẹn”:   + “Từ ngày”:     - User picks a date.     - Default date: the next day.     - Required.   + “Đến ngày”     - User picks a date.     - Default date: the next 4 day.     - Required. * “Cách giao hàng": user chooses one of two options: “Cửa hàng đến nhận” or “Tự mang đến”.   Systems also show following component:   * “Chọn ảnh”: choose an image in user pc (when there is no chosen image). * “Trở về”: send command to go back to store-choosing step. * “Gửi yêu cầu”: send command to complete inputting personal information.   For customer, system will use information of customer’s profile to fill some fields automatically.  For guest, the first time guest consigns, system will save information and save it for the next time guest consign. But it will only last a short time.  [Exception 6, 7, 8, 9, 10, 11,12] | | 5 | User inputs and sends personal information to system.  [Alternative 9, 10, 11, 12, 13, 14] | System completes consigning request with all the information user provides. A successful message will be shown with information about consigned product. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User chooses “Tên sản phẩm” option. | “Tên sản phẩm”, “Hãng” will be shown and “Mã sản phẩm” will be hided. System will use “Tên sản phẩm” and “Hãng” fields to check similar products online. | | 2 | User types in “Mã sản phẩm” input in providing-product’s information step. | “Mã sản phẩm” will be shown and “Tên sản phẩm” with “Hãng” will be hided. System will use “Mã sản phẩm” field to check exact products online. | | 3 | User sends request to go back to providing-product’s information step in product-choosing step. | Systems shows providing product’s info step. | | 4 | System find exact products online with provided information when searching product by “Mã sản phẩm” in product-choosing step. | System will show user store-choosing step with suggested prices of each store. | | 5 | System cannot find any products online with provided information when searching product by “Tên sản phẩm” in product-choosing step. | System will show user store-choosing step without suggested price. Instead of that in “Giá” information will show “Thương lượng” text. | | 6 | User clicks “Không phải sản phẩm của bạn” | System will show user store-choosing step without suggested price. Instead of that in “Giá” information will show “Thương lượng” text. | | 7 | User sends request to go back to product-choosing step in store-choosing step when system can find list of products with provided product’s information. | System will show user product-choosing step. | | 8 | User sends request to go back to product-choosing step in store-choosing step when system cannot find any product online with provided product’s information. | System will show user providing product’s information step. | | 9 | User have already chosen a product in product-choosing step. | “Hình ảnh” will show an image of chosen product automatically. | | 10 | User chooses “Tiền mặt” option in providing personal info step. | “Mã tài khoản” free text input will be hidden. | | 11 | User chooses “Tài khoản paypal” option in providing personal info step. | “Mã tài khoản” free text input will be shown and be required. | | 12 | User selects image in providing product’s info step. | System requires user to browse image in user’s computer and upload on page. | | 13 | User changes image in providing product’s info step. | System requires user to browse image in user’s computer and upload on page. | | 14 | User sends request to go back to store-choosing step. | Systems shows store-choosing step. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | “Tên sản phẩm” text input and “Mã sản phẩm” text input are empty | System shows error message: “Xin nhập tên hoặc mã sản phẩm”. | | 2 | Inputted value of “Tên sản phẩm” field is less than 5 characters or longer than 100 characters. | System shows error message: “Yêu cầu 5 đến 100 ký tự”. | | 3 | “Loại” is not selected. | System shows error message: “Xin chọn loại sản phẩm”. | | 4 | Cannot find products with provided “Mã sản phẩm”. | System shows error message: “Không thể tìm thấy sản phẩm với mã số này”. | | 5 | “Giá mong muốn” is empty or not number format | System shows error message: “Xin nhập đúng giá”. | | 5 | Inputted value of “Full name” field is less than 5 characters or longer than 50 characters. | System shows error message: “Tên phải từ 5 đến 50 ký tự”. | | 6 | “Từ ngày” date is not selected. | System shows error message: “Xin chọn ngày hẹn đầu”. | | 7 | “Đến ngày” date is not selected. | System shows error message: “Xin chọn ngày hẹn cuối”. | | 8 | “Từ ngày” date is bigger than “Đến ngày” date. | System shows error message: “Từ ngày phải nhỏ hơn đến ngày”. | | 9 | Inputted value of “Điện thoại” field is empty or invalid. | System shows error message: “Xin nhập đúng số điện thoại”. | | 10 | Inputted value of “Email” field is not empty and invalid. | System shows error message: “Xin nhập đúng email”. | | 11 | Inputted value of “Mã tài khoản” field is required and is empty or invalid. | System shows error message: “Xin nhập đúng mã tài khoản”. |   **Relationships**:   * Including use-case: Include Search Product Online.   **Business Rules**:   * When request is sent successfully, consignment’s status is “waiting” and product’s status is “not available”. This request will be checked by store owner as new request. System will send message and email to user to confirm request has been sent successfully. * If user provides correct UPC, system will find correct product. * If user provides name not UPC in providing-product’s information step, system fill find online similar product with information of “Tên sản phẩm”, “Loại” and “Hãng” (not necessary) then return a list of product with similar information and show it to user. System also shows a message: “Nếu không tìm thấy sản phẩm của bạn, có thể bạn đã nhập sai tên. Vui lòng nhấn quay lại để sửa thông tin”. * If list of products is too much (greater than 10 items) or empty (no item), system will skip product-choosing step and require user choose a store in next step with no suggested price. Store owner will come, check and negotiate with user later when user’s request is accepted. * In store-choosing step, system finds stores that sell this kind of product, based on their category. Then system uses price of chosen product in product-choosing step as basic price. System will calculate suggestion price for each store based on basic price and formula of each store. System also shows a message: “Giá cửa hàng đề nghị có thể không chính xác. Cửa hàng sẽ kiểm tra và định giá chính xác khi đến nhận hàng”. * Formula for suggested price: * If there is no product found online, there will be no suggest price. Instead of that, In “Giá” column will show “Thương lượng” and a message: “Chúng tôi không thể tìm thấy sản phẩm của bạn. Cửa hàng sẽ lưu lại yêu cầu ký gửi này và định giá sản phẩm của bạn khi đến nhận hàng” will be shown. * After sending consignment request, user will receive a code to track product’s status and also consignment’s status. User can also use it to cancel consignment or extend it. * Store owner will contact and receive product on one of the day user chose. But store owner has the right to refuse request if it’s not what store owner need, and also refuse accepting product when checking it if product doesn’t resemble the information user provided. * For customer, store owner will use information in consignment, not customer’s information. So that, consignor is no need to be customer. But this deal will be saved in customer’s history. | | | |

#### <Guest> Order Product

**Use Case Diagram**

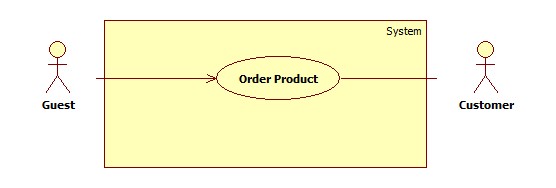


Figure : <Guest> Order Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS005 | | | |
| **Use Case No.** | HPS005 | Use Case Version | 2.0 |
| **Use Case Name** | Order Product | | |
| **Author** | Nguyen Huy Hoang | | |
| **Date** | May 22, 2015 | Priority | High |
| **Actor**:   * Guest * Customer   **Summary**:   * This use case allows user order product.   **Goal**:   * Product will be ordered by user.   **Triggers**:   * User send “Order Product” command.   **Preconditions**:   * User is in product detail page.   **Post Conditions**:   * Success: Show success message, create an order of this product with user information * Fail: Show an error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send “Order Product” command.  [Alternative 1] | System will navigate to enter user’s information include:   * “Tên”: Free text input, max length: 50. * “Email”: Free text input, regular expression: “^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$". * “Địa Chỉ”: Free text input. * “Số Điện Thoại”: Free text input, min length: 9, max length: 11. * A command to finish action. | | 2 | User input information. |  | | 3 | User send “Confirm Information” command.  [Alternative 2] | System create an order of this product with user information.  [Exception 1,2] | |  |  |  |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If product have been ordered | System will show message: “Món hàng này đã có A người đặt, có thể bạn sẽ không mua được món hàng, vui lòng liên hệ với chủ cửa hàng để biêt thêm." With A is number of orders. | | 2 | If product is not special product. | System will send sms to notify price to user. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User not fills all required textbox | Show message: “Vui lòng nhập thông tin vào nhưng ô bắt buộc.” | | 2 | User enter phone number is existed in another order of this product. | Show message: “Số điện thoại đã sử dụng để đặt món hàng này”. |   **Relationships**:   * Extending Use-cases: Extend from Product Detail (Search function has the result and user send “View Product Detail” command).   **Business Rules**:   * All require fill must be filled. * User can’t use same phone number to order same product. * If user login into system, user’s information is autofill. * On success scenario, system will send SMS to customer to notify success order. * Product status is change to “ordered”. | | | |

#### <Store Owner> Accept Request

**Use Case Diagram**

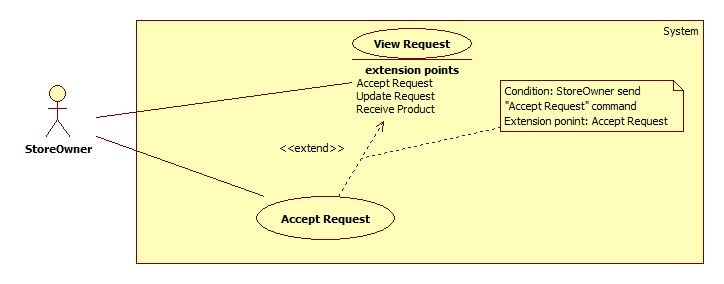


Figure : <Store Owner> Accept Request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS007** | | | |
| **Use Case No.** | HPS007 | Use Case Version | 2.0 |
| **Use Case Name** | Accept Request | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Store Owner accepts a new request.   **Goal**:   * Store Owner successfully accepts a request.   **Triggers**:   * Store Owner send command to accept a request.   **Preconditions**:   * Store Owner has logged onto the system. * Store Owner is viewing details of a new request.   **Post Conditions**:   * **Success**: Successfully accept a request. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | While viewing details of a new request, Store Owner sends command to accept a request with appointment date.  [Alternative 1] | System saves appointment date and changes this request to “Chấp nhận yêu cầu”. |   **Alternative**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to refuse request | System requires a reason from Store Owner then changes this request to “Từ chối yêu cầu”. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to accept request when request has already handled. | System shows message: “Yêu cầu này đã được xử lý”. |   **Relationships**:   * Extended Use-case: View Request.   **Business Rules**:   * After accepting request, this request’s status will change to “Chấp nhận yêu cầu”. Consignment’s status is “Accepted” and product’s status is “Not available”. * If store Owner cancels request, this request’s status will change to “Từ chối yêu cầu”. Consignment’s status is “Refused” and product’s status is “Not available”. * System will send message and email to consignor to confirm action on request. * If accepted request has the value of “Cách giao hàng” is “Cửa hang đến nhận” and “Ngày giao hàng” is current day, system will notify Store Owner in mobile application about this request immediately. | | | |

#### <Store Owner> Receive Product

**Use Case Diagram**

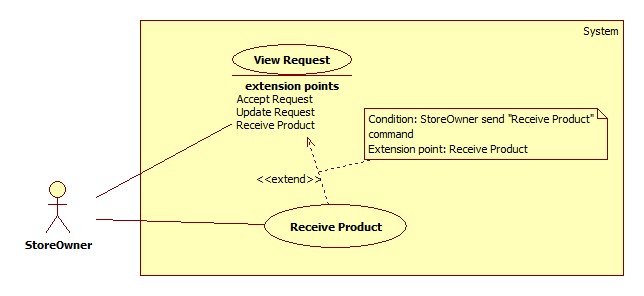


Figure : <Store Owner> Receive Product

**Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS009** | | | |
| **Use Case No.** | HPS009 | Use Case Version | 2.0 |
| **Use Case Name** | Receive Product | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Store Owner can receive product.   **Goal**:   * Store Owner successfully accept product.   **Triggers**:   * Store Owner send command to accept product.   **Preconditions**:   * Store Owner has logged onto the system. * Store Owner is viewing details of an accepted request.   **Post Conditions**:   * **Success**: Successfully accept product. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | While viewing details of an accepted request, Store Owner sends command to accept product.  [Alternative 1] | System saves provided information and changes this request to “Đã nhận hàng”.  [Exception 1, 2] |   **Alternative**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to refuse product. | System requires a reason from Store Owner then changes this request to “Từ chối sản phẩm”. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner does not provide valid information. | System shows message: “Xin nhập đúng thông tin”. | | 2 | Store Owner sends command to accept product when this request is already handled. | System shows message: “Yêu cầu này đã được xử lý”. |   **Relationships**:   * Extended Use-case: View Request.   **Business Rules**:   * When product is received, its status is displayed as “Đã nhận hàng” and this product will disappear in the request list and appear in product management. Consignment’s status now is “Received” and product’s is “Available”. * If product is refused, its sstatus will be displayed as “Từ chối sản phẩm”. Consignment’s status now is “Refuse” and product’s is “Not available”. * After accepting product, Store Owner will store the product and set the day to go on web for shopping. * Mobile application have the same function as this, to receive product and bring it back to store. * System will send message and email to consignor to confirm action on product. | | | |

#### <Store Owner> Publish On Web

**Use Case diagram**

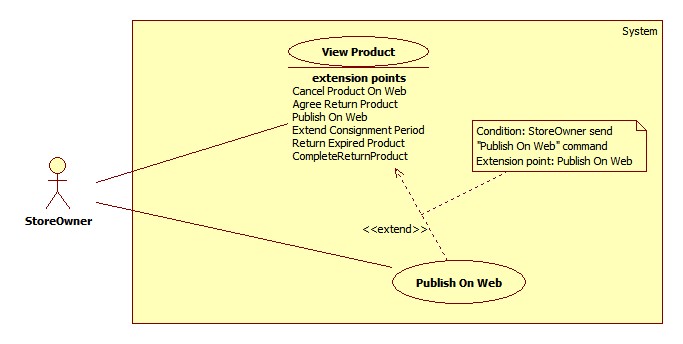


Figure : <Store Owner> Publish On Web

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS011** | | | |
| **Use Case No.** | **HPS011** | Use Case Version | 2.0 |
| **Use Case Name** | Publish On Web | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to publish product on web.   **Goal**:   * Successfully publish product so it can be found by search function.   **Triggers**:   * When store owner sends publish product on web command.   **Preconditions**:   * Guest is logged in as Store Owner. * Product status must be “Available”.   **Post Conditions**:   * **Success**: Product status is changed to “On Web” and can be found by search function. * **Fail**: Show error, no information has been changed.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send view available product detail command. | System load product detail information into pop-up with following information:   * Tên sản phẩm:   + Free text input.   + Min-length: 5 characters.   + Max-length: 50 characters.   + Required. * Số seri:   + Free text input.   + Min-length: 5 characters.   + Max-length: 50 characters.   + Required. * Độ mới:   + Free text input.   + Number (0-100). * Loại sản phẩm:   + Single choice option. * Hãng:   + Min-length: 5 characters.   + Max-length: 50 characters.   + Required. * Mùa:   + Multi-choice option. * Tự động gửi giá:   + Single choice option. * Mô tả:   + Free text area. * Ảnh sản phẩm:   + Image. | | 2 | Store owner enters or modifies product detail information.  Store owner sends publish product on web command.  [Alternative 1] | System validates inputted data. [Exception 1,2,3]  System update product detail information in database.  System notifies store owner after complete.  System redirects to view available product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner dismiss popup. | System closes popop, redirect to view product page.  No information is recorded. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | “Tên sản phẩm” is empty. | System show message “Vui lòng nhập tên sản phẩm”. | | 2 | “Hãng sản phẩm” is empty. | System show message “Vui lòng nhập hãng sản phẩm”. | | 3 | “Độ mới” is not a number or less than 0 or greater than 100. | System show message “Vui lòng nhập đúng độ mới của sản phẩm”. |   **Relationships**:   * Extended Use-case: View Product.   **Business Rules**:   * Only product in “Available” status can be published on web. * System will change product status on database base on product id, product status is changed to “On Web”. * If store owner chooses seasons for product, it will be display on home page during those season. If store owner doesn’t choose any season, it will display in “Sản phẩm nổi bật” category. * If store owner chooses “Tự động gửi giá” is “Có”, when product is ordered, system will automatic send product price to customer. * If store owner doesn’t choose new image for product, product image will remain as the image at beginning. * Product name, product brand and new status can not be empty. | | | |

#### <Store Owner> Pay Consignor

**Use Case diagram**

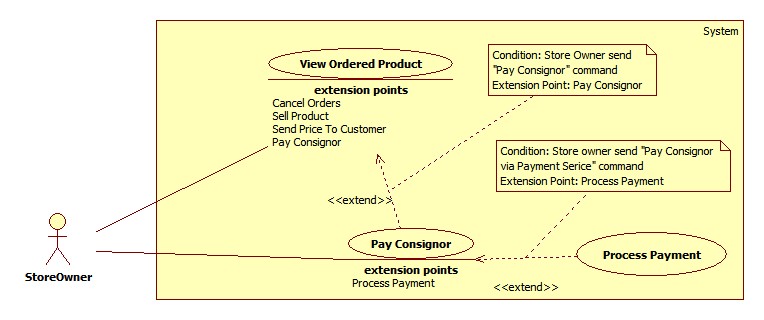


Figure : <Store Owner> Pay Consignor

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS018** | | | |
| **Use Case No.** | **HPS018** | Use Case Version | 2.0 |
| **Use Case Name** | Pay Consignor | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to pay consignor after sell their product.   **Goal**:   * Complete consignment and return money to consignor.   **Triggers**:   * When store owner send pay consignor command.   **Preconditions**:   * Guest is logged in as Store owner and is on view sold product page.   **Post Conditions**:   * **Success**: Change product status to “Not Available” and consignment status to “Completed”. * **Fail**: Show error message. No information has been changed.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view product detail command. | System loads information about the product into pop-up with following information.   * Họ tên: Label. * Địa chỉ: Label. * Số điện thoại: Label. * Email: Label. * Tài khoản Paypal: Label. * Tên sản phẩm: Label. * Mã kí gửi: Label. * Giá kí gửi: Label. * Giá thực bán: Label. * Ngày kí gửi: Label. * Tiền trả khách hàng: Input field, Editable | | 2 | Store owner enters return price | System validated inputted data. [Exception 1]. | | 3 | Store owner sends process normal payment command.  [Alternative 1]  [Alternative 2] | System update database.  System redirects to view ordered product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner dismiss popup. | System closes popop, redirect to view ordered product page.  No information is recorded. | | 2 | Store owner sends pay consignor via payment service.  Store owner processes payment on payment service page.  Store owner send return to HPS website | System loads information online payment with following information.   * Cmd: text, hidden. * Business: text, hidden. * ReturnURL: text, hidden. * Rm: number, hidden. * Item\_name: text, hidden. * Item\_number: number, hidden. * Currency\_code: text, hiddent. * Amount : number, hidden.   System redirect to payment service page.  System retrieves return value of payment process. [Exception 2]  System updates database.  System redirects to view ordered product page. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | If inputted value is not valid. | System shows notification and requests store owner to re-enter the value. | | 2 | Returned value of paypent service is not valid | System cancel payment, redirect to view ordered product page.  No information has been changed. |   **Relationships**:   * Extended Use-case: View Ordered Product. * Extending Use-case: Process Payment (Store owner sends pay consignor via payment service command).   **Business Rules**:   * The amount of money which return to consignor must be greater than 0. * The amount of money is negotiated by consignor and store owner. * Store owner cannot change receiver’s information. * System will load payment information from database. * Store owner has to provide right information about their online payment account. * Only consignment which has consignor online payment account can performance this use case. * When payment is completed, system will send SMS to consignor to notify them about the payment. | | | |

#### <System> Search Product Online

**Use Case diagram**



Figure : <System> Search Product Online

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS025** | | | |
| **Use Case No.** | **HPS025** | Use Case Version | 2.0 |
| **Use Case Name** | Search Product Online | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * System   **Summary**:   * This use case allows system search product based on information that was given by consignor.   **Goal**:   * System return list of product which match with information that consignor has provided.   **Triggers**:   * When consignor send consign command.   **Preconditions**:   * Guest consigns product and has inputted necessary information.   **Post Conditions**:   * **Success**: Return list of products. * **Fail**: Return nothing.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System gets data which is provided by consignor.  System send search product command. | System sends search product detail to price suggestion services.  [Exception 1]  System gets return value.  System parses return value to list of products.  [Exception 2]  System returns result. |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System returns empty list. | Error occurs during search product detail phase. | | 2 | System returns message “Too many result” | Return result has more than 10 items |   **Relationships**: N/A  **Business Rules**:   * Product name or UPC (Universal Product Code) information must be provided to system. * If return result is more than 10 items, system will return empty list. * If return result is less than 10 items, system will return list of product which contain those items. * If return result is empty, system will return empty list. * Product price is the lowest new price of the product. * Product has following information: Product name, lowest new price of the product, product image. | | | |

## Conceptual Diagram

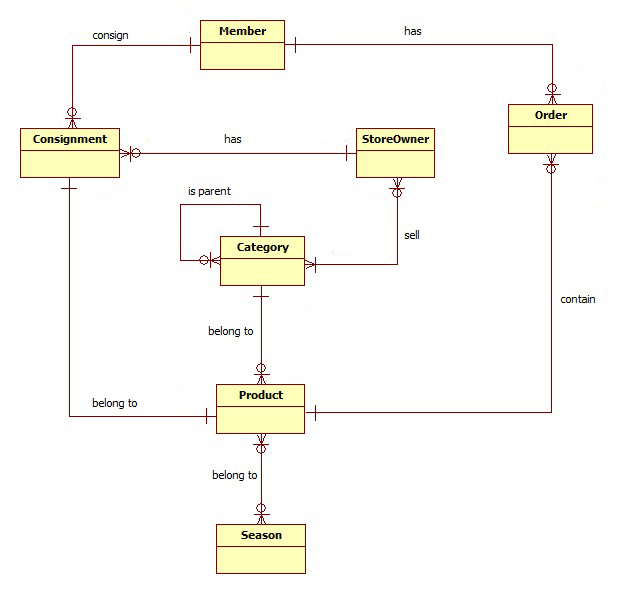


Figure : Conceptual Diagram

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| Consignment | Describe all consignments in the system. |
| Customer | Describe all customers in the system. |
| Store Owner | Describe all store owners in the system. |
| Product | Describe all products in the system. |
| Category | Describe all product categories in the system. |
| Order | Describe all orders in the system. |
| Season | Describe all seasons. |

# Report No. 4 Software Design Description

## System Architectural Design

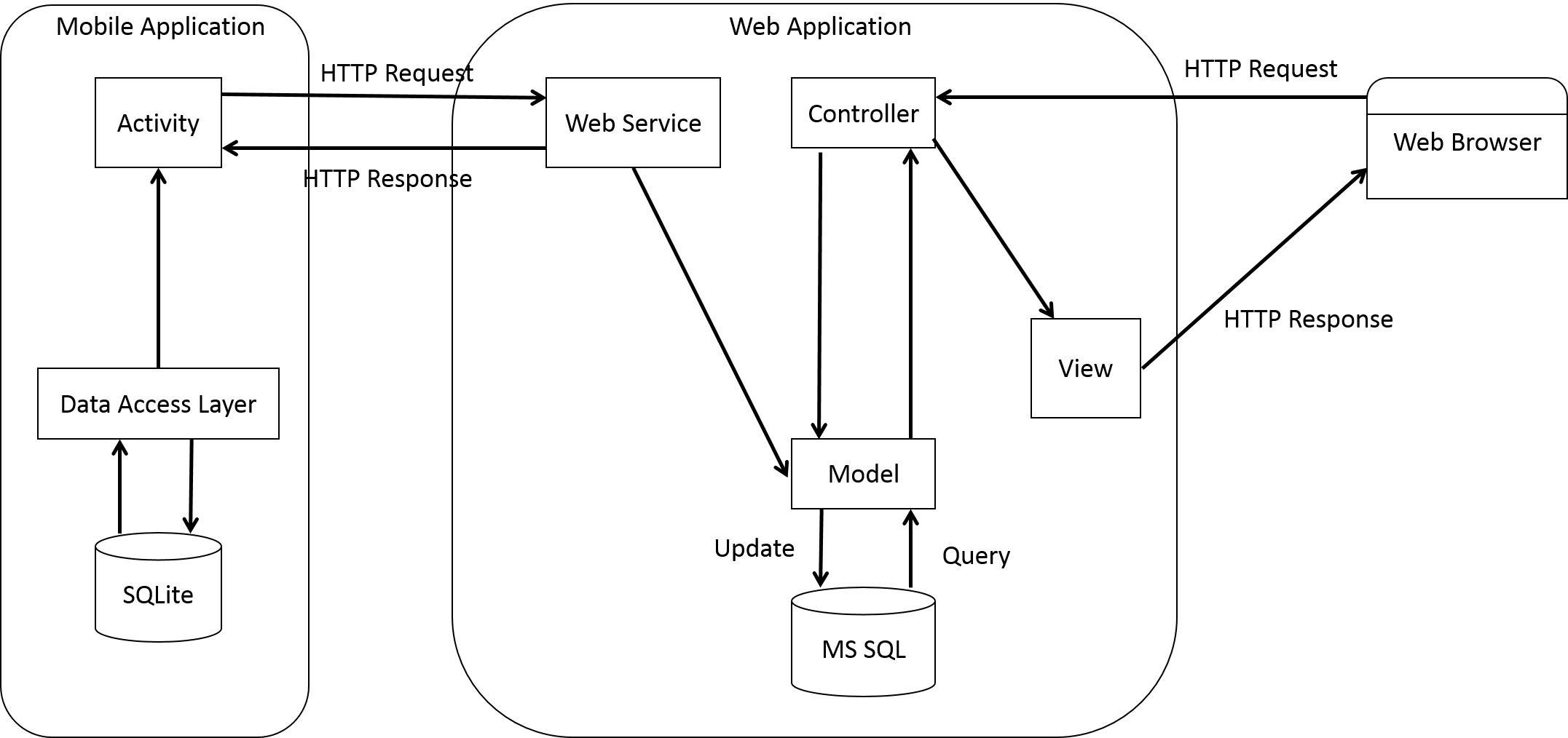


Figure : System Architectural Design

### Web Application Architecture Description

* The system is developed from MVC architecture style and customized with passive model to adapt to JAVA MVC, RESTful web service technologies.
* **Model** is a part of application which uses to storage and retrieval data.
* View is the part of the application that visualizes the data to help user **easily** to view, update or customize them.
* **Controller** is the parts of the application that acts like event handler to handles user interaction. Typically controller read data from a request and returns respective view for user.
* **Web** **Service** is the parts of the application which use to provide interface for mobile application to do some background operation such as get consignment, update consignment status…

### Mobile Application Architecture Description

* The application is developed as an Android native application. In general, the application architecture follow by Android architecture.
* **Activity** is basic core of an Android application that handles user input, call web service…
* **Data** **Transfer** Object is the parts of the application that mapping SQLite database to storage and retrieval the consignment’s orders which assigned for employee.

## Component Diagram

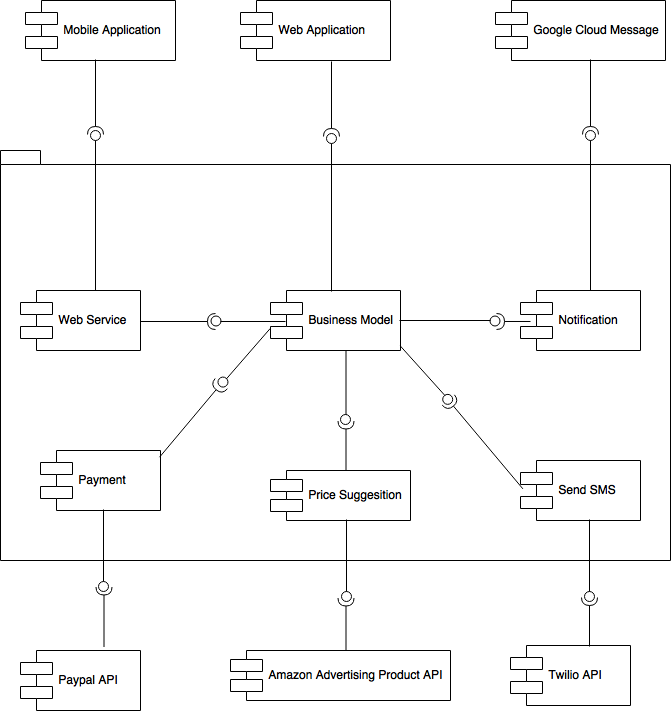


Figure : Component Diagram

|  |  |
| --- | --- |
| **Component dictionary: describe component** | |
| **Component Name** | **Description** |
| Web Service | Content all web service of the system. |
| Business Model | Handle business process. |
| Notification | Use to send notification message to mobile. |
| Google Cloud Message | Use to push notification to mobile application. |
| Mobile Application | Android Application. |
| Web Application | Web Application. |
| Web Service | Use to connect with mobile application. |
| Payment | Use to call Paypal API |
| Paypal API | Use to online payment. |
| Price Suggestion | Use to call Amazon Advertising Product API. |
| Send SMS | Use to call Twilio API. |
| Amazon Advertising Product API | Use to get price suggestion. |
| Twilio API | Use to send SMS. |

## Interaction Diagram

### Sequence Diagram

#### <Guest> Consign Product

**Summary:** This diagram shows how Guest can consign product.

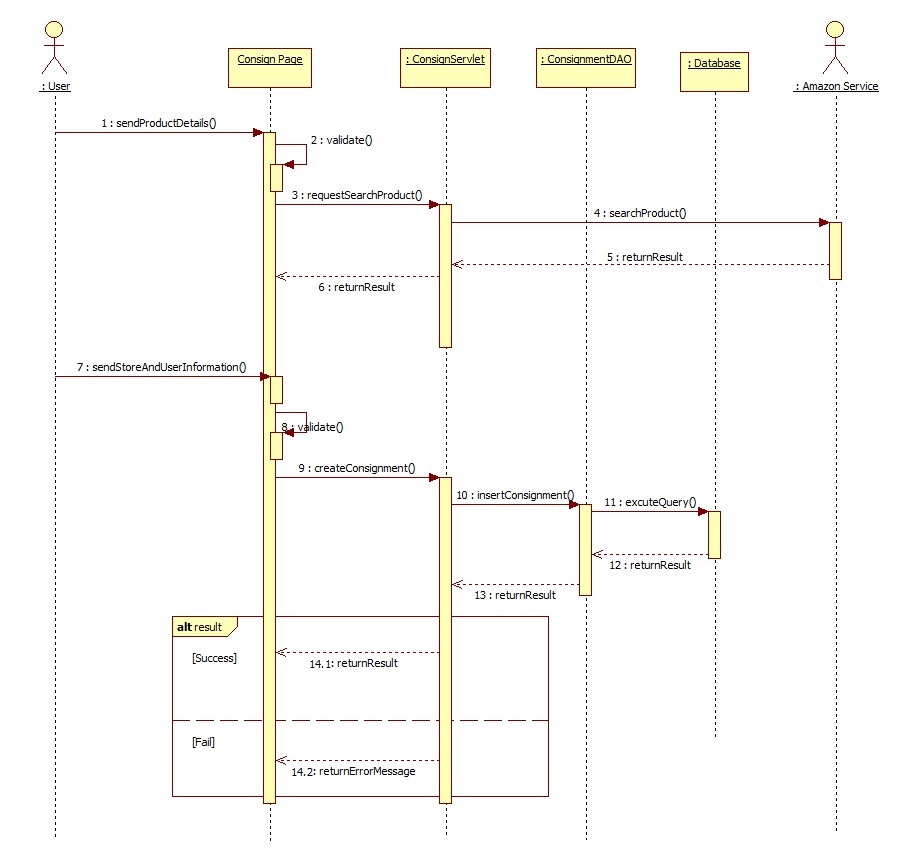


Figure : Consign Product

#### <Guest> Order Product

**Summary:** This diagram shows how Guest can order product

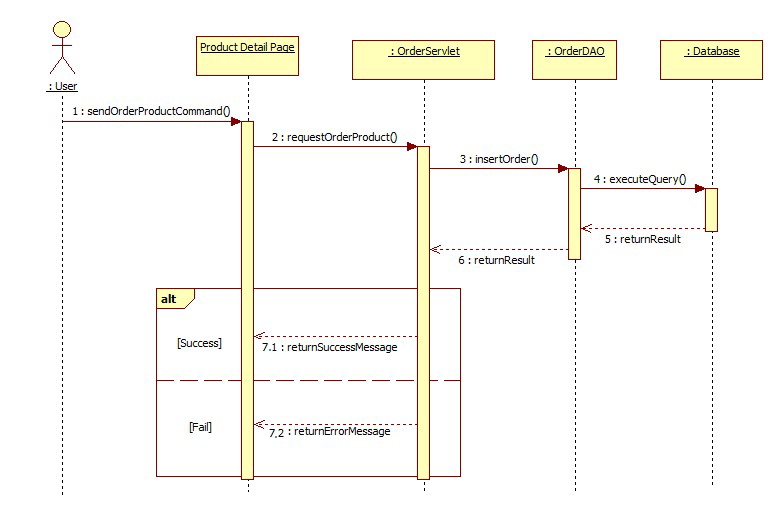
****

Figure : Order Product

#### <Store Owner> Accept Request

**Summary:** This diagram shows how store owner accept a request

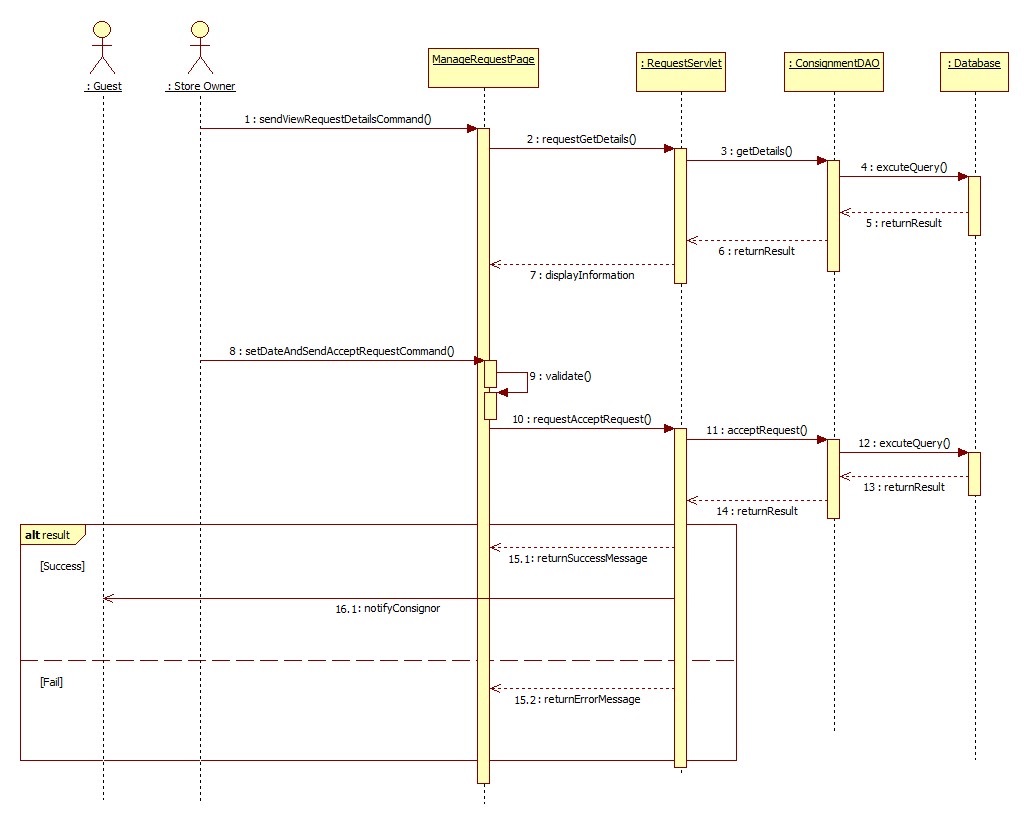


Figure : Accept Request

#### <Store Owner> Receive Product

**Summary**: This diagram shows how store owner accepts and receives product from consignor.

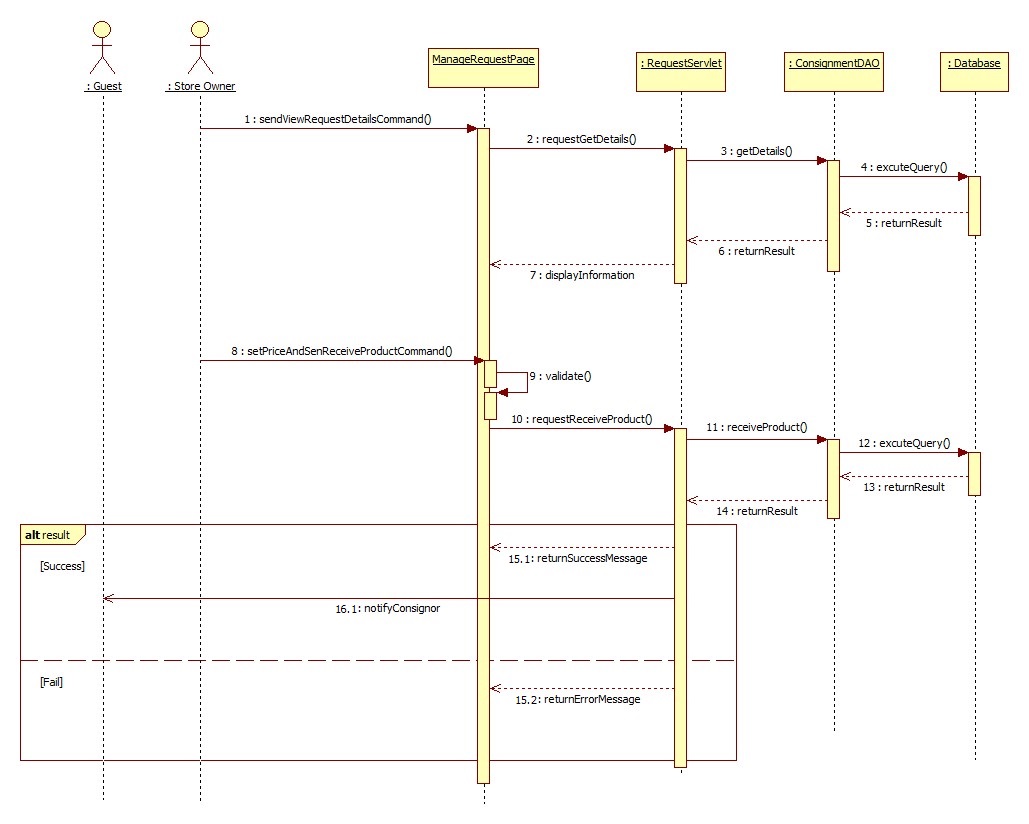


Figure : Receive Product

#### <Store Owner> Publish On Web

**Summary**: This diagram shows how store owner publish a product to website.

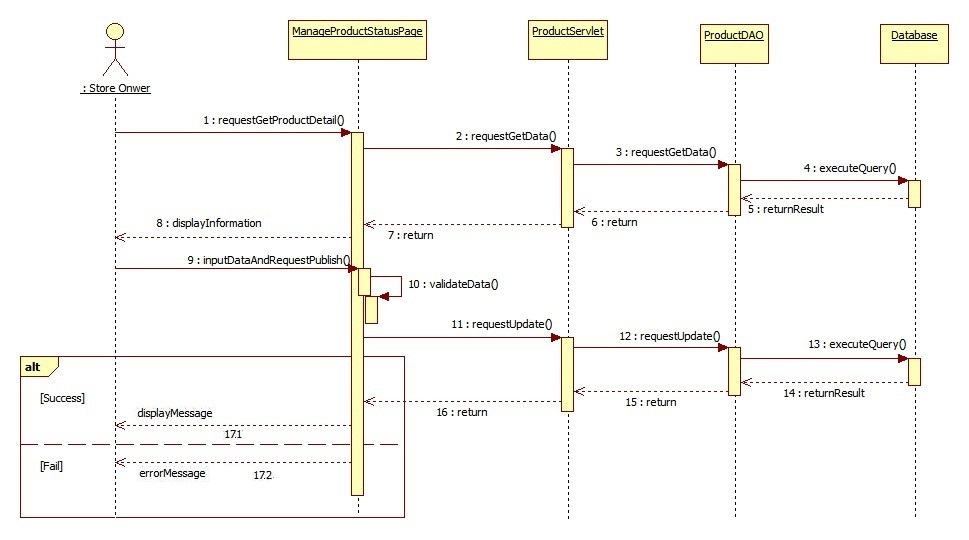


Figure : Publish On Web

#### <Store Owner>Pay Consignor

**Summary**: This diagram shows how store owner pay consignor without online payment method.

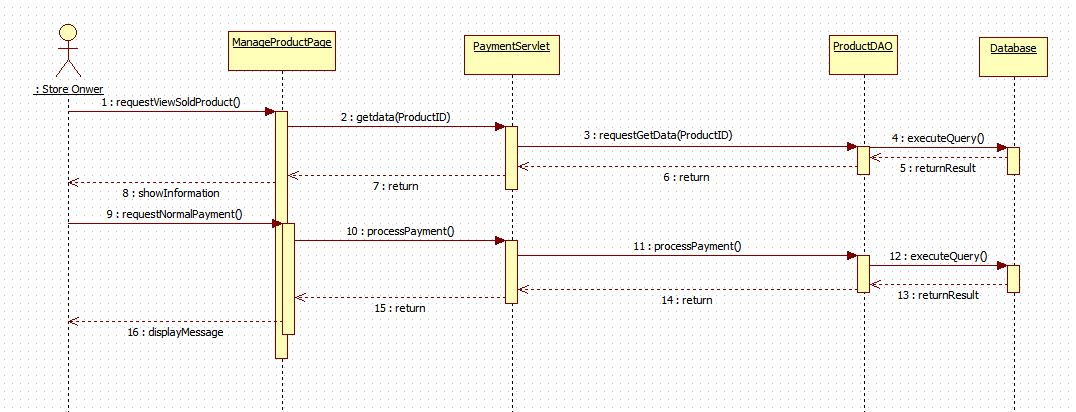


Figure : Pay Consignor

### Activity Diagram

#### <Store Owner> Auto Synchronize

**Summary:** This diagram shows how Mobile Application auto synchronize data.

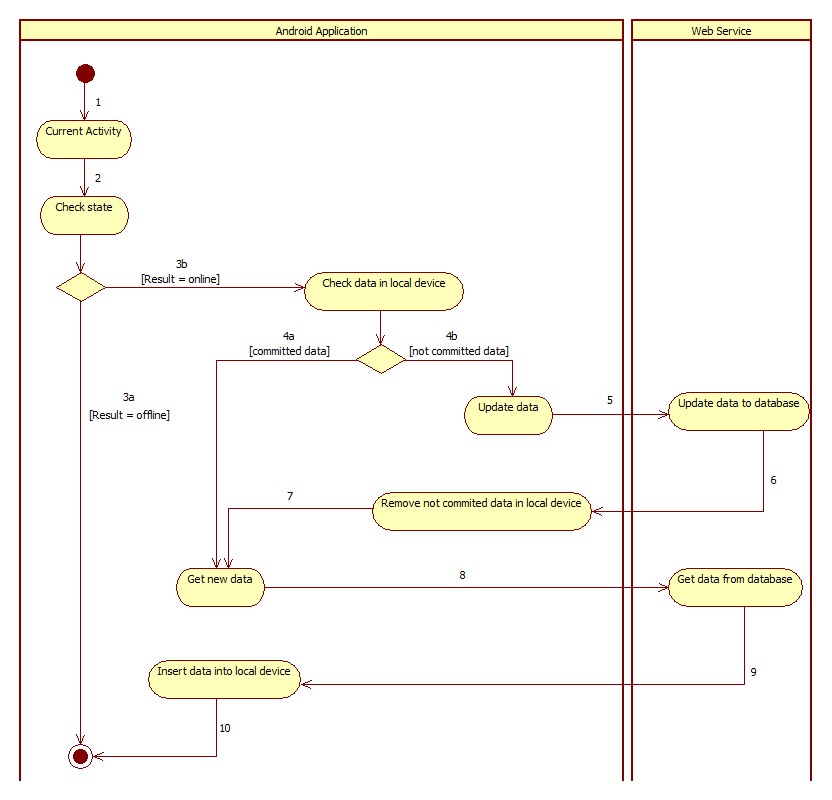
****

Figure : Auto Synchronize

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Call checkState() | 3a. Finish |
| 3b. Check data in local device, if exist data have not committed, update to server | 4a. Call getNewData() | 4b. call updateData() |
| 5. Send data to web service | 4. Remove not committed data | 7. Call getNewData() |
| 8. Get data from web service | 9. Insert data into local device | 10. Finish |

#### <StoreOwner> Accept Product

**Summary:** This diagram shows how StoreOwner receive product.

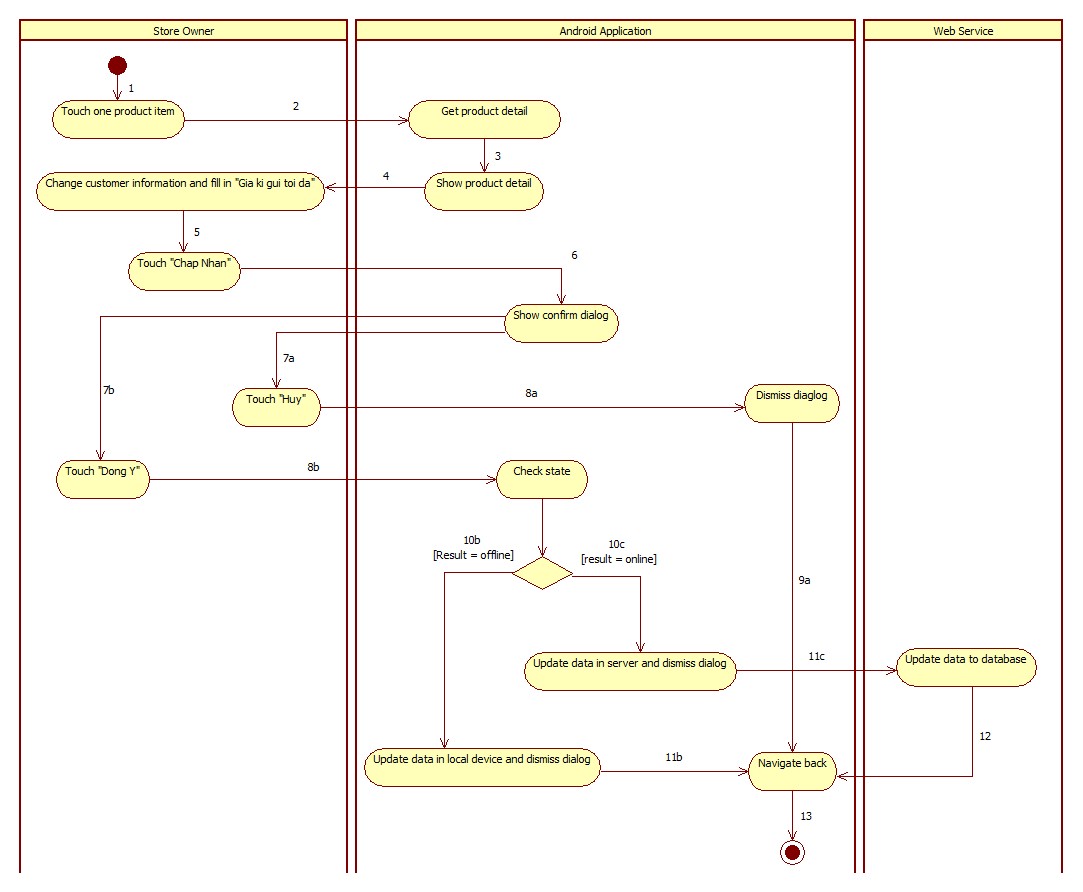


Figure : Accept Product Activity Diagram

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Call getProductDetail() | 3. Show product detail |
| 4. Store Owner change customer information and enter “Giá kí gửi tối đa” | 5. Store Owner touch “Chấp Nhận” button | 6. Show confirm dialog |
| 7a. Store Owner touch “Hủy” button to cancel action | 7b. Store Owner touch “Đồng ý” button | 8a. Dismiss confirm dialog |
| 8b. Call checkState() | 9a. Navigate back | 10b. Update data in local device and dismiss dialog |
| 10c. Call service to update data in server and dismiss dialog | 11b. Navigate back | 11c. Send data to web service |
| 12. Navigate back | 13. Finish |  |

## Algorithms

### Price Suggestion

#### Define Problem

When user consign product, they enter product name or product serial. We base on this information to get suggestion price for them.

#### Solution

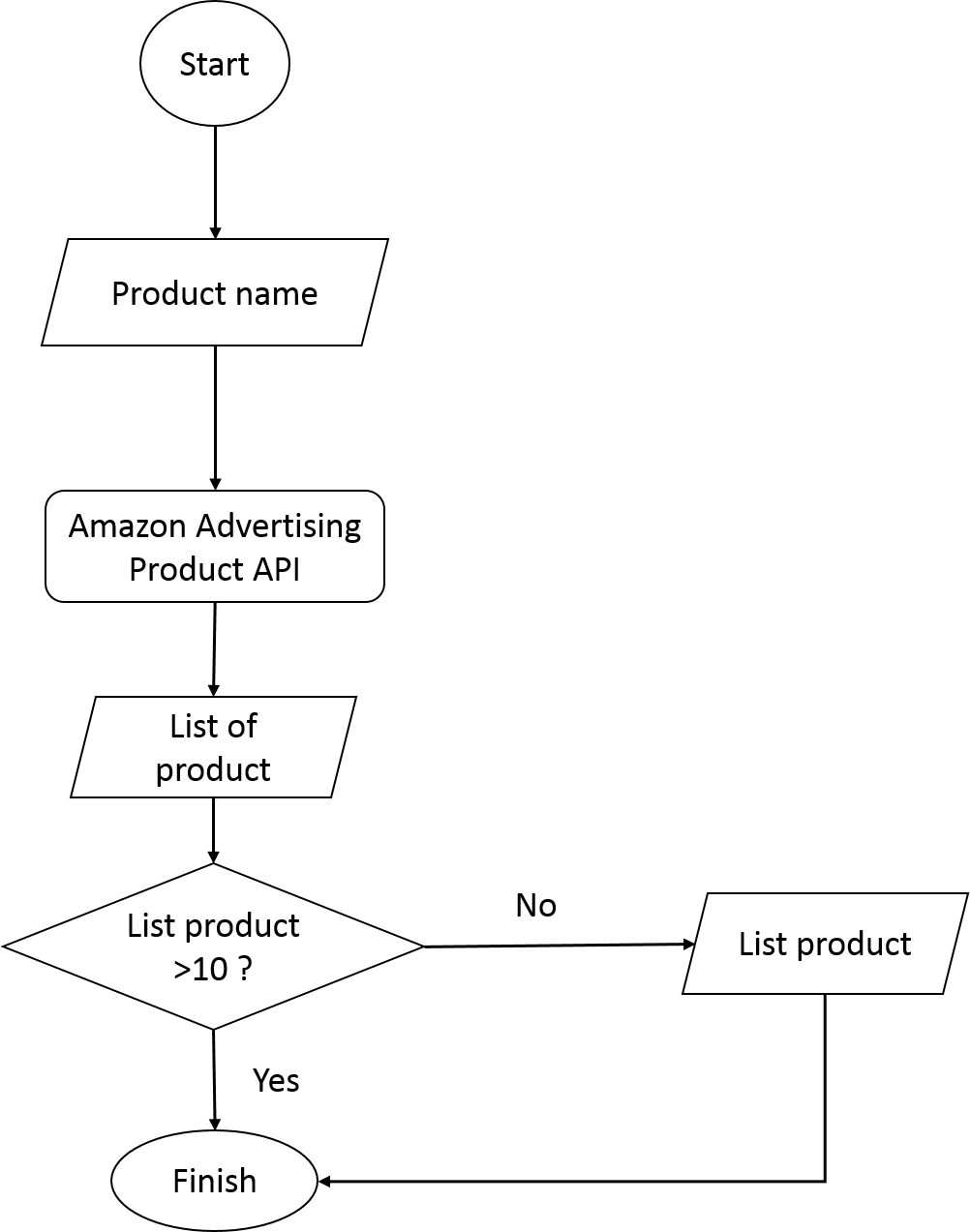
To archive the goal above, we use **Amazon Product Avertising API** to get product infomation base on product name or product serial.

We can get product information from amazon following:

* If search product by product serial, **Amazon Product Avertising API** will return exactly 1 product if product serial is exist. We will using this product information.
* If search product by product name, **Amazon Product Avertising API** will return list product correctpond with product name. If list product is less than 10 products, we will get this list and show to user choose 1 of them.

#### Flowchart

##### Search product by product name



##### Search product by product serial



### Consignment State

The consignment’s states describes as below:

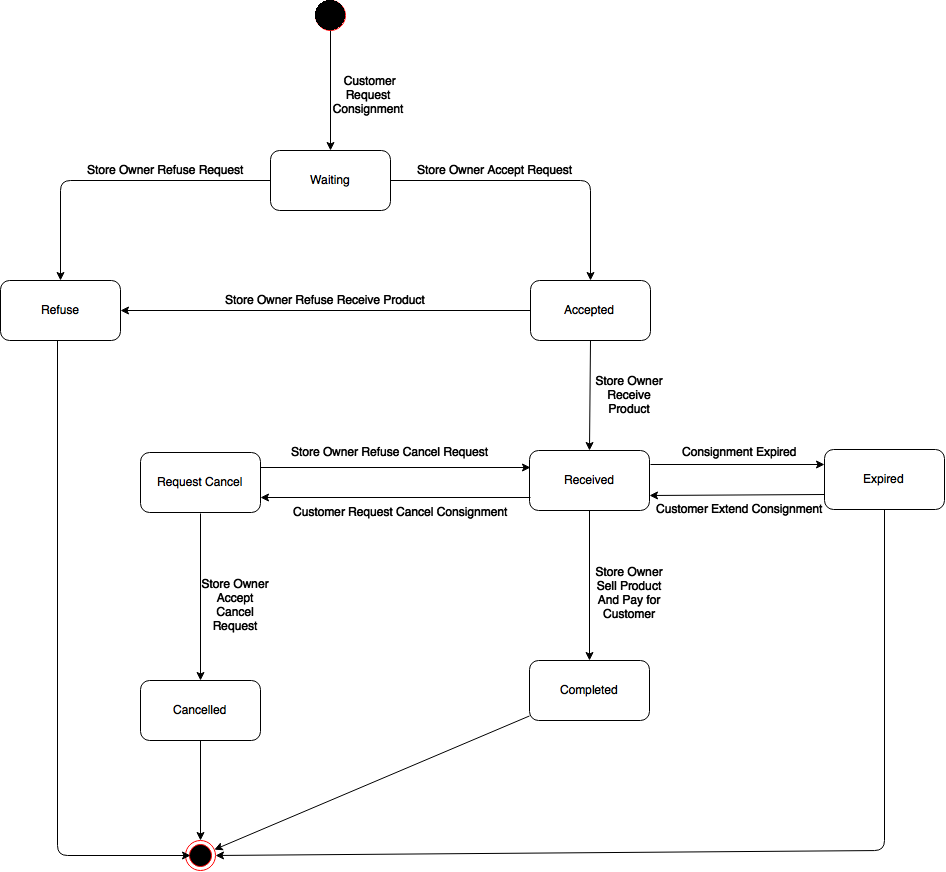


Figure : Consignment State

|  |  |
| --- | --- |
| **State** | **Description** |
| Waiting | Customer requests new consignment |
| Refused | Storeowner refuses consignment’s request |
| Accepted | Storeowner accepts consignment’s request |
| Received | Storeowner receives product |
| Completed | Storeowner pays for customer |
| Request Cancel | Customer requests cancel consignment |
| Cancelled | Storeowner accepts request cancel |
| Expired | Consignment is overdue |

|  |  |  |  |
| --- | --- | --- | --- |
| **Stimulus** | **Description** | **State before** | **State after** |
| Create request | Customer creates new consignment | N/A | Waiting |
| Accept request | Store owner accept consignment request | Waiting | Accepted |
| Refuse request | Store owner refuse consignment request | Waiting | Refused |
| Receive product | Store owner receive product | Accepted | Received |
| Request cancel | Customer request cancel their consignment | Received | Request Cancel |
| Agree cancel | Store owner agree customer cancel request | Request Cancel | Cancelled |
| Disagree cancel | Store owner disagree customer cancel request | Request Cancel | Received |
| Expire | Customer consignment is overdue | Received | Expired |
| Extend | Customer extend their consignment period | Expired | Received |
| Pay | Store owner pay for customer consignment | Received | Completed |

# Report No. 5 System Implement & Test Description

## Database Relationship Diagram

### Physical Diagram

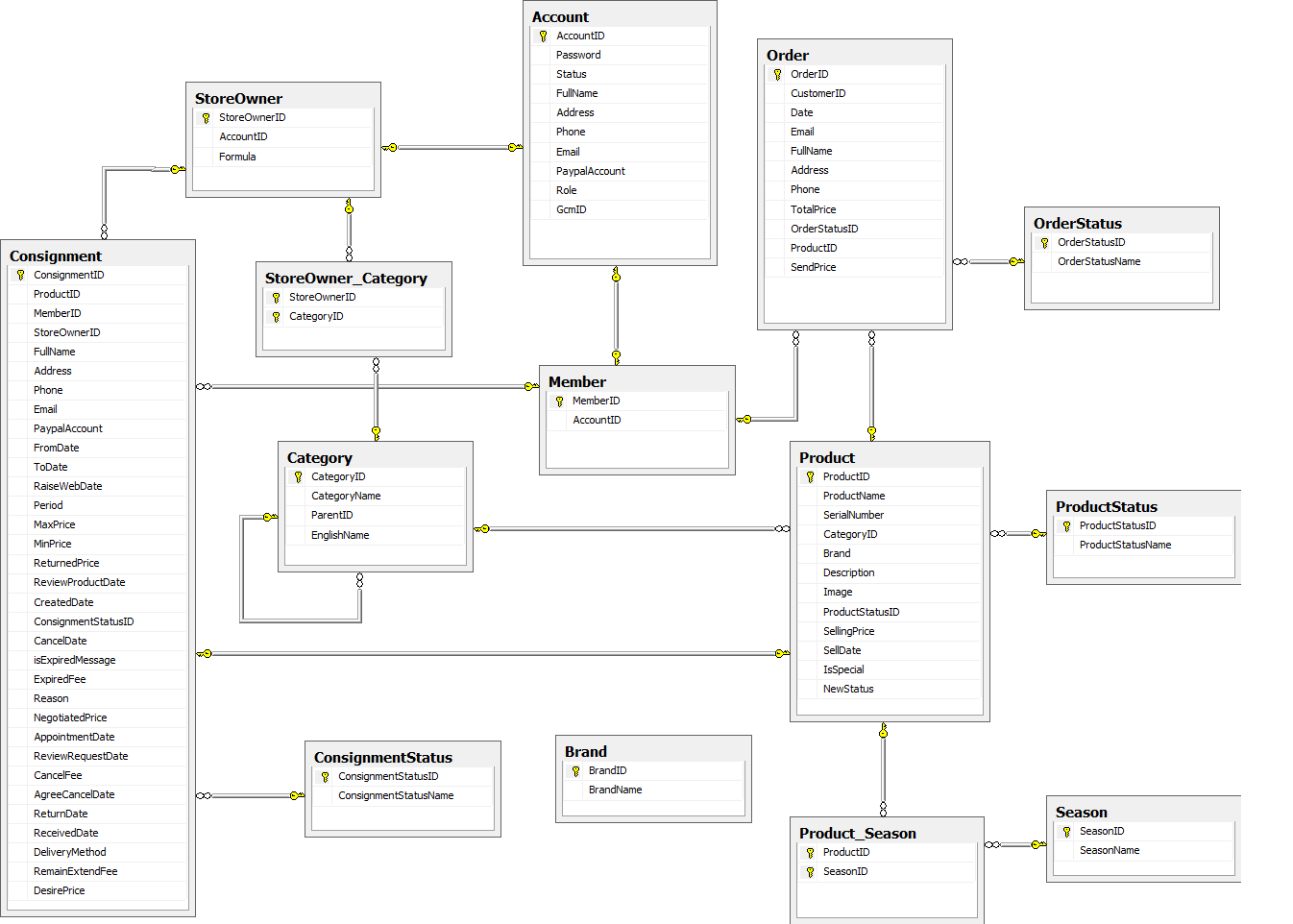


Figure : Physical Database