|  |  |
| --- | --- |
|  | **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*

Table of Contents

[A. Report No.1 Introduction 11](#_Toc427058905)

[1. Project Information 11](#_Toc427058906)

[2. Introduction 11](#_Toc427058907)

[3. Current Situation 11](#_Toc427058908)

[4. Problem Definition 11](#_Toc427058909)

[5. Proposed Solution 12](#_Toc427058910)

[5.1 Feature functions 12](#_Toc427058911)

[5.2 Advantages and disadvantages 12](#_Toc427058912)

[6. Functional Requirements 13](#_Toc427058913)

[7. Role and Responsibility 13](#_Toc427058914)

[B. Report No.2 Software Project Management Plan 14](#_Toc427058915)

[1. Problem Definition 14](#_Toc427058916)

[1.1 Name of this Capstone Project 14](#_Toc427058917)

[1.2 Problem abstract 14](#_Toc427058918)

[1.3 Project Overview 14](#_Toc427058919)

[2. Project Organization 16](#_Toc427058920)

[2.1 Software Process Model 16](#_Toc427058921)

[2.2 Roles and responsibilities 19](#_Toc427058922)

[2.3 Tools and Techniques 19](#_Toc427058923)

[3. Project Management Plan 20](#_Toc427058924)

[3.1 Software development life cycle 21](#_Toc427058925)

[3.2 Phase Detail 24](#_Toc427058926)

[3.3 All Meeting Minute 25](#_Toc427058927)

[4. Coding Convention 25](#_Toc427058928)

[C. Report No.3 Software Requirement Specification 27](#_Toc427058929)

[1. User Requirement Specification 27](#_Toc427058930)

[1.1 Guest Requirement 27](#_Toc427058931)

[1.2 Customer Requirement 27](#_Toc427058932)

[1.3 Store Owner Requirement 27](#_Toc427058933)

[1.4 Admin Requirement 27](#_Toc427058934)

[2. User Requirement Specification 28](#_Toc427058935)

[2.1 External Interface Requirement 28](#_Toc427058936)

[2.2 System Overview Use Case 28](#_Toc427058937)

[2.3 List of Use Case 29](#_Toc427058938)

[3. Software System Attribute 112](#_Toc427058939)

[3.1 Usability 112](#_Toc427058940)

[3.2 Reliability 112](#_Toc427058941)

[3.3 Availability 112](#_Toc427058942)

[3.4 Security 112](#_Toc427058943)

[3.5 Maintainability 112](#_Toc427058944)

[3.6 Portability 112](#_Toc427058945)

[3.7 Performance 112](#_Toc427058946)

[2 Conceptual Diagram 113](#_Toc427058947)

[4.1 Conceptual Diagram 113](#_Toc427058948)

[D. Report No. 4 Software Design Description 114](#_Toc427058949)

[1. Design Overview 114](#_Toc427058950)

[2. System Architectural Design 114](#_Toc427058951)

[4.1 Web Application Architecture Description 114](#_Toc427058952)

[4.2 Mobile Application Architecture Description 115](#_Toc427058953)

[3. Component Diagram 115](#_Toc427058954)

[4. Detailed Description 117](#_Toc427058955)

[4.1 Class Diagram 117](#_Toc427058956)

[4.2 Class Diagram Explanation 117](#_Toc427058957)

[4.3 Interaction Diagram 123](#_Toc427058958)

[5. User Interface Design 142](#_Toc427058959)

[5.1 Guest/Customer Interface Design 142](#_Toc427058960)

[5.2 StoreOwner Interface Design 142](#_Toc427058961)

[5.3 Admin Interface Design 142](#_Toc427058962)

[6. Database Design 143](#_Toc427058963)

[6.1 Logical Diagram 143](#_Toc427058964)

[6.2 Data Dictionary 143](#_Toc427058965)

[7. Algorithms 143](#_Toc427058966)

[E. Report No. 5 System Implement & Test Description 143](#_Toc427058967)

[1. Introduction 143](#_Toc427058968)

[1.1 Overview 143](#_Toc427058969)

[1.2 Test Approach 143](#_Toc427058970)

[2. Database Relationship Diagram 143](#_Toc427058971)

[2.1 Physical Diagram 143](#_Toc427058972)

[2.2 Data Dictionary 143](#_Toc427058973)

[3. Performance Measurement 144](#_Toc427058974)

[4. Test Plan 144](#_Toc427058975)

[4.1 Features to be tested 144](#_Toc427058976)

[4.2 Features not to be tested 144](#_Toc427058977)

[5. System Testing Test Case 144](#_Toc427058978)

[5.1 StoreOwner Core Flow 144](#_Toc427058979)

[5.2 Guest Core Flow 144](#_Toc427058980)

**List of Tables**

[Table 1: Roles And Responsibilities 13](#_Toc427058981)

[Table 2: Hardware Requirement For Mobile 15](#_Toc427058982)

[Table 3: Hardware Requirement For Server 15](#_Toc427058983)

[Table 4: Software Process Guideline Question Decision 18](#_Toc427058984)

[Table 5: Project Roles And Responsibilities 19](#_Toc427058985)

[Table 6: Software Development Lifecycle Detail 23](#_Toc427058986)

[Table 7: Phase 1: Requirement Analysis 24](#_Toc427058987)

[Table 8: Phase 2: System and Software Design 24](#_Toc427058988)

[Table 9: Phase 3: Implementation and Unit Testing 25](#_Toc427058989)

[Table 10: Phase 4: Integration and System Testing 25](#_Toc427058990)

[Table 11: Phase 5: Operation and Maintenance 25](#_Toc427058991)

**List of figures**

[Figure 1: The Waterfall Model (Sommerville, Software Engineering, 2011) 18](#_Toc427058992)

[Figure 2: System Overview Use Case 29](#_Toc427058993)

[Figure 3: <Guest> Overview Use Case 30](#_Toc427058994)

[Figure 4: <Guest> Consign Product Use Case 30](#_Toc427058995)

[Figure 5: <Guest> Track product status Use Case 39](#_Toc427058996)

[Figure 6: <Guest> Cancel Consigned Product Use Case 41](#_Toc427058997)

[Figure 7: <Guest> Search Product Use Case 43](#_Toc427058998)

[Figure 8: <Guest> View Detail Use Case 45](#_Toc427058999)

[Figure 9: <Guest> View Detail Use Case 47](#_Toc427059000)

[Figure 10: <Customer> Overview User Case 49](#_Toc427059001)

[Figure 11: <Store Owner> Overview Use Case 50](#_Toc427059002)

[Figure 12: <StoreOwner> Manage New Request Use Case 51](#_Toc427059003)

[Figure 13: <Store Owner> Accept Request Use Case 54](#_Toc427059004)

[Figure 14: <Store Owner> Refuse Request Use Case 56](#_Toc427059005)

[Figure 15: <Store Owner> Manage Accepted Request 58](#_Toc427059006)

[Figure 16: <Store Owner> Update Request 61](#_Toc427059007)

[Figure 17: <Store Owner> Accept Product 63](#_Toc427059008)

[Figure 18: <Store Owner> Refuse Product 65](#_Toc427059009)

[Figure 19: <Store Owner> View Refused Request 67](#_Toc427059010)

[Figure 20; <Store Owner> View Cancelled Request 69](#_Toc427059011)

[Figure 21: <Store Owner> View Available Product 71](#_Toc427059012)

[Figure 22: <Store Owner> Publish On Web 72](#_Toc427059013)

[Figure 23: <Store Owner> View Sold Product 76](#_Toc427059014)

[Figure 24: <Store Owner> Process Payment 77](#_Toc427059015)

[Figure 25: <Store Owner> View Canceled Product 79](#_Toc427059016)

[Figure 26: <Store Owner> Disagree Return Product 81](#_Toc427059017)

[Figure 27: Complete Return Product 83](#_Toc427059018)

[Figure 28: <Store Owner> View Ordered Product 85](#_Toc427059019)

[Figure 29: <Store Owner> Send Product Price To Customer 87](#_Toc427059020)

[Figure 30: <Store Owner> Send Product Price To Customer 89](#_Toc427059021)

[Figure 31: <Store Owner> Extend Consignment Period 91](#_Toc427059022)

[Figure 32: <Store Owner> View On Web Product 93](#_Toc427059023)

[Figure 33: <Store Owner> Update Product 95](#_Toc427059024)

[Figure 34: <Store Owner> Cancel Product On Web 98](#_Toc427059025)

[Figure 35: <Store Owner> View Consignment Statistic 100](#_Toc427059026)

[Figure 36: < Admin > Create User Use case 105](#_Toc427059027)

[Figure 37: < Admin > Edit User Use case 107](#_Toc427059028)

[Figure 38: < Admin > Active/Deactive User Use case 109](#_Toc427059029)

[Figure 38: <System> Search Product Online 110](#_Toc427059030)

[Figure 39: Conceptual Diagram 113](#_Toc427059031)

[Figure 40: HPS System Architectural 114](#_Toc427059032)

[Figure 41: Component Diagram 116](#_Toc427059033)

[Figure 42: Class Diagram 117](#_Toc427059034)

[Figure 43: <Store Owner> Publish On Web 126](#_Toc427059035)

[Figure 44: <Store Owner> Pay Consignor 126](#_Toc427059036)

[Figure 45: <Store Owner> Process Payment 127](#_Toc427059037)

[Figure 46: <Store Owner> Agree Return Product 127](#_Toc427059038)

[Figure 47: <Store Owner> Disagree Return Product 128](#_Toc427059039)

[Figure 48: <Store Owner> Sell Product 128](#_Toc427059040)

[Figure 49: <Store Owner> Send Product Price To Customer 129](#_Toc427059041)

[Figure 50: <Store Owner> Cancel Orders 129](#_Toc427059042)

[Figure 51: <Store Owner> Extend Consignment Period 130](#_Toc427059043)

[Figure 52: <Store Owner> Update Product 130](#_Toc427059044)

[Figure 53: <Store Owner> View Available Product 131](#_Toc427059045)

[Figure 54: <Store Owner> View Consignment Statistic 131](#_Toc427059046)

[Figure 33: <Store Owner> Refuse Request 132](#_Toc427059047)

[Figure 35: <Store Owner> Accept Product 132](#_Toc427059048)

[Figure 36: <Store Owner> Refuse Product 133](#_Toc427059049)

[Figure 37: <Store Owner> Update Request 133](#_Toc427059050)

[Figure 39: <System> Search Product Online 134](#_Toc427059051)

**Definition, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| HPS | Hanging Product Store |
| App | Application |
| OS | Operating System |

# 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 hobbit, 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.Problem

## Proposed Solution

Our system include a web app helps user sell consigned products and 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.
* Store owner can manage consignment request, manage receive product, raise their products on website and manage order product. The system support Store owner tracking consigned product status, payment online method, price suggestion, website statistics.
* Mobile Application:
* StoreOwner can use their Android phone to manage receive product.

### Advantages and disadvantages

* Advantages:
* Support price suggestion.
* Support online payment method.
* Support website statistics.
* Support notification about consigned product status.
* Support tracking product status.
* Disadvantages:
* Support only fashion product.
* Internet connection is required.

## Functional Requirements

Function requirements of the system are listed as below:

* Account management
* Product items management
* Raise products on website
* Order management
* Website’s statistics
* Price suggestion
* Online payment method
* Product status notification
* Product status tracking

## 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 1: Roles And Responsibilities

# 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 consignor 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 and deal with consignors.

### 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 a consignor, especially inexperienced consignor is in need of consignment, they couldn't know how to find a good consignee and wonder if that consignee is reliable. And when product has already been consigned, 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 manage receive 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 payment 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.

#### Development Environment

##### Hardware requirement

For mobile:

|  |  |  |
| --- | --- | --- |
| Mobile | Minimum Requirements | Recommended |
| Internet Connection | 2 Mbps | 4 Mbps |
| Operating System | Android 4.4 | Android 5.0 |
| Hardware | Wifi supported | Wifi supported |
| Memory | 1GB | 2GB |

Table 2: Hardware Requirement For Mobile

For server:

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window Server 2008 | Window Server 2008 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53Gz |

Table 3: Hardware Requirement For Server

##### Software requirement

* GitHub & TortoiseGit: used for source control.
* NetBean 8.0.2 and Apache Tomcat 8.0.15: used to develop and implement.
* Android Studio 2.1: use to code client application
* Database: Microsoft SQL Server 2008.
* Genymotion 2.4.0: use to run and test Android app.
* StarUML: used to create models and diagrams.
* Skype: use for meeting and conservation between group members

## Project Organization

### Software Process Model

To determine the right process model for the software process, we answer ten guidelines questions (Sommerville, Software Engineering, 2011):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Question** | **Answer** | **Plan-Driven** | **Agile** |
| **1.** | Is it important to have a very detailed specification and design before moving to implementation? If so, you probably need to use a plan-driven approach. | We need a correct specification and design to coding but a very detailed one is not necessary. | 0.5 | 0.5 |
| **2.** | Is an incremental delivery strategy, where you deliver the software to customers and get rapid feedback from them, realistic? If so, consider using agile methods. | No. This is generic product, not bespoke product. A rapid feedback from target customer is not necessary. | 1 | 0 |
| **3.** | How large is the system that is being developed? Agile methods are most effective when the system can be developed with a small co-located team who can communicate informally. This may not be possible for large systems that require larger development teams so a plan-driven approach may have to be used. | This system will be developed in four month by four people, so we think it is a medium system. | 0.5 | 0.5 |
| **4.** | What type of system is being developed? Systems that require a lot of analysis before implementation (e.g., real-time system with complex timing requirements) usually need a fairly detailed design to carry out this analysis. A plan-driven approach may be best in those circumstances. | This is an ecommerce system. Core components require analysis carefully. | 1 | 0 |
| **5.** | What is the expected system lifetime? Long-lifetime systems may require more design documentation to communicate the original intentions of the system developers to the support team. However, supporters of agile methods rightly argue that documentation is frequently not kept up to date and it is not of much use for long-term system maintenance. | This system may operate in two or three years. | 1 | 0 |
| **6.** | What technologies are available to support system development? Agile methods often rely on good tools to keep track of an evolving design. If you are developing a system using an IDE that does not have good tools for program visualization and analysis, then more design documentation may be required. | We have a good tool to tracking change. The IDE support tracking phase. | 0.5 | 0.5 |
| **7.** | How is the development team organized? If the development team is distributed or if part of the development is being outsourced, then you may need to develop design documents to communicate across the development teams. You may need to plan in advance what these are. | We are a small with four people and we are all living in Ho Chi Minh city. We can have meeting every day. | 0 | 1 |
| **8.** | Are there cultural issues that may affect the system development? Traditional engineering organizations have a culture of plan-based development, as this is the norm in engineering. This usually requires extensive design documentation, rather than the informal knowledge used in agile processes. | We are all young people and we share common software development knowledge. However, we come from difference part of country and we never met before. | 0.5 | 0.5 |
| **9.** | How good are the designers and programmers in the development team? It is sometimes argued that agile methods require higher skill levels than plan-based approaches in which programmers simply translate a detailed design into code. If you have a team with relatively low skill levels, you may need to use the best people to develop the design, with others responsible for programming. | Although we are confident about software development skill. We are still student and we have no or little experience. | 1 | 0 |
| **10.** | Is the system subject to external regulation? If a system has to be approved by an external regulator (e.g., the Federal Aviation Authority [FAA] approve software that is critical to the operation of an aircraft) then you will probably be required to produce detailed documentation as part of the system safety case. | No | 0 | 0 |
| **Total** | | | **6** | **3** |

Table 4: Software Process Guideline Question Decision

As the result, the plan-driven process we choose is waterfall model because of its easy to use and commonly used in software process.



Figure 1: The Waterfall Model (Sommerville, Software Engineering, 2011)

Because of the small size of the team (four people included project manager), we may overlap five phases in waterfall model to cope with change better and to reduce the wasted time (due to the blocking nature of the waterfall model – the following phase must wait until the previous phase has finished).

During Requirement Definition, we will design some component that has clear requirement. During coding, we will fix design problem and so on. We will modify the document produced in each phase to reflect the changes made.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Full name | Team Role | Responsibilities |
| **1** | Kieu Trong Khanh | * Senior Manager | * Monitor project * Mentor Project Manager * Specify user requirement * Review documents * Support technology |
| **2.** | Quan Tien Dan | * Project Manager * Business Analyst * Developer * Tester | * Monitor project * Plan and execute project * Design * Programing * Testing |
| **3.** | Tran Quang Phuc | * Team member * Business Analyst * Developer * Tester | * Assist in tools and techniques * Design * Programing * Testing |
| **4** | Nguyen Huy Hoang | * Team member * Business Analyst * Developer * Tester | * Assist in tools and techniques * Design * Programing * Testing |
| **5** | Huynh Cong Duc | * Team member * Business Analyst * Developer * Tester | * Assist in tools and techniques * Design * Programing * Testing |

Table 5: Project Roles and Responsibilities

### Tools and Techniques

* Back-end technologies:
  + Website: Java EE
  + Web service: Java + Jersey
  + Third-party libraries: Gson 2.3.1, Amazon Product Advertising API, Twilio API.
* Front-end technologies: HTML5, CSS3, JavaScript, Jquery, AJAX.
* Mobile: Android Lollipop 4.0 and later
* Web Server: Installed Tomcat 8.0.15.0
* Database Management System: MSSQL Server 2008 R2 Enterprise.

## Project Management Plan

### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constraints** | **Risks** |
| Requirement Definition | The process of establishing what services are required and the constraints on the system’s operation and development. | Software requirement specification:   * System models * User and system requirements   Executable prototype  Acceptance test plan | 52 man days |  | Lack of domain knowledge.  Communication misunderstanding.  Conflict objectives, conflict viewpoints. |
| System and Software Design | Design a software structure and an overall system architecture that realizes the specification. | System design description:   * System architecture * Database specification * Interface specification * Component Specification   System integration test plan | 52 man days | Platform information  Software requirement specification  Data description | Change to requirements that require major design rework are proposed.  Specifications of essential interfaces are not available on schedule.  The size of the system has been underestimated. |
| Implementation and Unit Testing | Translate design structure into an executable program or program unit.  Unit testing involves verifying that each unit meets its specification. | Program Units  Test case | 90 man days | System design description | Software tool cannot be integrated.  The time required to develop the software is underestimated.  The underlying technology on which the system is build is superseded by new technology. |
| Integration and System Testing | Integrated the individual program unit.  Test the complete system to ensure that the software requirements have been meet. | Executable software  Test result | 58 man days | System design description  System integration test plan  Acceptance test plan | Reusable software components contain defects that mean they cannot be used as planned. |
| Operation and Maintenance | This is the longest life cycle phase.  The system is installed and put into practical use.  Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system’s services as new requirements are discovered. | Installation guide  User guide | 92 man days | Executable software  Software requirement specification | A competitive product is marketed before the system is completed.  The database used in the system cannot process as many transactions per second as expected. |

Table 6: Software Development Lifecycle Detail

### Phase Detail

#### Phase 1: Requirement Definition

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| Collect requirements | Find which systems currently provide similar service, their advantages and dis advantages. | DanQT, PhucTQ, HoangNH, DucHC |
| Identify and clarify main functions | Define main functions of system. | DanQT, PhucTQ, HoangNH, DucHC |
| Create System Introduction | Complete Introduction Report. | DanQT, PhucTQ, HoangNH, DucHC |
| Software Project Management Plan. | Prepare Project Management Plan. | DanQT, PhucTQ, HoangNH, DucHC |
| Website Prototype. | Build a prototype of system. | DanQT, PhucTQ, HoangNH, DucHC |
| SRS | Create SRS document. | DanQT, PhucTQ, HoangNH, DucHC |

Table 7: Phase 1: Requirement Analysis

#### Phase 2: System and Software Design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| Architecture Design | Create an architecture design that can meet system requirements. | DanQT, PhucTQ, HoangNH, DucHC |
| Detailed Design | Create a detailed design with diagrams and relevant artifacts. | DanQT, PhucTQ, HoangNH, DucHC |
| Database Design | Design database. | DanQT, PhucTQ, HoangNH, DucHC |
| Technology research | Research solution for ordered functions. | DanQT, PhucTQ, HoangNH, DucHC |

Table 8: Phase 2: System and Software Design

#### Phase 3: Implementation and Unit Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| Price suggestion and payment function. | Suggest price for consignor and consignee.  Perform online payment method. | DanQT |
| Price negotiation between consignor and consignee. | Submit desirable price.  Accept suggested price.  Refuse suggested price. | PhucTQ |
| Raise product on web and advanced search function. | Raise product on web by season.  Search product function.  Support shopping online. | HoangNH |
| Consign product and track consigned-product status. | Consign product.  Support tracking and notification consigned-product status. | DucHC |
| Unit testing. | Create and perform unit testing for price suggestion and online payment function. | DanQT |
| Create and perform unit testing for price negotiation function. | PhucTQ |
| Create and perform unit testing for raise product on web and advanced search function. | HoangNH |
| Create and perform unit testing for consign product and tracking consigned-product status function. | DucHC |

Table 9: Phase 3: Implementation and Unit Testing

#### Phase 4: Integration and System Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| Integration Testing | Write test case and test for integrate functions. | DanQT, PhucTQ, HoangNH, DucHC |
| System Testing | Write test case and test for whole system. | DanQT, PhucTQ, HoangNH, DucHC |

Table 10: Phase 4: Integration and System Testing

#### Phase 5: Operation and Maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| Installation Guide | Complete installation guide | DanQT, PhucTQ, HoangNH, DucHC |
| User’s Guide | Complete user’s guide | DanQT, PhucTQ, HoangNH, DucHC |

Table 11: Phase 5: Operation and Maintenance

### All Meeting Minute

All meeting minute are placed in Meeting Minute at GitHub repository.

## Coding Convention

* Naming Convention:
* For variable and function name: Camel case
* For class name: Pascal case
* Code Examples
* Follow “Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999”. <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

# 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 base on consignmentID.

### 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.
* View selling statistic.
* Edit profile.

### Admin Requirement

An Admin of this system has the following abilities:

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

## User Requirement Specification

### External Interface Requirement

#### User Interface

* General requirement for graphic user interface design of mobile device is the GUI should be simple, clear, intuitive, and reminiscent.
* The interface design is an iterate process includes design, sketching, prototyping, user assessment.
* Some design principles are taken into consideration:
  + Material design for android Lollipop – Google [Ref: <https://www.google.com/design/spec/material-design/introduction.html>]

#### Hardware Interface

* N/A

#### Software Interface

* Run with FireFox (version 37.0.1), Chrome (version 42.0.2311.90 m) browser and must support Javascript and HTML5.

#### Communication Protocol

* Using HTTP protocol

### System Overview Use Case

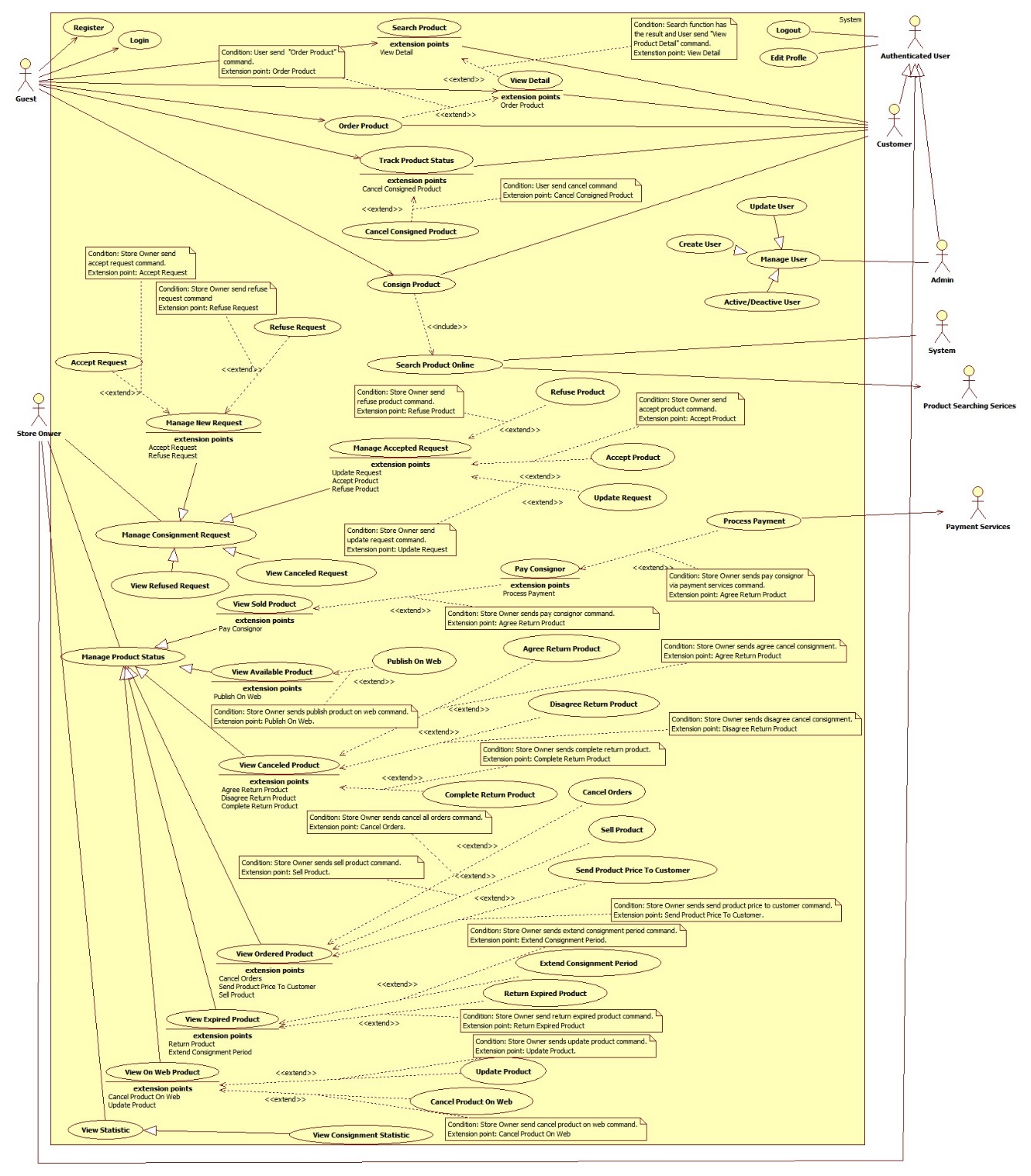


Figure 2: System Overview Use Case

### List of Use Case

#### <Guest> Overview Use Case

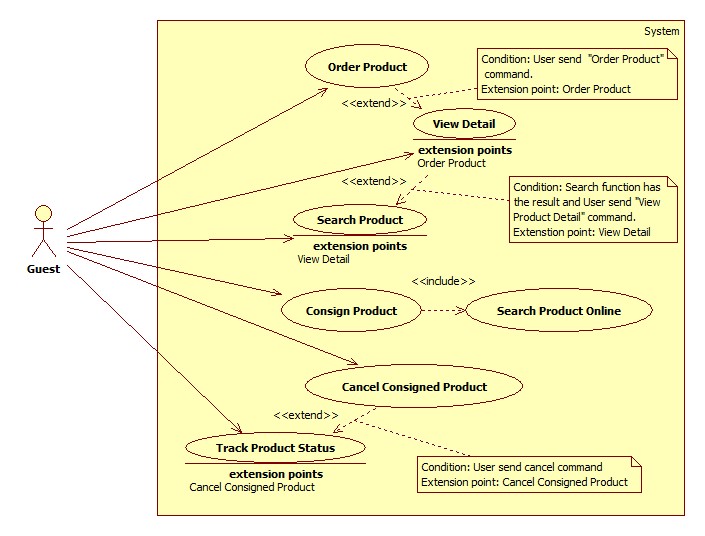


Figure 3: <Guest> Overview Use Case

##### <Guest> Consign Product Use Case

**Use Case Diagram**

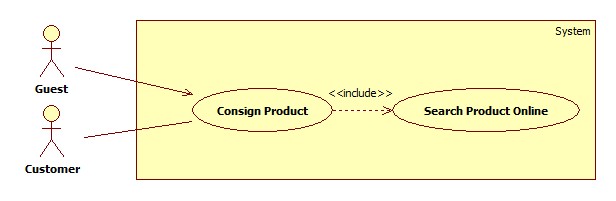


Figure 4: <Guest> Consign Product Use Case

**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:   * “Tên sản phẩm”:   + Free text input.   + Min-length: 5 characters.   + Max-length: 50 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 -> 100   + 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, 3, 4] | 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:   * “Trở về”: send command to go back to providing-product’s information step.   “Bước kế”: send command to finish choosing product. | | 3 | User chooses a product and inform system.  [Alternative 5, 6, 7] | 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. * “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 8, 9] | 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. * “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). * “Thay ảnh”: change image by choosing one in user’s computer (where there is 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 10, 11, 12, 13, 14, 15, 16] | 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 types in “Tên sản phẩm” input in providing-product’s information step. | “Mã sản phẩm” will be disabled and cannot be typed. System will text in “Tên sản phẩm” field to check similar products online. | | 2 | User deletes text in “Tên sản phẩm” input in providing-product’s information step. | Mã sản phẩm” will be enabled and can be typed. | | 3 | User types in “Mã sản phẩm” input in providing-product’s information step. | “Tên sản phẩm” will be disabled and cannot be typed. System will text in “Mã sản phẩm” field to check exact products online. | | 4 | User deletes text in “Mã sản phẩm” input in providing-product’s information step. | Mã sản phẩm” will be enabled and can be typed. | | 5 | User sends request to go back to providing-product’s information step in product-choosing step. | Systems shows providing product’s info step. | | 6 | 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. | | 7 | 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. | | 8 | 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. | | 9 | 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. | | 10 | User have already chosen a product in product-choosing step. | “Hình ảnh” will show an image of chosen product automatically. | | 11 | User chooses “Tiền mặt” option in providing personal info step. | “Mã tài khoản” free text input will be hidden. | | 12 | User chooses “Tài khoản paypal” option in providing personal info step. | “Mã tài khoản” free text input will be shown and be required. | | 13 | User selects image in providing product’s info step. | System requires user to browse image in user’s computer and upload on page. | | 14 | User changes image in providing product’s info step. | System requires user to browse image in user’s computer and upload on page. | | 15 | User removes image in providing product’s info step. | Image is removed. | | 16 | 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 empty or invalid. | System shows error message: “Xin nhập đúng mã tài khoản”. |   **Relationships**:   * Including use-case: Include Search Product Online.   **Business Rules**:   * 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> Track product status Use Case

**Use Case Diagram**

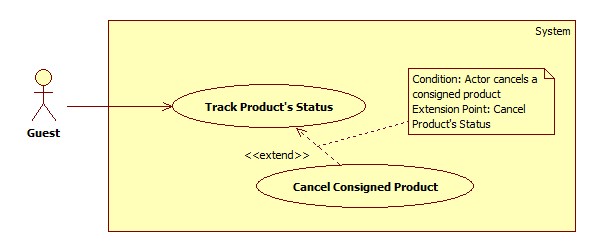
****

Figure 5: <Guest> Track product status Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS002** | | | |
| **Use Case No.** | HPS002 | Use Case Version | 2.0 |
| **Use Case Name** | Track product status | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | May 29, 2015 | Priority | High |
| **Actor**:   * Guest   **Summary**:   * This use case allows guest to track product status by code.   **Goal**:   * Guest views product’s current status successfully.   **Triggers**:   * Guest sends tracking product’s status command.   **Preconditions**: N/A  **Post Conditions**:   * **Success:** View product’s current status. * **Fail:** Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends tracking product’s status command. | System shows guest following:   * “Mã sản phẩm”: free text input * “Kiểm tra”: send command to track product’s status. | | 2 | Guest inputs “Mã sản phẩm” and sends command to track product’s status. | System finds product with provided code and shows guest with following information:   * “Hình ảnh”: A large field to show image of product. * “Tên sản phẩm”: label. * “Loại”: label. * “Hãng”: label * “Ngày ký gửi”: label. * “Trạng thái”: label. * “Người ký gửi”:   + “Tên”: label.   + “Địa chỉ”: label.   + “Điện thoại”: label.   + “Email”: label. * “Cửa hàng”: label.   System also shows following:   * “Hủy ký gửi”: send command to cancel consignment.   [Exception: 1, 2] |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Inputted value of “Mã sản phẩm” field is empty. | System shows error message: “Xin nhập mã sản phẩm”. | |  | System cannot find product with inputted “Mã sản phẩm”. | System shows error message: “Sản phẩm với mã số này không tồn tại”. |   **Relationships**:   * Extending Use-case: Cancel Consigned Product (Guest sends a request to cancel a consigned products) and Extend Consigned Product (Guest send a request to extend product).   **Business Rules**:   * Guest can only track product’s status by code. * Code is ID of consignment. * User can only cancel product before it is ordered. | | | |

##### <Guest> Cancel Consigned Product Use Case

**Use Case Diagram**

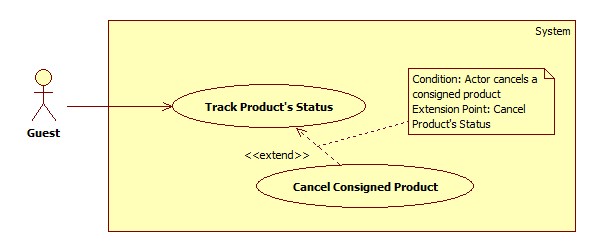
****

Figure 6: <Guest> Cancel Consigned Product Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS003** | | | |
| **Use Case No.** | HPS003 | Use Case Version | 2.0 |
| **Use Case Name** | Cancel consigned product | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | May 29, 2015 | Priority | High |
| **Actor**:   * Guest   **Summary**:   * This use case allows guest to cancel a consigned product.   **Goal**:   * Guest cancels a consigned product successfully.   **Triggers**:   * Guest sends canceling consigned product command.   **Preconditions**:   * Guest is viewing a consigned product’s information. * Status of product is neither “Sold” nor “Completed”.   **Post Conditions**:   * **Success:** Product’s status is changed to “Canceled”. * **Fail:** Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends canceling consigned product command.  [Alternative 1, 2] | System shows a confirming message if guest really wants to cancel this consigned product with two options “Đồng ý” and “Không”. | |  | Guest chooses “Đồng ý” option.  [Alternative 3] | Product’s status is changed to “Đã hủy” or “Đăng ký hủy”. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends canceling consigned product command when product is not received | System shows a message to make sure they really want to cancel it with 2 options: “Đồng ý” or “Không”. | | 2 | Guest sends canceling consigned product command when product is already received but not ordered. | System shows a message to make sure they really want to cancel it with a price for storing product and 2 options: “Đồng ý” hoặc “Không”. | | 3 | Guest chooses “No” option. | Message will disappear. |   **Exceptions**: N/A  **Relationships**:   * Extended Use-case: Track product status.   **Business Rules**:   * Guest can only cancel product before it’s ordered. * If guest cancels before store owner gets product, consign product process will be stopped. * If guest cancels after store owner gets product, guest will be informed to take back product in a specific time and pay a forfeit for canceling consignment. After that time and guest doesn’t come to take back product, store owner will have it. | | | |

##### <Guest> Search Product Use Case

**Use Case Diagram**

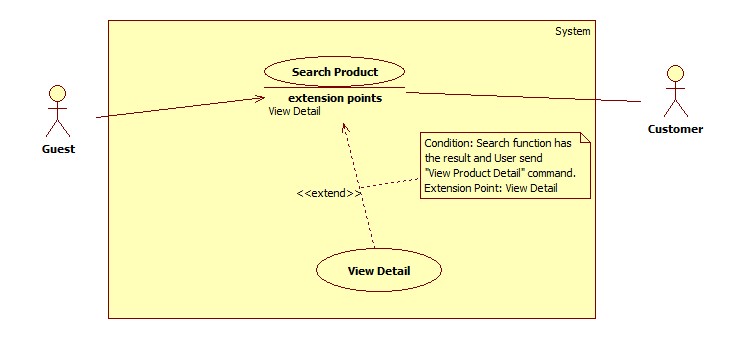


Figure 7: <Guest> Search Product Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS004 | | | |
| **Use Case No.** | HPS004 | Use Case Version | 2.0 |
| **Use Case Name** | Search Product | | |
| **Author** | Nguyen Huy Hoang | | |
| **Date** | May 22, 2015 | Priority | High |
| **Actor**:   * Guest/Customer   **Summary**:   * This use case allows user to search product information by its name and category.   **Goal**:   * A list of products will be shown to user.   **Triggers**:   * User can search product to get product’s information. * User can choose top product’s category include: “Nam”, “Nữ”, “Trẻ em” or “Tất cả”, inputs to a free text input and send “Search Product” command.   **Preconditions**:   * User is in home page or category page.   **Post Conditions**:   * Success: Show search results which include a list of product. * Fail: Show an error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User choose product’s category include: “Nam”, “Nữ”, “Trẻ em” or “Tất cả”. |  | | 2 | User inputs the product’s name into free text input. |  | | 3 | User send “Search Product” command. | [Exception 1]  System will show a list of product as a grid and order by product name.  [Alternative 1] |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Search not found | System will show a message: “Không tìm thầy sản phẩm nào phù hợp.” |   **Exceptions**: N/A  **Relationships**:   * Extending Use-cases: Extended by View Detail (Search function has the result and user send “View Product Detail” command).   **Business Rules**:   * Search by product name and category. * All products with relative name which user inputted will be show. The results will show list of product as a gird. | | | |

##### <Guest> View Detail Use Case

**Use Case Diagram**

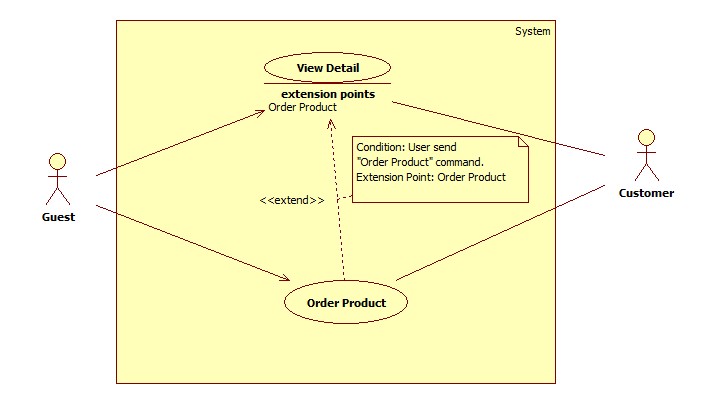


Figure 8: <Guest> View Detail Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS005 | | | |
| **Use Case No.** | HPS005 | Use Case Version | 2.0 |
| **Use Case Name** | View Detail | | |
| **Author** | Nguyen Huy Hoang | | |
| **Date** | May 22, 2015 | Priority | High |
| **Actor**:   * Guest/Customer   **Summary**:   * This use case allows user to view product detail information.   **Goal**:   * A product detail will be shown to user.   **Triggers**:   * User send “View Product Detail” command.   **Preconditions**:   * At least 1 product item available.   **Post Conditions**:   * Success: Show product detail information in detail page. * Fail: Show an error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click “XEM CHI TIẾT” button. | System will show product information which contains:   * “Tên” : Label * “Tình trạng”: Label * “Số Serial” : Label * “Mô tả”: Textarea * “Hình ảnh”: Image * “Tên chủ cửa hàng”: Label * “Email”: Label * “Số điện thoại”: Label * “Địa chỉ”: Label * A command to order product. |   **Alternative Scenario**: N/A  **Exceptions**: N/A  **Relationships**:   * Extending Use-cases: Extend from Search Product (Search keyword has the result and user click a result item). Extended by Order Product(User send “Order Product” command).   **Business Rules**:   * Details of product will be displayed to user. | | | |

##### <Guest> Order Product Use Case

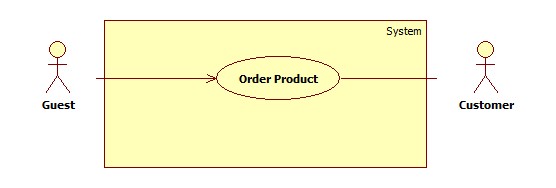


Figure 9: <Guest> View Detail Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS006 | | | |
| **Use Case No.** | HPS006 | 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. | 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.   [Alternative 1] | | 2 | User input information. |  | | 3 | User send “Confirm Information” command. | [Exception 1,2]  [Alternative 2]  System create an order of this product with user information. | |  |  |  |   **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. | | | |

#### <Customer> Overview User Case

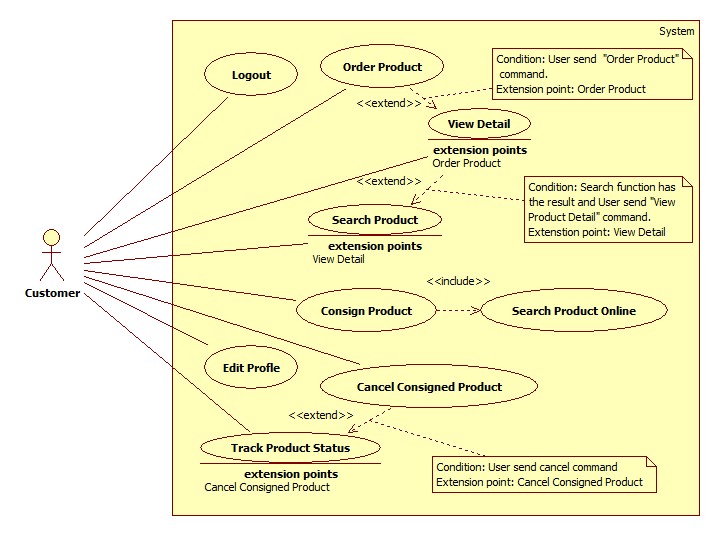


Figure 10: <Customer> Overview User Case

#### <Store Owner> Overview Use Case

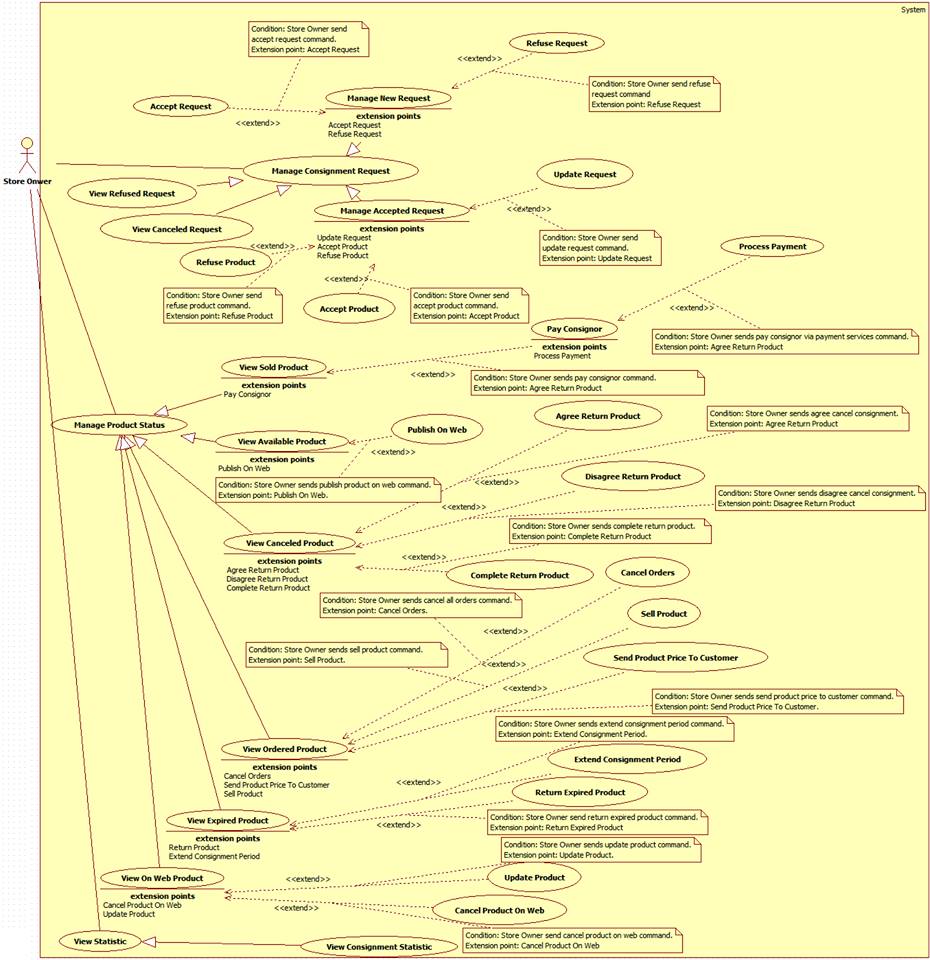


Figure 11: <Store Owner> Overview Use Case

##### <StoreOwner> Manage New Request Use Case

**Use Case Diagram**

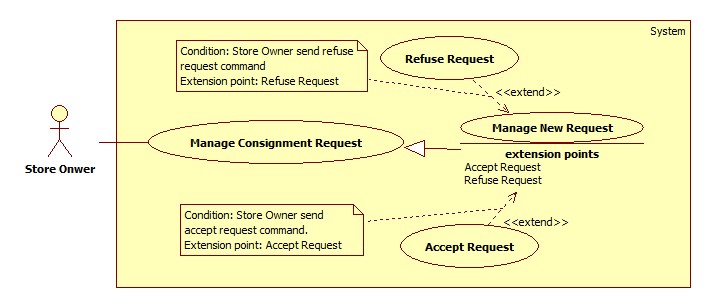


Figure 12: <StoreOwner> Manage New Request Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS007** | | | |
| **Use Case No.** | HPS007 | Use Case Version | 2.0 |
| **Use Case Name** | Manage New Request | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Show list of new request for current store.   **Goal**:   * Store Owner view list of new request.   **Triggers**:   * Store Owner send request the system to view new request list.   **Preconditions**:   * Store Owner has logged onto the system.   **Post Conditions**:   * **Success**: List of consignment request is shown. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send request the system to view new request list. | System shows new request list include following information:   * “Tên sản phẩm”: label. * “Mã ký gửi”: label. * “Ngày gửi yêu cầu”: label. * “Ngày hẹn”: label. * “Giá mong muốn”: label.   System also shows on each row a “Xem” component for viewing details of that request. | | 2 | Store owner chooses a request to view details. | System shows details of that request with listed information:   * “Ảnh”: image of product. * “Tên sản phẩm”: label. * “Loại”: label. * “Hãng”: label. * “Mô tả”: label. * “Độ mới”: label. * “Họ tên”: label. * “Điện thoại”: label. * “Email”: label. * “Địa chỉ”: label. * “Mã tài khoản”: label. * “Ngày ký gửi”: label. * “Ngày hẹn”: label. * “Cách giao hàng”: label. * “Ngày giao hàng”:   + Default by the first date of “Ngày hẹn”.   + Required. * “Giá tham khảo”: label. * “Giá mong muốn”: label.   System also shows some components below:   1. “Chấp nhận”: accept this request. 2. “Từ chối”: refuse this request. 3. “Đóng”: close details to go back to list new request.   [Exception: 1] |   **Alternative**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to view details when request has already handled. | System shows message: “Yêu cầu này đã được xử lý”. |   **Relationships**:   * Included Use-case: N/A. * Extended Use-case: N/A. * Including Use-case: N/A. * Extending Use-case: Accept Request (Store Owner sends command to accept request), Refuse Request (Store Owner sends command to refuse request).   **Business Rules**:   * Filter bar shall be shown to help Store Owner find consignment faster. * List are sorted by on request date in reverse chronological order. The most recent request appears first in the list, while the oldest one appears last. * This list shows new requests which haven’t been handled. * Store Owner will consider this request if store owner wants to accept it or not. | | | |

##### <Store Owner> Accept Request Use Case

**Use Case Diagram**

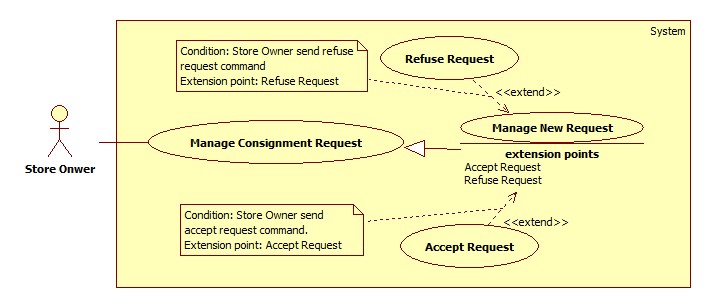


Figure 13: <Store Owner> Accept Request Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS008** | | | |
| **Use Case No.** | HPS008 | 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. | System make that request accepted and inform Store Owner that the request has been accepted. |   **Alternative**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to accept a request when request has already handled. | System shows message: “Yêu cầu này đã được xử lý”. |   **Relationships**:   * Included Use-case: N/A. * Extended Use-case: Manage New Request. * Including Use-case: N/A. * Extending Use-case: N/A   **Business Rules**:   * After accepting request, this request’s status will change to “Đã chấp nhận yêu cầu”. * If accepted request has the value of “Cách giao hàng” is “Cửa hang đến nhận” and “Ngày giao hàng” is current day, system will inform Store Owner in mobile application about this request immediately. * After accepted request, Store Owner can update it later in managing accepted request. | | | |

##### <Store Owner> Refuse Request Use Case

**Use Diagram**

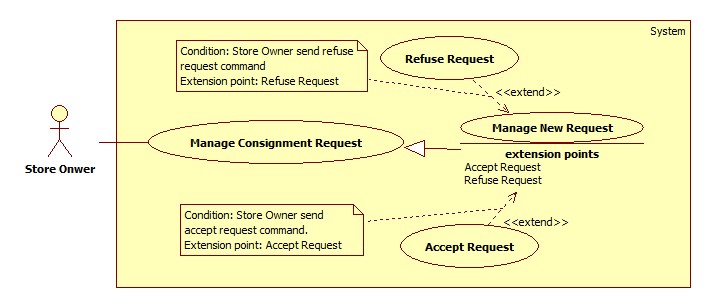


Figure 14: <Store Owner> Refuse Request Use Case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS009** | | | |
| **Use Case No.** | HPS009 | Use Case Version | 2.0 |
| **Use Case Name** | Refuse Request | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Store Owner refuse a new request.   **Goal**:   * Store Owner successfully refuse a request.   **Triggers**:   * Store Owner send command to refuse a request.   **Preconditions**:   * Store Owner has logged onto the system. * Store Owner is viewing details of a new request.   **Post Conditions**:   * **Success**: Successfully refuse 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 refuse a request. | System ask Store Owners to inform the reason. | | 2 | Store Owners provides a reason and send it to system. | System saves this reason and make this request refused.  [Exception 1] |   **Alternative**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to refuse a without providing a reason | System ask Store Owners to providing a reason. |   **Relationships**:   * Included Use-case: N/A. * Extended Use-case: Manage New Request. * Including Use-case: N/A. * Extending Use-case: N/A   **Business Rules**:   * After refusing request, this request’s status will change to “Đã từ chối”. * User can check this request in viewing refused request. | | | |

##### <Store Owner> Manage Accepted Request

**Use Diagram**



Figure 15: <Store Owner> Manage Accepted Request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS010** | | | |
| **Use Case No.** | HPS010 | Use Case Version | 2.0 |
| **Use Case Name** | Manage Accepted Request | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Show list of accepted request for current store.   **Goal**:   * Store Owner view list of accepted request.   **Triggers**:   * Store Owner send request the system to view accepted request list.   **Preconditions**:   * Store Owner has logged onto the system.   **Post Conditions**:   * **Success**: List of consignment request is shown. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store owner send request the system to view new request list. | System shows accepted request list include following information:   * “Tên sản phẩm”: label. * “Mã ký gửi”: label. * “Ngày nhận yêu cầu”: label. * “Ngày hẹn”: label. * “Giá mong muốn”: label. * “Cách giao hàng”: label   System also shows on each row a “Xem” component for viewing details of that request. | | 2 | Store owner chooses a request to view details. | System shows details of that request with listed information:   * “Ảnh”: image of product. * “Tên sản phẩm”:   + Free text input   + Length: 5 – 100.   + Required. * “Loại”:   + Choose from a list (“Giày nam”, “Áo khác nam”, “Túi xách nữ”, “Giày dép nữ”…).   + Required. * “Hãng”:   + Free text input.   + Length: 0-100. * “Mô tả”: Free text input. * “Độ mới”:   + Free text input.   + Format: Number 1- 99.   + Required. * “Họ tên”:   + Free text input.   + Format: Number 1- 99.   + Required. * “Điện thoại”:   + Free text input   + Format: number.   + Length: 9-11.   + Required. * “Email”: Free text input. * “Địa chỉ”: Free text input. * “Cách thanh toán”:   + Store Owner chooses from 2 option: “Tiền mặt” or “Tài khoản paypal”.   + Default chosen by who send this request. * “Mã tài khoản”: Free text input. * “Ngày ký gửi”: label. * “Ngày hẹn”: label. * “Cách giao hàng”:   + Store Owner chooses from 2 option: “Cửa hang đến nhận” or “Tự mang đến”.   + Default chosen by who send this request.   + Required. * “Ngày giao hàng”:   + Default by the first date of “Ngày hẹn”.   + Required. * “Giá tham khảo”: label. * “Giá mong muốn”: label. * “Giá thỏa thuận”:   + Free text input.   + Format: Number.   + Required.   System also shows some components below:   * “Cập nhật”: update this request * “Chấp nhận”: accept product. * “Từ chối”: refuse product. * “Đóng”: close details to go back to list accepted request.   [Exception: 1] |   **Alternative**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to view details when request has already handled. | System shows message: “Yêu cầu này đã được xử lý”. |   **Relationships**:   * Included Use-case: N/A. * Extended Use-case: N/A. * Including Use-case: N/A. * Extending Use-case: Update Request, (Store Owner sends command to update request), Accept Product (Store Owner sends command to accept product), Refuse Product (Store Owner sends command to refuse product).   **Business Rules**:   * Filter bar shall be shown to help Store Owner find consignment faster. * List are sorted by on request date in reverse chronological order. The most recent request appears first in the list, while the oldest one appears last. * This list shows accepted requests which haven’t been handled. * In website application, Store Owner can view all the accepted request including if store has to come to customer house to take product or customers bring product to store. * In mobile application, store owner can only view which those store owner has to come to take product. Store owner will receive on each day a list of products have to be taken in current day and the next day. | | | |

##### <Store Owner> Update Request

**Use Case Diagram**



Figure 16: <Store Owner> Update Request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS011** | | | |
| **Use Case No.** | HPS011 | Use Case Version | 2.0 |
| **Use Case Name** | Update Request | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Store Owner update an accepted request.   **Goal**:   * Store Owner successfully update an accepted request.   **Triggers**:   * Store Owner send command to update request.   **Preconditions**:   * Store Owner has logged onto the system. * Store Owner is viewing details of an accepted request.   **Post Conditions**:   * **Success**: Successfully update accepted request. * **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 update it. | System update this request with provided information.  [Exception 1,2] |   **Alternative**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner does not provide valid information. | System show a message: “Xin nhập đúng thông tin”. | | 2 | Store Owner sends command to update when this request is already handled. | System show a message: “Yêu cầu này đã được xử lý”. |   **Relationships**:   * Included Use-case: N/A. * Extended Use-case: Manage Accepted Request. * Including Use-case: N/A. * Extending Use-case: N/A   **Business Rules**:   * After updating request, this request information will be changed. * If updated request has “Ngày giao hàng” is “current Date and “Cách giao hàng” is “Cửa hàng đến nhận”, system will inform Store Owner in mobile application about this request. | | | |

##### <Store Owner> Accept Product

**Use Case Diagram**



Figure 17: <Store Owner> Accept Product

**Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS012** | | | |
| **Use Case No.** | HPS012 | Use Case Version | 2.0 |
| **Use Case Name** | Accept Product | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Store Owner accept 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. | System saves provided information and changes this request to “Đã nhận hàng”.  [Exception 1, 2] |   **Alternative**: N/A  **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**:   * Included Use-case: N/A. * Extended Use-case: Manage New Request. * Including Use-case: N/A. * Extending Use-case: N/A   **Business Rules**:   * 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 accept an product and bring it back to store. | | | |

##### <Store Owner> Refuse Product

**Use Case Diagram**



Figure 18: <Store Owner> Refuse Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS013** | | | |
| **Use Case No.** | HPS013 | Use Case Version | 2.0 |
| **Use Case Name** | Refuse Product | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Store Owner refuse a product.   **Goal**:   * Store Owner successfully refuse a product.   **Triggers**:   * Store Owner send command to refuse a product.   **Preconditions**:   * Store Owner has logged onto the system. * Store Owner is viewing details of an accepted request.   **Post Conditions**:   * **Success**: Successfully refuse a product. * **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 refuse a product. | System ask Store Owners to inform the reason. | | 2 | Store Owners provides a reason and send it to system. | System saves this reason and make this request refused.  [Exception 1] |   **Alternative**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store Owner sends command to refuse a without providing a reason | System ask Store Owners to providing a reason. |   **Relationships**:   * Included Use-case: N/A. * Extended Use-case: Manage New Request. * Including Use-case: N/A. * Extending Use-case: N/A   **Business Rules**:   * After refusing request, this request’s status will change to “Đã từ chối”. * User can check this request in viewing refused request. | | | |

##### <Store Owner> View Refused Request

**Use Case Diagram**

****

Figure 19: <Store Owner> View Refused Request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS014** | | | |
| **Use Case No.** | HPS014 | Use Case Version | 2.0 |
| **Use Case Name** | View Refused Request | | |
| **Author** | Huynh Cong Duc | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Show list of refused request for current store.   **Goal**:   * Store Owner view list of refused request.   **Triggers**:   * Store Owner send request the system to view refused request list.   **Preconditions**:   * Store Owner has logged onto the system.   **Post Conditions**:   * **Success**: List of consignment request is shown. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send request the system to view new request list. | System shows new request list include following information:   * “Tên sản phẩm”: label. * “Mã ký gửi”: label. * “Ngày gửi yêu cầu”: label. * “Ngày từ chối”: label.   System also shows on each row a “Xem” component for viewing details of that request. | | 2 | Store owner chooses a request to view details. | System shows details of that request with listed information:   * “Ảnh”: image of product. * “Tên sản phẩm”: label. * “Mô tả”: label. * “Độ mới”: label. * “Họ tên”: label. * “Điện thoại”: label. * “Email”: label. * “Địa chỉ”: label. * “Ngày ký gửi”: label. * “Giá mong muốn”: label. * “Trạng thái”: label. * “Lí do”: label. |   **Alternative**: N/A  **Exceptions**: N/A  **Relationships**:   * Included Use-case: N/A. * Extended Use-case: N/A. * Including Use-case: N/A. * Extending Use-case: N/A.   **Business Rules**:   * Filter bar shall be shown to help Store Owner find consignment faster. * List are sorted by on request date in reverse chronological order. The most recent request appears first in the list, while the oldest one appears last. * This list shows refused requests. * Store Owner will review the request store refused and the reason why. | | | |

##### <Store Owner> View Cancelled Request

**Use Case Diagram**

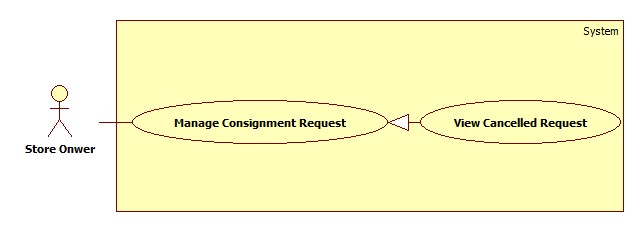


Figure 20; <Store Owner> View Cancelled Request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS015** | | | |
| **Use Case No.** | HPS015 | Use Case Version | 2.0 |
| **Use Case Name** | View Cancelled Request | | |
| **Author** | DucHC | | |
| **Date** | August 8, 2015 | Priority | High |
| **Actor**:   * Store Owner.   **Summary**:   * Show list of refused request for current store.   **Goal**:   * Store Owner view list of refused request.   **Triggers**:   * Store Owner send request the system to view refused request list.   **Preconditions**:   * Store Owner has logged onto the system.   **Post Conditions**:   * **Success**: List of consignment request is shown. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send request the system to view new request list. | System shows new request list include following information:   * “Tên sản phẩm”: label. * “Mã ký gửi”: label. * “Ngày gửi yêu cầu”: label. * “Ngày từ chối”: label.   System also shows on each row a “Xem” component for viewing details of that request. | | 2 | Store owner chooses a request to view details. | System shows details of that request with listed information:   * “Ảnh”: image of product. * “Tên sản phẩm”: label. * “Mô tả”: label. * “Độ mới”: label. * “Họ tên”: label. * “Điện thoại”: label. * “Email”: label. * “Địa chỉ”: label. * “Ngày ký gửi”: label. * “Giá mong muốn”: label. * “Trạng thái”: label. |   **Alternative**: N/A  **Exceptions**: N/A  **Relationships**:   * Included Use-case: N/A. * Extended Use-case: N/A. * Including Use-case: N/A. * Extending Use-case: N/A.   **Business Rules**:   * Filter bar shall be shown to help Store Owner find consignment faster. * List are sorted by on request date in reverse chronological order. The most recent request appears first in the list, while the oldest one appears last. * This list shows refused requests. * Store Owner will review the request store refused and the reason why. | | | |

##### <Store Owner> View Available Product

**Use Case diagram**

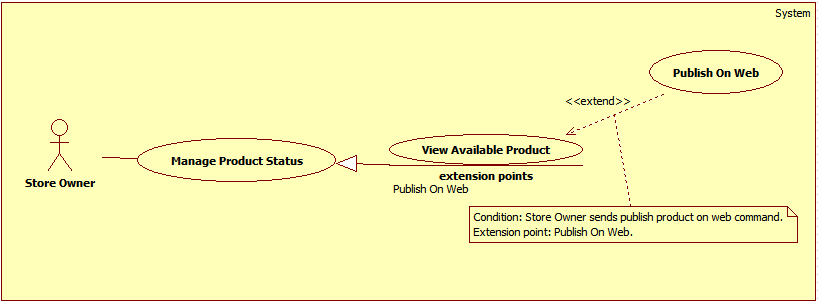


Figure 21: <Store Owner> View Available Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS016** | | | |
| **Use Case No.** | **HPS016** | Use Case Version | 2.0 |
| **Use Case Name** | View Available Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to view products which was received from consignor.   **Goal**:   * Successfully view available products so they can publish them on web.   **Triggers**:   * Store owner sends view available product command.   **Preconditions**:   * Guest is logged in as Store Owner.   **Post Conditions**:   * **Success**: Available products is display to store owner. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view available product command. | System loads and displays information of available products in table with following information:  + STT : label.  + Tên sản phẩm: label.  + Mã kí gửi: label.  + Chi tiết : send command to view detail.  [Exception 1] |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Error occurs during get available product information phase. | System shows error. |   **Relationships**: N/A  **Business Rules**:   * Only product in “In Inventory” status can be displayed in this table. * List are sorted action date in chronological order. The most recent received product will appears first in the list, while the oldest one appears last. * System will display the first 10 available products. * System will display available product which belongs to this store owner. | | | |

##### <Store Owner> View Available Product

**Use Case diagram**

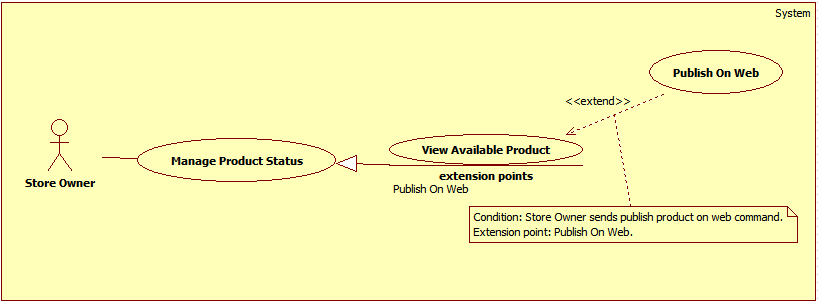


Figure 22: <Store Owner> Publish On Web

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS017** | | | |
| **Use Case No.** | **HPS017** | 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 on “Home page”.   **Triggers**:   * When store owner publish product on web command.   **Preconditions**:   * Guest is logged in as Store Owner and is on view available product detail page.   **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 textarea. * 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 2,3,4]  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 sends cancel command. | System redirects to view available 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 by View Available Product (Send view detail command).  **Business Rules**:   * Only product in “In Inventory” status can be published on web. * System will change product status on database base on product id. * 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. | | | |

##### <Store Owner> View Sold Product

**Use Case diagram**



Figure 23: <Store Owner> View Sold Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS018** | | | |
| **Use Case No.** | **HPS018** | Use Case Version | 2.0 |
| **Use Case Name** | View Sold Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to view products which was.   **Goal**:   * Successfully view sold products so they can pay consignor.   **Triggers**:   * Store owner sends view sold product command.   **Preconditions**:   * Guest is logged in as Store Owner.   **Post Conditions**:   * **Success**: Sold products is display to store owner. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view available product command. | System loads and displays information of available products in table with following information:  + STT : label.  + Tên sản phẩm: label.  + Mã kí gửi: label.  + Chi tiết : send command to view detail.  [Exception 1] |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Error occurs during get available product information phase. | System shows error. |   **Relationships**: N/A.  **Business Rules**:   * Only product in “Sold” status can be displayed in this table. * List are sorted action date in chronological order. The most recent received product will appears first in the list, while the oldest one appears last. * System will display the first 10 available products.   System will display available product which belongs to this store owner. | | | |

##### <Store Owner> Process Payment

**Use Case diagram**



Figure 24: <Store Owner> Process Payment

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS019** | | | |
| **Use Case No.** | **HPS019** | Use Case Version | 2.0 |
| **Use Case Name** | Process Payment | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | Normal |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner pay consignor via payment service.   **Goal**:   * Pay consignor with online payment.   **Triggers**:   * When store owner send pay consignor via payment service command.   **Preconditions**:   * Consignor has provided online payment account and guest is logged in as store owner and is on view sold product detail pop-up.   **Post Conditions**:   * **Success**: Change product and consignment status to “completed”. Completed return money to consignor. * **Fail**: Show error message. No information has been changed.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends pay consignor via payment service. | 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. * Amout : number, hidden.   System redirect to payment service page. | | 2 | Store owner processes payment on payment service page.  Store owner send return to HPS website | System retrieves return value of payment process. [Exception 1]  System updates database.  System redirects to view sold product page. |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Error occurs during pay consignor phase. | System cancel payment, redirect to view sold product page. |   **Relationships**: Included from Pay Consignor.  **Business Rules**:   * Store owner can not 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 or email to consignor to notify them about the payment. | | | |

##### <Store Owner> View Canceled Product

**Use Case diagram**

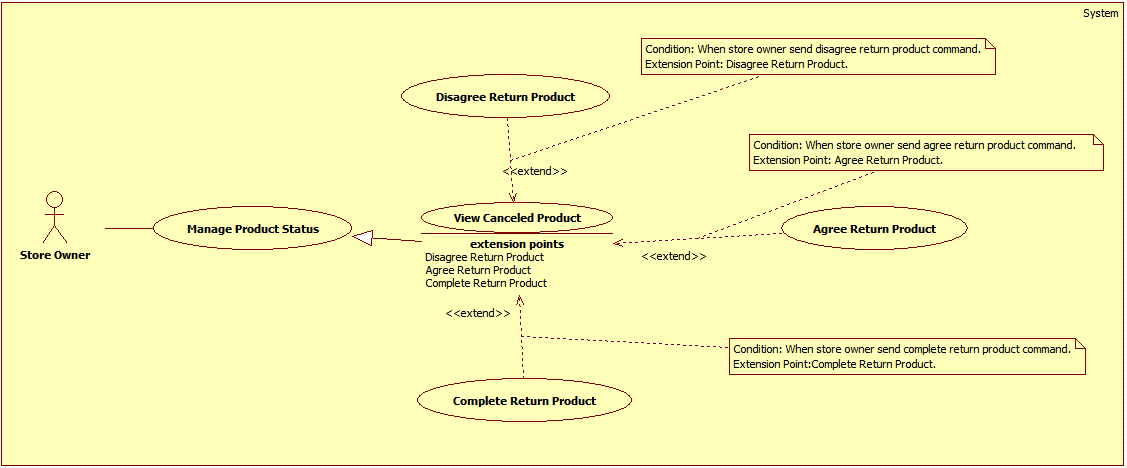


Figure 25: <Store Owner> View Canceled Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS020** | | | |
| **Use Case No.** | **HPS020** | Use Case Version | 2.0 |
| **Use Case Name** | View Canceled Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to view products which was requested cancel from consignor.   **Goal**:   * Successfully view canceled products.   **Triggers**:   * Store owner sends view canceled product command.   **Preconditions**:   * Guest is logged in as Store Owner.   **Post Conditions**:   * **Success**: Cancled products is display to store owner. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view available product command. | System loads and displays information of canceled products in table with following information:  + STT : label.  + Tên sản phẩm: label.  + Mã kí gửi: label.  + Trạng thái: label.  + Chi tiết : send command to view detail.  [Exception 1] |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Error occurs during get available product information phase. | System shows error. |   **Relationships**: N/A  **Business Rules**:   * Only product in “Canceled” or “NotYetReceive” status can be displayed in this table. * List are sorted action date in chronological order. The most recent received product will appears first in the list, while the oldest one appears last. * System will display the first 10 available products. * System will display available product which belongs to this store owner. * If store owner agrees consignor cancel request, “Trạng thái” will be “Chờ nhận hàng”. - If store owner not yet handle cancel request, “Trạng thái” will be “Chờ duyệt”. | | | |

##### <Store Owner> Disagree Return Product

**Use Case diagram**

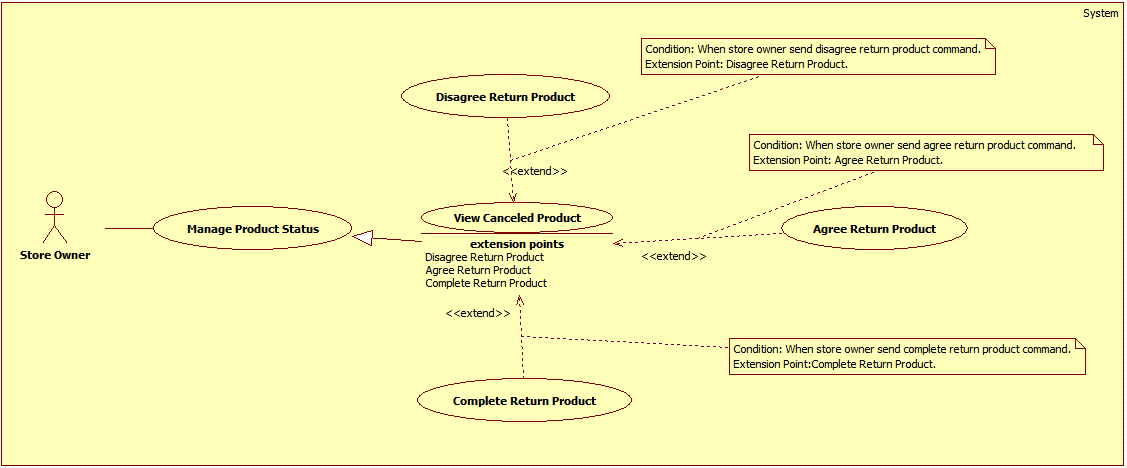


Figure 26: <Store Owner> Disagree Return Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS021** | | | |
| **Use Case No.** | **HPS021** | Use Case Version | 2.0 |
| **Use Case Name** | Disagree Return Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to refuse cancel product request from consignor.   **Goal**:   * Refuse consignor cancel product request and notifies them about the cancellation.   **Triggers**:   * When store owner send disagree return product.   **Preconditions**:   * Users has logged in as “Store Owner” role and is in view canceled product page.   **Post Conditions**:   * **Success**: Product status is changed to “OnWeb”, and can be found by search funtion. * **Fail**: Show error message. No information has been changed.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send view canceled product detail command. | System load product’s information into pop-up with following information:   * Họ tên : label. * Địa chỉ : label. * Số điện thoại : label. * Email : label. * Tên sản phẩm : label. * Mã hàng kí gửi: label. * Giá kí gửi : label. * Ngày nhận hàng : label. * Ngày đăng kí hủy kí gửi : label. | | 2 | Store owner send disagree return product command.  [Alternative 1] | System update database.  [Exception 1]  System shows success message.  System redirects to view canceled product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send cancel command. | System redirects to view canceled product page. No information is changed. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Errors occurs during update database phase. | System displays error.  System redirects to view canceled product page. |   **Relationships**: Extended to View canceled product.  **Business Rules**:   * Product status will be changed to “OnWeb”. * If consignor has both phone and email information. System will send SMS and email to consignor. If not system will send SMS to consignor to notifies about the cancellation. | | | |

##### <Store Owner> Complete Return Product

**Use Case diagram**

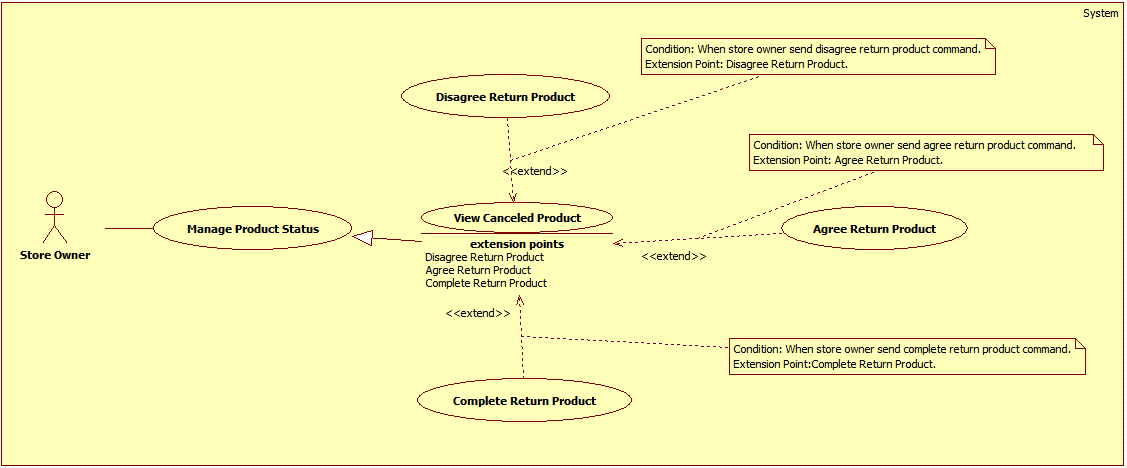


Figure 27: Complete Return Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS022** | | | |
| **Use Case No.** | **HPS022** | Use Case Version | 2.0 |
| **Use Case Name** | Complete Return Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to return canceled product to consignor.   **Goal**:   * Return product and complete consignment.   **Triggers**:   * When store owner send complete return product.   **Preconditions**:   * Users has logged in as “Store Owner” role and is in view canceled product page.   **Post Conditions**:   * **Success**: Product status is changed to “completed”, and update this consignment to completed. * **Fail**: Show error message. No information has been changed.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send view canceled product detail command. | System load product’s information into pop-up with following information:   * Họ tên : label. * Địa chỉ : label. * Số điện thoại : label. * Email : label. * Tên sản phẩm : label. * Mã hàng kí gửi: label. * Giá kí gửi : label. * Ngày nhận hàng : label. * Ngày đăng kí hủy kí gửi : label. | | 2 | Store owner sends complete return product command.  [Alternative 1] | System shows pop-up requires store owner to enter cancel product fee. | | 3 | Store owner enter cancel product fee.  [Alternative 2] | System validates inputted data.  [Exception 1]  System update database.  [Exception 2]  System shows success message.  System redirects to view canceled product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner send cancel command. | System redirects to view canceled product page. No information is changed. | | 2 | Store owner send cancel command. | System redirects to view canceled product detail pop-up. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Cancel product fee is not valid | System show message “Vui lòng nhập tiền phạt” and requires store owner to re-enter. | | 2 | Errors occurs during update database phase. | System displays error.  System redirects to view canceled product page. |   **Relationships**: Extended to View canceled product.  **Business Rules**:   * Product status will be changed to “completed”. * If consignor has both phone and email information. System will send SMS and email to consignor. If not system will send SMS to consignor to notifies about the cancellation. | | | |

##### <Store Owner> View Ordered Product

**Use Case diagram**

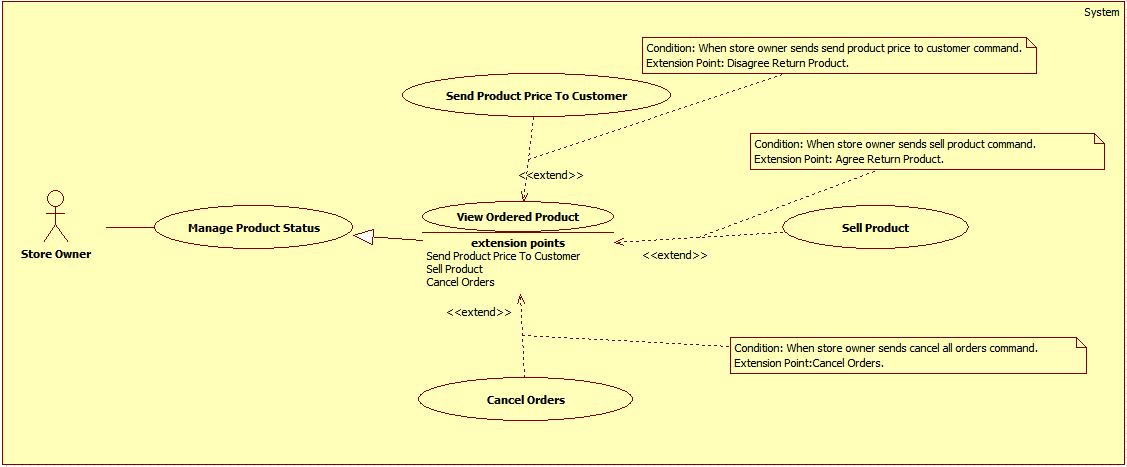


Figure 28: <Store Owner> View Ordered Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS023** | | | |
| **Use Case No.** | **HPS023** | Use Case Version | 2.0 |
| **Use Case Name** | View Ordered Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to view products which was ordered by customer.   **Goal**:   * Successfully view ordered products.   **Triggers**:   * Store owner sends view ordered product command.   **Preconditions**:   * Guest is logged in as Store Owner.   **Post Conditions**:   * **Success**: Ordered products is display to store owner. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view ordered product command. | System loads and displays information of ordered products in table with following information:  + STT : label.  + Tên sản phẩm: label.  + Mã kí gửi: label.  + Số người đặt: label  + Chi tiết : send command to view detail.  [Exception 1] |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Error occurs during get ordered product information phase. | System shows error. |   **Relationships**: N/A  **Business Rules**:   * Only product in “Ordered” status can be displayed in this table. * List are sorted action date in chronological order. The most recent received product will appears first in the list, while the oldest one appears last. * System will display the first 10 available products. * System will display available product which belongs to this store owner. | | | |

##### <Store Owner> Send Product Price To Customer

**Use Case diagram**

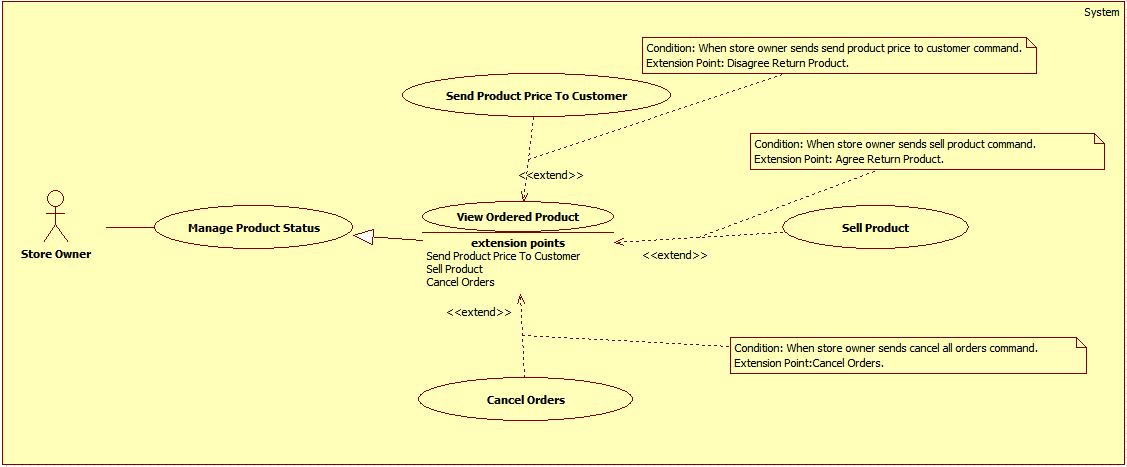


Figure 29: <Store Owner> Send Product Price To Customer

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS024** | | | |
| **Use Case No.** | **HPS024** | Use Case Version | 2.0 |
| **Use Case Name** | Send Product Price To Customer | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | Normal |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to send product price to customer who ordered this product.   **Goal**:   * Successfully send product information to customer.   **Triggers**:   * Store owner sends send product price to customer command.   **Preconditions**:   * Guest is logged in as Store Owner. Product has been ordered by at least 1 customer, and product “IsSpecial” is 1.   **Post Conditions**:   * **Success**: Send SMS or email to customer about product price. * **Fail**: N/A.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view ordered product detail command. | System loads and displays information of ordered products in table with following information:   * Giá kí gửi: Label. * Giá gửi khách hàng: Free input text. * Danh sách người đặt mua: table with 5 columns:   + checkbox   + Tên khách hàng : label   + Ngày đặt mua: label   + Thông tin liên hệ : label   + Chi tiết : label | | 2 | Store owner chooses 1 or more customer.  Store enter price which is sent to customer.  Store owner sends send product price to customer command.  [Alternative 1] | System validates inputted data. [Exception 1,2]  System notifies to customer.  System updates database.  [Exception 3]  System shows success message.  System redirects to view ordered product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store owner send cancel command. | System redirects to view ordered product page. No information is changed. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store doesn’t choose any customer or choose more than one customer. | System shows error message “Vui lòng chỉ chọn một khách hàng”. | | 2 | Selling price is not number or less than 0 or is empty. | System shows error message “Vui lòng nhập giá bán”. | | 3 | Error occurs during process sell product price phase. | System shows error. |   **Relationships**: Extended to View Ordered Product (send Sell Product command).  **Business Rules**:   * Other orders will be cancel and system will send nofitication to customer who has ordered this product. * System will notify consignor that his/her product has been sold. * Product status will be changed to “Sold”. | | | |

##### <Store Owner> Cancel Orders

**Use Case diagram**

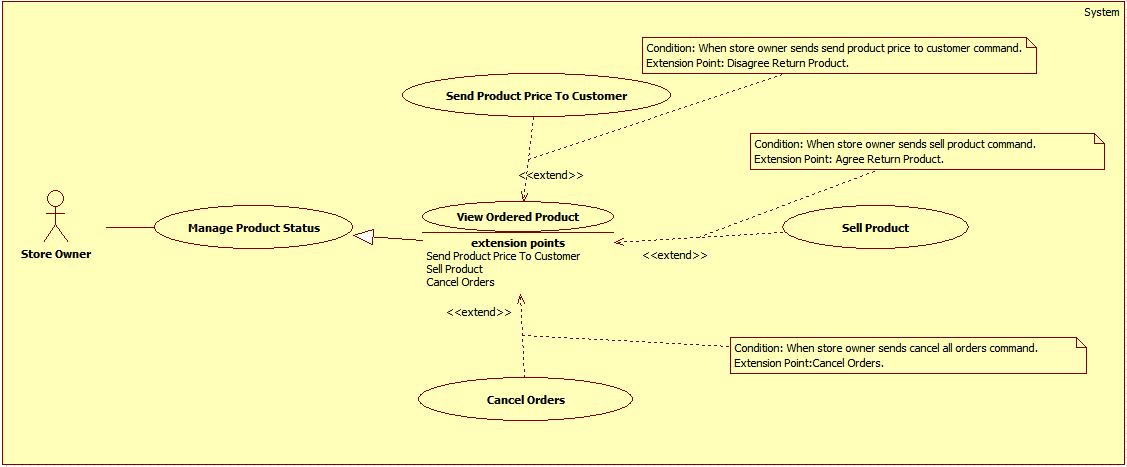


Figure 30: <Store Owner> Send Product Price To Customer

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS025** | | | |
| **Use Case No.** | **HPS025** | Use Case Version | 2.0 |
| **Use Case Name** | Send Product Price To Customer | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | Normal |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to cancel all orders which relates to this product.   **Goal**:   * Successfully cancel all orders.   **Triggers**:   * Store owner sends cancel all orders command.   **Preconditions**:   * Guest is logged in as Store Owner. Product has been ordered by at least 1 customer.   **Post Conditions**:   * **Success**: Send SMS or email to customer about order cancellation. * **Fail**: N/A.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view ordered product detail command. | System loads and displays information of ordered products in table with following information:   * Giá kí gửi: Label. * Giá gửi khách hàng: Free input text. * Danh sách người đặt mua: table with 5 columns:   + checkbox   + Tên khách hàng : label   + Ngày đặt mua: label   + Thông tin liên hệ : label   + Chi tiết : label | | 2 | Store owner sends cancel all orders command.  [Alternative 1] | System updates database.  System notifies to customer.  [Exception 1]  System shows success message.  System redirects to view ordered product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store owner send cancel command. | System redirects to view ordered product page. No information is changed. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Error occurs during process cancel orders product phase. | System shows error. |   **Relationships**: Extended to View Ordered Product (send Sell Product command).  **Business Rules**:   * Other orders will be cancel and system will send nofitication to customer who has ordered this product. * System will notify consignor that his/her product has been sold. * Product status will be changed to “Sold”. | | | |

##### <Store Owner> Extend Consignment Period

**Use Case diagram**

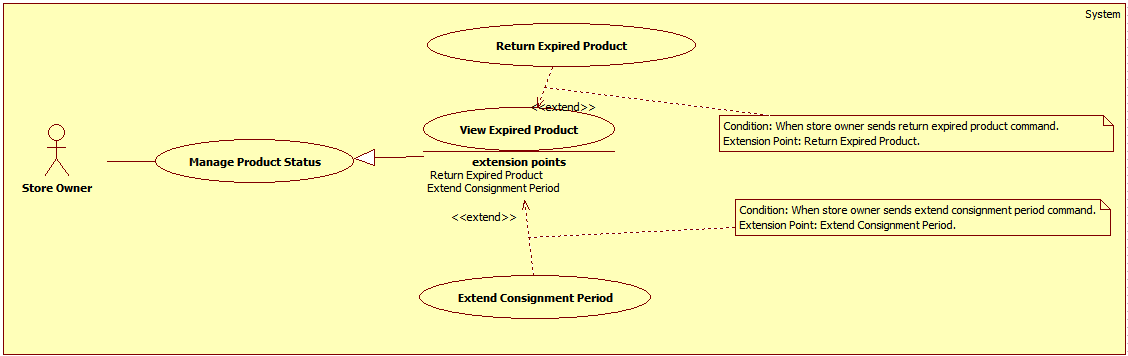


Figure 31: <Store Owner> Extend Consignment Period

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS026** | | | |
| **Use Case No.** | **HPS026** | Use Case Version | 2.0 |
| **Use Case Name** | Extend Consignment Period | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to extend consignment period which was expired.   **Goal**:   * Successfully view extend consignment period. Product is displayed on web.   **Triggers**:   * Store owner sends extend consignment period command.   **Preconditions**:   * Guest is logged in as Store Owner.   **Post Conditions**:   * **Success**: Product status is changed to “OnWeb”. * **Fail**: N/A.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view expired detail product command. | System loads information of expired products into pop-up with following information:   * Họ tên:   + Label. * Địa chỉ:   + Label. * Số điện thoại:   + Label. * Email:   + Label. * Tên sản phẩm:   + Label. * Mã hàng kí gửi:   + Label. * Ngày kí gửi:   + Label. * Giá thỏa thuận:   + Label. * Số ngày quá hạn:   + Label. | | 2 | Store owner sends extend consignment period command.  [Alternative 1] | System show suggest stored fee.  System asks store owner to confirm about the extension. | | 3 | Store owner enter stored fee and send complete command. | System updates database.  [Exception 1]  System shows success message.  System redirects to view expired product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Store owner send cancel command. | System redirects to view expired product page. No information is changed. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Error occurs during get expired product information phase. | System shows error. |   **Relationships**: Extended to View Expired Product (send extend consignment period command).  **Business Rules**:   * After extend consignment period, this consignment will be expired after 30 days later. * If consignor is late to extend consignment period, they have to pay stored fee. * Stored fee is suggest by system by this formula :   + If negotiated price >= 1.000.000 vnd. Stored fee = 10.000 x expiredDays (vnd).  + If negotiated price < 1.000.000 vnd. Stored fee = 5.000 x expiredDays (vnd). | | | |

##### <Store Owner> View On Web Product

**Use Case diagram**

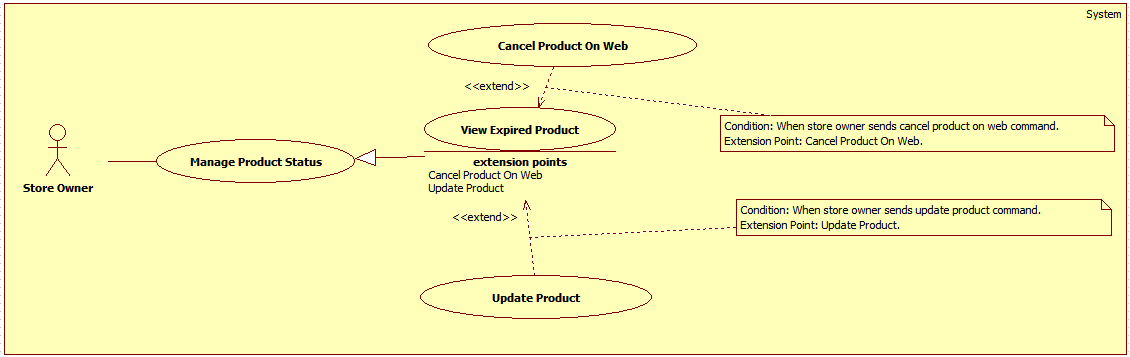


Figure 32: <Store Owner> View On Web Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS027** | | | |
| **Use Case No.** | **HPS027** | Use Case Version | 2.0 |
| **Use Case Name** | View On Web Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to view products which was received from consignor.   **Goal**:   * Successfully view available products so they can publish them on web.   **Triggers**:   * Store owner sends view available product command.   **Preconditions**:   * Guest is logged in as Store Owner.   **Post Conditions**:   * **Success**: Available products is display to store owner. * **Fail**: Show error message.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view available product command. | System loads and displays information of available products in table with following information:  + STT : label.  + Tên sản phẩm: label.  + Mã kí gửi: label.  + Chi tiết : send command to view detail.  [Exception 1] |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Error occurs during get available product information phase. | System shows error. |   **Relationships**: N/A  **Business Rules**:   * Only product in “On Web” status can be displayed in this table. * List are sorted action date in chronological order. The most recent received product will appears first in the list, while the oldest one appears last. * System will display the first 10 available products. * System will display available product which belongs to this store owner. | | | |

##### <Store Owner> Update Product

**Use Case diagram**



Figure 33: <Store Owner> Update Product

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS028** | | | |
| **Use Case No.** | **HPS028** | Use Case Version | 2.0 |
| **Use Case Name** | Update Product | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to update product information after publish product.   **Goal**:   * Change product information bases on information that was given by store owner.   **Triggers**:   * When store owner send update product command.   **Preconditions**:   * Product status must be “On Web”.   **Post Conditions**:   * **Success**: Product information is changed. * **Fail**: Product information remains like the beginning. No information has been changed   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view product detail command. | System loads and displays product information 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. * Ngày mua hàng:   + Label.   + Not editable. * 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 textarea. * Ảnh sản phẩm:   + Image. | | 2 | Store owner enters product’s information to fields. | System validates inputted values. [Exception 1,2,3] | | 3 | Store owner sends update command.  [Alternative 1]. | System changes product information.  [Exception 4].  System redirects to view on web product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends cancel command | System forwards to view on web 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”. | | 4 | Error occurs during change product information phase. | System displays error detail. System forwards to view on web product page. |   **Relationships**: Extended to View On Web Product (send update product command).  **Business Rules**:   * System will change product information bases on the information which is given by store owner. * 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. * Only product has “On Web” status can be update. | | | |

##### <Store Owner> Cancel Product On Web

**Use Case diagram**



Figure 34: <Store Owner> Cancel Product On Web

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS029** | | | |
| **Use Case No.** | **HPS029** | Use Case Version | 2.0 |
| **Use Case Name** | Cancel Product On Web | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | High |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to cancel product when consignor come to store and ask to take back product.   **Goal**:   * Change product and consignment status to complete. Sucessfully return product to consignor.   **Triggers**:   * When store owner send cancel product on web command.   **Preconditions**:   * Product status must be “On Web”.   **Post Conditions**:   * **Success**: Product information is changed. * **Fail**: Product information remains like the beginning. No information has been changed   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view product detail command. | System loads and displays product information 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. * Ngày mua hàng:   + Label.   + Not editable. * 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 textarea. * Ảnh sản phẩm:   + Image. | | 2 | Store owner sends cancel product on web command.  [Alternative 1] | System shows pop-up requires store owner to enter cancel product fee. | | 3 | Store owner enter cancel product fee.  [Alternative 2] | System validates inputted data.  [Exception 1]  System update database.  [Exception 2]  System shows success message.  System redirects to view on web product page. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends cancel command | System forwards to view on web product page. No information is recorded. |   **Exceptions**: N/A  **Relationships**: Extended to View On Web Product (send cancel product on web command).  **Business Rules**: | | | |

##### <Store Owner> View Consignment Statistic

**Use Case diagram**

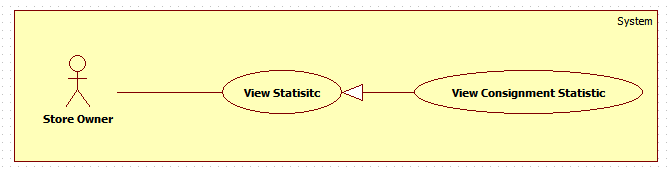
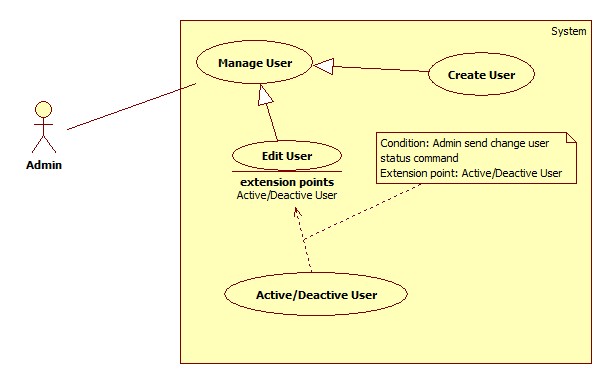


Figure 35: <Store Owner> View Consignment Statistic

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – HPS030** | | | |
| **Use Case No.** | **HPS030** | Use Case Version | 2.0 |
| **Use Case Name** | View Consignment Statistic | | |
| **Author** | Quan Tien Dan | | |
| **Date** | May 26, 2015 | Priority | Normal |
| **Actor**:   * Store Owner   **Summary**:   * This use case allows store owner to view consignment statistic.   **Goal**:   * Give store owner the overall view of the works in their store.   **Triggers**:   * When store owner send view consignment statistic.   **Preconditions**:   * Users has logged in as “Store Owner” role and is in statistic page.   **Post Conditions**:   * **Success**: View all consignment status of the store. * **Fail**: N/A   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner sends view consignment statistic command. | System loads information about the consignments of their store. [Exception 1] | | 2 | Store owner sends view detail of consignment statistic command. | System loads detail information about the consignment with following format.  [Exception 2]   * Tên sản phẩm:   + Label. * Loại sản phẩm:   + Label. * Hãng:   + Label. * Người kí gửi:   + Label * Điện thoại:   + Label. * Email:   + Label. * Ngày tạo ký gửi:   + Label. * Ngày từ chối sản phẩm:   + Label. * Ngày nhận sản phẩm:   + Label. * Ngày trả tiền khách hàng:   + Label. * Ngày hủy kí gửi:   + Label. * Ngày bán sản phẩm:   + Label. * Ngày duyệt yêu cầu:   + Label. * Ngày từ chối yêu cầu:   + Label. * Ngày đăng lên web:   + Label. * Ngày hoàn trả sản phẩm:   + Label. * Tiền trả khách hàng:   + Label. |   **Alternative Scenario**: N/A  **Exceptions**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Store owner confirm error and send return to management page command. | Error occurs during get detail of consignment statistic information phase. System display error detail.  System forwards to “Management Page”. | | 2 | Store owner confirm error and send return to management page command. | Error occurs during get consignment statistic information phase. System display error detail.  System forwards to “Management Page”. |   **Relationships**: N/A  **Business Rules**:   * Consignment is displayed 10 items per page. * List are sorted created date in chronological order. The most recent appears last in the list, while the oldest one appears first. * Store owner can view information of other consignment status by change the value of “Trạng thái” drop down list. * Store owner can view consignment status information of the other days by change the value of “Từ ngày” and “Đến ngày” datetime picker. * When store owner clicks on see detail, depends on consignment status, and product status system will display detail information differently:  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Ngày duyệt yêu cầu | Ngày từ chối yêu cầu | Ngày từ chối sản phẩm | Ngày nhận sản phẩm | Ngày trả tiền khách hàng | Ngày hủy kí gửi | Ngày bán sản phẩm | Ngày đăng lên web | Ngày hoàn trả sản phẩm | Tiền trả khách hàng | | ProductStatus : NotAvailable  ConsignmentStatus : Waiting | x |  |  |  |  |  |  |  |  |  | | ProductStatus : Available  ConsignmentStatus : Accepted | x |  |  | x |  |  |  |  |  |  | | ProductStatus : OnWeb  ConsignmentStatus : Accepted | x |  |  | x |  |  |  | x |  |  | | ProductStatus : Sold  ConsignmentStatus : Accepted | x |  |  | x |  |  | x | x |  |  | | ProductStatus : NotAvailable  ConsignmentStatus:Completed | x |  |  | x | x |  | x | x |  | x | | ProductStatus : NotAvailable  ConsignmentStatus : Refuse | x | x | x |  |  |  |  |  |  |  | | ProductStatus : Cancel  ConsignmentStatus : Cancel | x |  |  | x |  | x |  | x | x |  | | | | |

#### <Admin> Overview Use Case

****

##### < Admin > Create User Use case

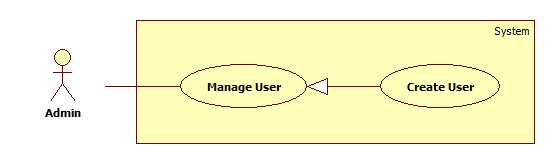


Figure 36: < Admin > Create User Use case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS031 | | | |
| **Use Case No.** | HPS031 | Use Case Version | 2.0 |
| **Use Case Name** | Create User | | |
| **Author** | Nguyen Huy Hoang | | |
| **Date** | August 7, 2015 | Priority | High |
| **Actor**:   * Admin   **Summary**:   * This use case allows admin add new user.   **Goal**:   * New user is added to database.   **Triggers**:   * Admin send “add new user” command.   **Preconditions**:   * Guest must login with role Admin.   **Post Conditions**:   * Success: New user is inserted to database. * Fail: Show an error message, nothing is changed in database.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin send “add new user” command. | System pop-up for admin input include the following information:   * “Tên Đăng Nhập”: free text input, min length: 6, max length: 50, required. * “Mật Khẩu”: free text input, min length: 8, max length: 20, required. * “Điện Thoại: free text input, format: number, * “Email”: free text input, regular expression: “^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$". * “Tên”: free text input. Max length : 50. * “Đia Chỉ”: free text input. * “Tài Khoản Paypal”: free text input. * “Quyền Hạn”: select option include: “Chủ Cửa Hàng” and “Thành Viên”, required. * A command to finish action. | | 2 | Admin input information and send command to finish add new user.  [Alternative 1] | [Exception 1,2]  Add new user to database and show list new user. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Admin dismiss pop-up. | Nothing to change. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User not fills all required field. | Show message: “Vui lòng nhập thông tin vào nhưng ô bắt buộc.” | | 2 | User invalid format | Show message: “Vui lòng nhập đúng thông tin”. |   **Relationships**: N/A  **Business Rules**:   * Status of new user is set to “active”. | | | |

##### < Admin > Edit User Use case

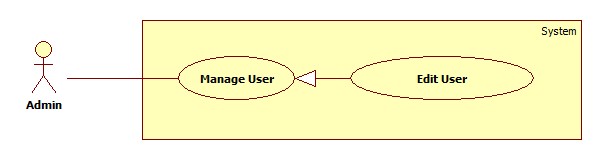


Figure 37: < Admin > Edit User Use case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS032 | | | |
| **Use Case No.** | HPS032 | Use Case Version | 2.0 |
| **Use Case Name** | Edit User | | |
| **Author** | Nguyen Huy Hoang | | |
| **Date** | August 7, 2015 | Priority | High |
| **Actor**:   * Admin   **Summary**:   * This use case allows admin edit user information.   **Goal**:   * User information is updated to database.   **Triggers**:   * Admin send “edit user” command.   **Preconditions**:   * Guest must login with role Admin.   **Post Conditions**:   * Success: User information is updated to database. * Fail: Show an error message, nothing is changed in database.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin send “edit user” command. | System pop-up for admin input include the following information:   * “Điện Thoại: free text input, format: number, * “Email”: free text input, regular expression: “^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$". * “Tên”: free text input. Max length : 50. * “Đia Chỉ”: free text input. * “Tài Khoản Paypal”: free text input. * A command to finish action. | | 2 | Admin change information and send command to finish edit user.  [Alternative 1] | [Exception 1]  Add new user to database and show list new user. |   **Alternative Scenario**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Admin dismiss pop-up. | Nothing to change. |   **Exceptions**:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User invalid format | Show message: “Vui lòng nhập đúng thông tin”. |   **Relationships**: N/A  **Business Rules**:   * Admin can’t change username and password of user. * Old information of user will be filled. | | | |

##### < Admin > Active/Deactive User Use case

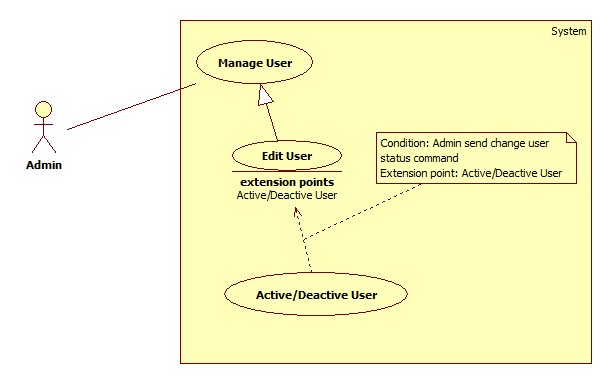


Figure 38: < Admin > Active/Deactive User Use case

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS033 | | | |
| **Use Case No.** | HPS033 | Use Case Version | 2.0 |
| **Use Case Name** | Active/Deactive User | | |
| **Author** | Nguyen Huy Hoang | | |
| **Date** | August 7, 2015 | Priority | High |
| **Actor**:   * Admin   **Summary**:   * This use case allows admin change user status.   **Goal**:   * User status is updated to database.   **Triggers**:   * Admin send “change user status” command.   **Preconditions**:   * Guest must login with role Admin.   **Post Conditions**:   * Success: User status is updated to database. * Fail: Show an error message, nothing is changed in database.   **Main Success Scenario**:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin send “change user status” command. | System show new status of user. |   **Alternative Scenario**: N/A  **Exceptions**: N/A  **Relationships**:   * Extending Use-cases: Extend from Edit User (Admin send “change user status” command).   **Business Rules**:   * If current status of user is Active, change to Deactivate, and if current status of user is Deactivate, change to Active. | | | |

#### <System> Search Product Use Case

**Use Case diagram**



Figure 39: <System> Search Product Online

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** HPS033 | | | |
| **Use Case No.** | **HPS033** | 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. | | | |

## Software System Attribute

### Usability

#### Graphic User Interface

* Texts and labels should be written in Vietnamese.

#### Usability

* Staff should need less than one week of training to use this system

#### Installation

* The system should be easy to deploy. Customer can deploy successfully and learn to configure, maintain the system within one day of training.
* The attached manual guide must be clear. Installation can be done by users themselves.

### Reliability

* Information about order of StoreOwner can be backup and restore easily using import/export function.

### Availability

* User need to have an internet connection to use every functions of the application in the best way.

### Security

* Each role of user has a specific permission to interact with system.

### Maintainability

* The system is divided into separated modules.

### Portability

* The system can be deployed into many type of servers those have Tomcat 8 Server.

### Performance

* The process of manage consignment’s request should be within 1 minute.

## Conceptual Diagram

### Conceptual Diagram

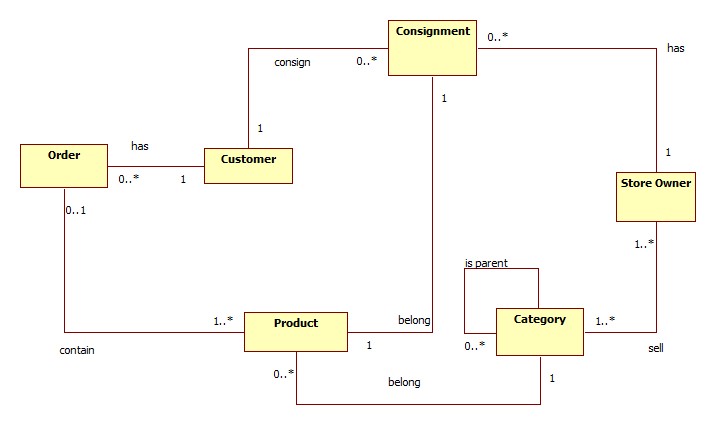


Figure 40: Conceptual Diagram

**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. |

# Report No. 4 Software Design Description

## Design Overview

* This document part describes the details about technical and user interface design of HPS system. The document includes system architectural design, component diagram, detailed description, interaction diagram, user interface design and database design.
* The architectural design gives us the information of how HPS system is organized.
* The component diagram describes the architecture of HPS main component and sub-system.
* The detailed description describes the implement details of HPS system. This part include class diagram, class diagram explanation and interaction diagram.
* The user interface design give us the user graphic interface of HPS system. It describes how does HPS system look like.
* The database design show us how data of HPS system is stored and the relationship between entities in HPS database.

## System Architectural Design

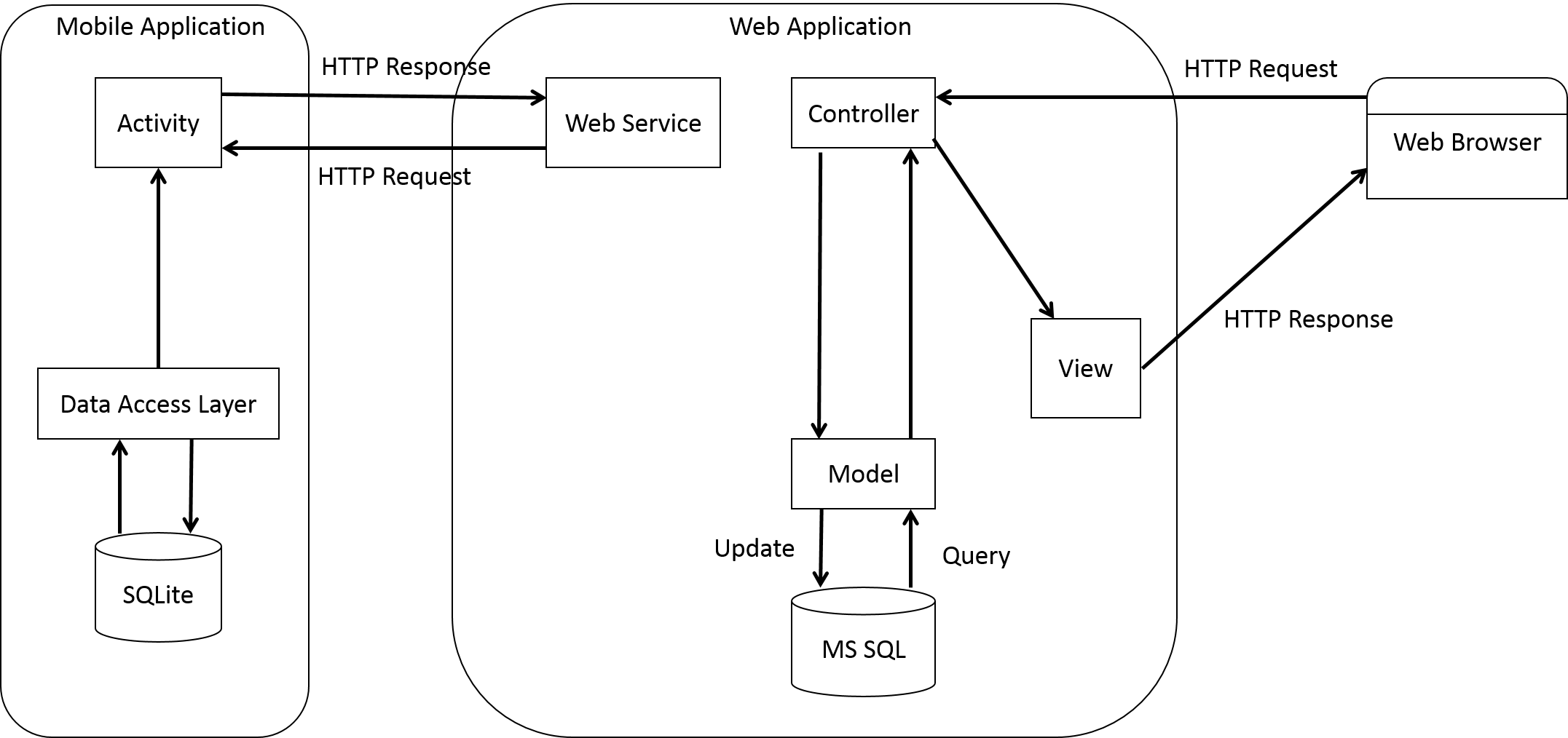


Figure 41: HPS System Architectural

### 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

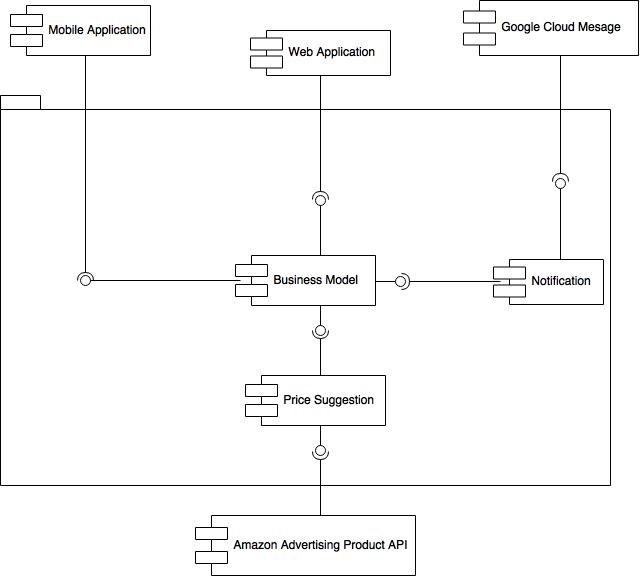


Figure 42: Component Diagram

|  |  |
| --- | --- |
| **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. |
| Price Suggestion | Use to call Amazon Advertising Product API |
| Amazon Advertising Product API | Use to get price suggestion. |

## Detailed Description

### Class Diagram



Figure 43: Class Diagram

### Class Diagram Explanation

#### Consignment

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ConsignmentID | String | Private | Unique identifier of Consignment |
| ProductID | int | Private | ID of Product |
| MemberID | int | Private | ID of Member |
| StoreOwnerID | int | Private | ID of StoreOwner |
| FullName | String | Private | Full name of customer |
| Address | String | Private | Address of customer |
| Phone | String | Private | Phone of customer |
| Email | String | Private | Email of customer |
| PaypalAccount | String | Private | Paypal account of customer |
| FromDate | DateTime | Private | Date Store Owner can receive |
| ToDate | DateTime | Private | The last date Store Owner can receive |
| RaiseWebDate | DateTime | Private | Date publish product on website |
| Period | Integer | Private | Period of consignment |
| MaxPrice | float | Private | Max price suggest to customer |
| MinPrice | float | Private | Min price suggest to customer |
| ReturnedPrice | float | Private | Price return to customer |
| ReviewProductDate | DateTime | Private | Date receive or refuse Product |
| CreatedDate | DateTime | Private | Date create |
| ConsignmentStatusID | int | Private | ID of ConsignmentStatus |
| CancelDate | DateTime | Private | Date Customer request cancel consignment |
| IsExpiredMessage | int | Private | Send message expired consignment |
| ExpiredFee | float | Private | Fee of expired consignment |
| Reason | String | Private | Reason of canceled consignment |
| NegotiatedPrice | float | Private | Negotiated price |
| AppointmentDate | DateTime | Private | Appointment date to receive product |
| ReviewRequestDate | DateTime | Private | Date accept or refuse consignment request |
| CancelFee | float | Private | Fee of cancel consignment |
| AgreeCancelDate | DateTime | Private | Date StoreOwner agree request cancel consignment |
| ReturnDate | DateTime | Private | Date return product to customer |
| ReceivedDate | DateTime | Private | Receive product Date |
| DeliveryMethod | int | Private | Method to delivery product |
| DesirePrice | float | Private | Desire price of customer |

Method

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

#### Product

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ProductID | int | Private | Unique identifier of Product |
| ProductName | String | Private | Name of Product |
| SerialNumber | String | Private | Serial number of Product |
| NewStatus | int | Private | Product new status |
| CategoryID | int | Private | ID of Category |
| Brand | String | Private | Brand of Product |
| Description | String | Private | Description of Product |
| Image | String | Private | Image of Product |
| ProductStatusID | int | Private | ID of ProductStatus |
| SellingPrice | float | Private | Selling Price |
| SellDate | DateTime | Private | Selling Date |
| IsSpecial | int | Private | Is special product |

Method

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

#### Order

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| OrderID | int | Private | Unique identifier of Order |
| CustomerID | int | Private | ID of Customer |
| Date | DateTime | Private | Order date |
| Email | String | Private | Email of customer |
| FullName | String | Private | Full name of customer |
| Address | String | Private | Address of customer |
| Phone | String | Private | Phone of Product |
| TotalPrice | float | Private | Price of Order |
| OrderStatusID | int | Private | ID of OrderStatus |
| ProductID | int | Private | ID of Product |
| SendPrice | int | Private | Price is sent to customer |

Method

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

#### Account

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| AccountID | String | Private | Unique identifier of Account |
| Password | String | Private | Password of user |
| Status | String | Private | Status of account |
| FullName | String | Private | Full name of user |
| Address | String | Private | Address of user |
| Phone | String | Private | Phone of user |
| Email | String | Private | Email of user |
| PaypalAccount | String | Private | Paypal account of user |
| Role | String | Private | Role of user |
| GcmID | String | Private | Google cloud message to receive notification on mobile application. |

Method

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

#### Store Owner

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| StoreOwnerID | int | Private | Unique identifier of StoreOwner |
| AccountID | String | Private | AccountID of StoreOwner |
| Formula | float | Private | Formula to calculate consignment price suggestion |

Method

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

#### Member

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| MemberID | int | Private | Unique identifier of Member |
| AccountID | String | Private | AccountID of Member |

Method

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

#### StoreOwnerCategory

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| StoreOwnerID | int | Private | Unique identifier of StoreOwnerCategory |
| CategoryID | String | Private | Unique identifier of StoreOwnerCategory |

Method

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

#### ConsignmentStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ConsignmentStatusID | int | Private | Unique identifier of ConsignmentStatus |
| ConsignmentStatusName | String | Private | Name of consignment status |

Method

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

#### ProductStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ProductStatusID | int | Private | Unique identifier of ProductStatus |
| ProductStatusName | String | Private | Name of product status |

Method

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

#### OrderStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| OrderStatusID | int | Private | Unique identifier of OrderStatus |
| OrderStatusName | String | Private | Name of order status |

Method

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

#### ProductSeason

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ProductID | int | Private | Unique identifier of ProductSeason |
| SessonID | int | Private | Unique identifier of ProductSeason |

Method

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

#### Season

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| SeasonID | int | Private | Unique identifier of Season |
| SessonName | String | Private | Name of Season |

Method

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

#### Category

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| CategoryID | int | Private | Unique identifier of Category |
| CategoryName | String | Private | Name of Category |
| ParentID | int | Private | ID of parent category |
| EnglishName | String | Private | English name to search product on amazon API |

Method

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

#### Brand

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| BrandID | int | Private | Unique identifier of Brand |
| BrandName | String | Private | Name of Brand |

Method

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| DatingMen | Boolean | Private | Wishes dating person is men or not. |
| DatingWomen | Boolean | Private | Wishes dating person is women or not. |
| DatingAge | String | Private | Age of wishes dating person. |

Method

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

### Interaction Diagram

#### Sequence Diagram

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

##### <Guest> Search Product

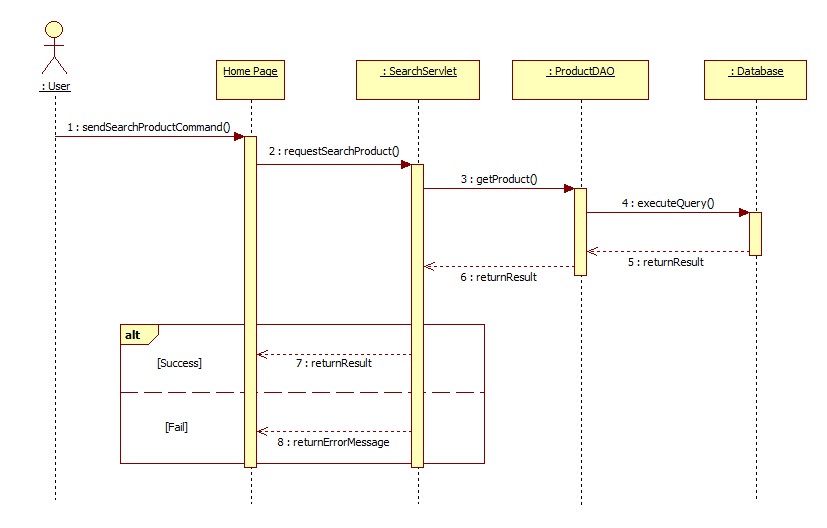


Figure : <Guest> Search Product

##### <Guest> View Detail

**Summary:** This diagram shows how Guest can view product detail.

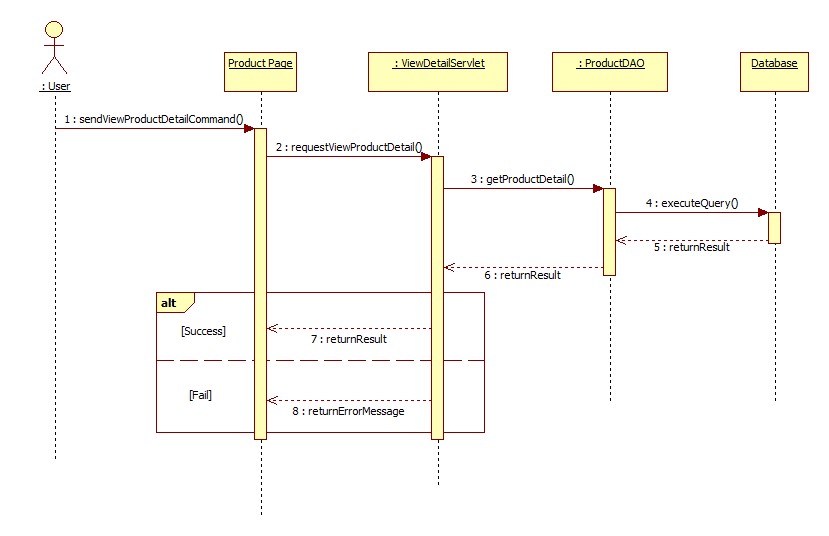


Figure : <Guest> View Detail

##### <Guest> Order Product

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

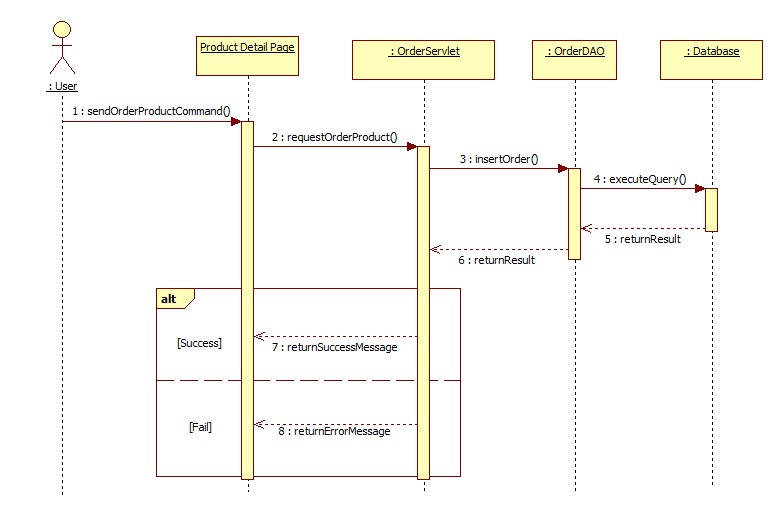
****

Figure : <Guest> Order Product

##### <Store Owner> Publish On Web

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

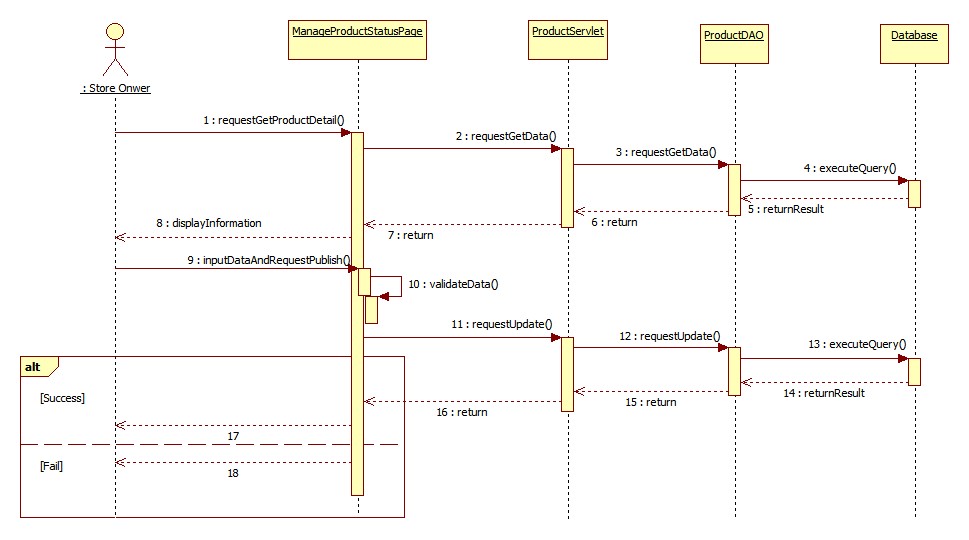


Figure 47: <Store Owner> Publish On Web

##### <Store Owner>Pay Consignor

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

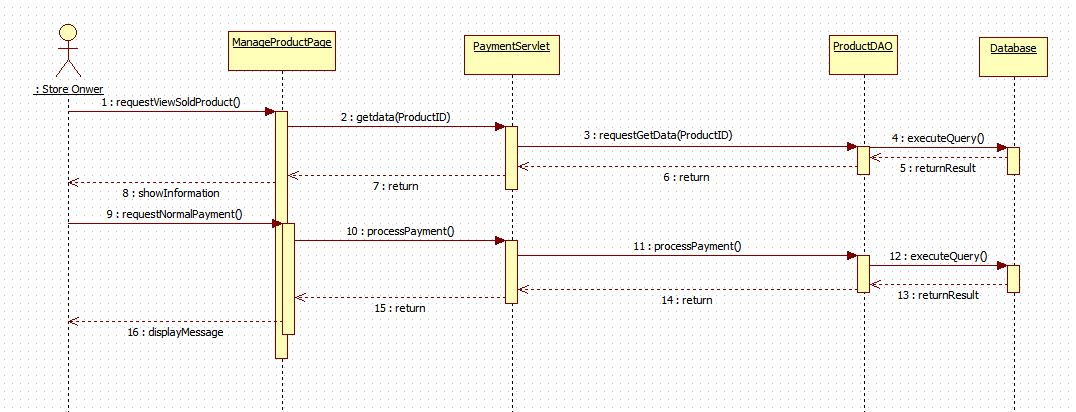


Figure 48: <Store Owner> Pay Consignor

##### <Store Owner> Process Payment

**Summary**: This diagram shows how store owner pay consignor via payment service.

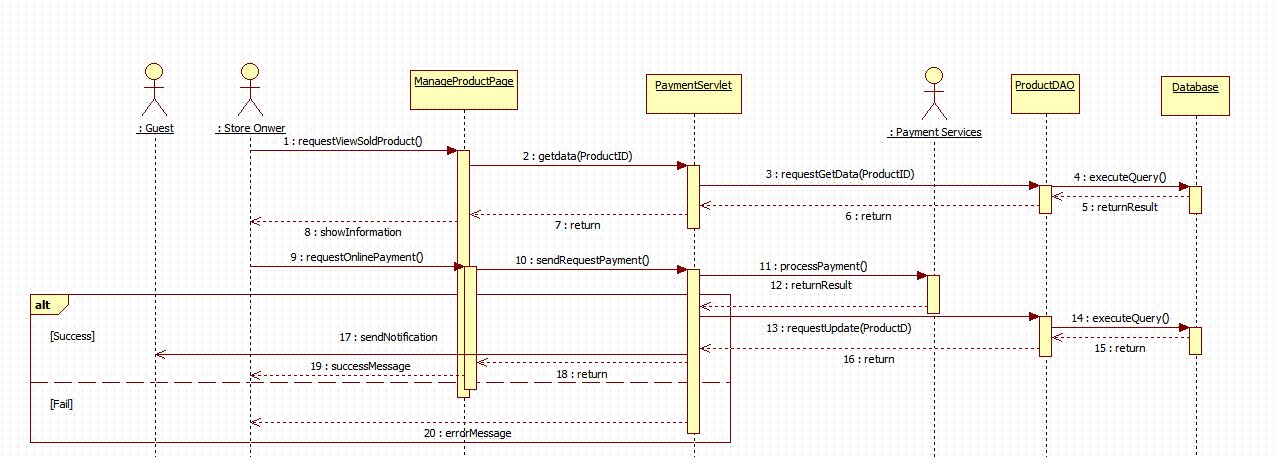


Figure 49: <Store Owner> Process Payment

##### <Store Owner> Agree Return Product

**Summary**: This diagram shows how store owner agree a cancel consignment request.

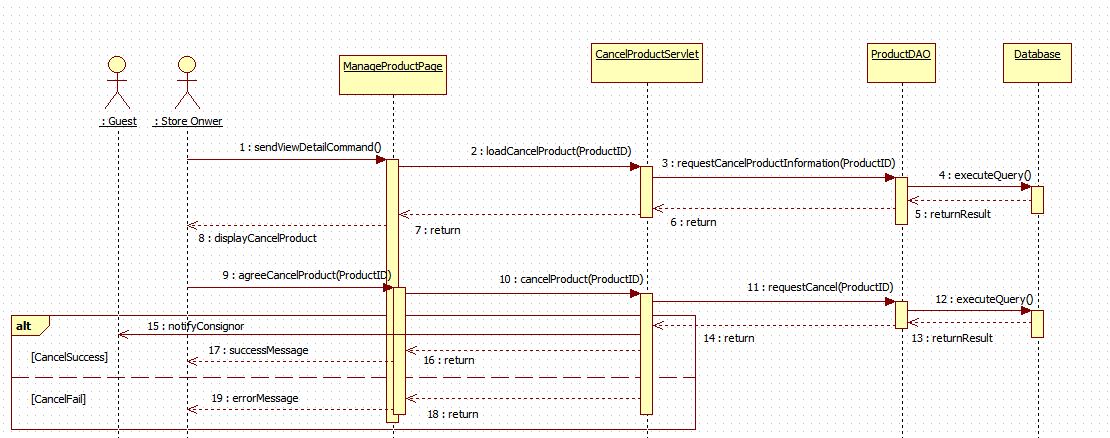


Figure 50: <Store Owner> Agree Return Product

##### <Store Owner> Disagree Return Product

**Summary**: This diagram shows how store owner refuse a cancel consignment request.

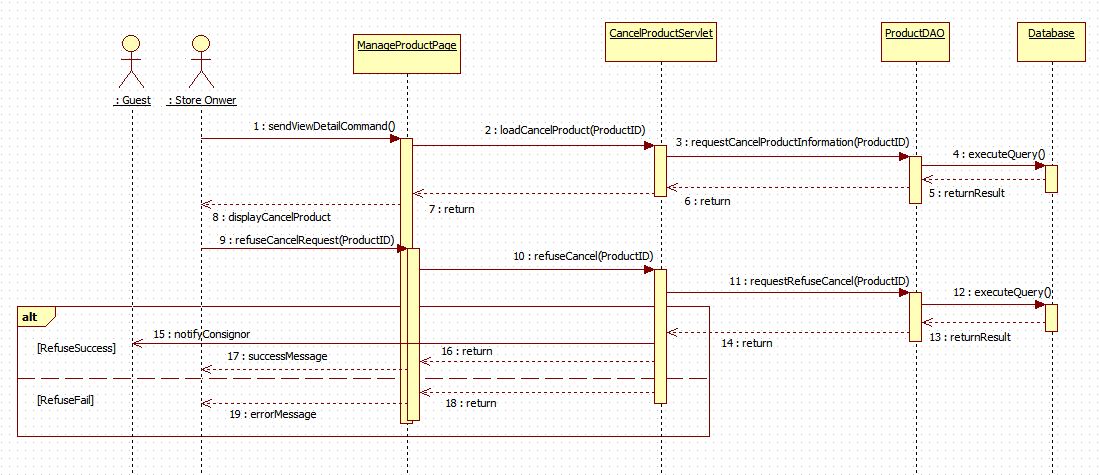


Figure 51: <Store Owner> Disagree Return Product

##### <Store Owner> Sell Product

**Summary**: This diagram shows how store owner sell a product which ordered by one or more customer.

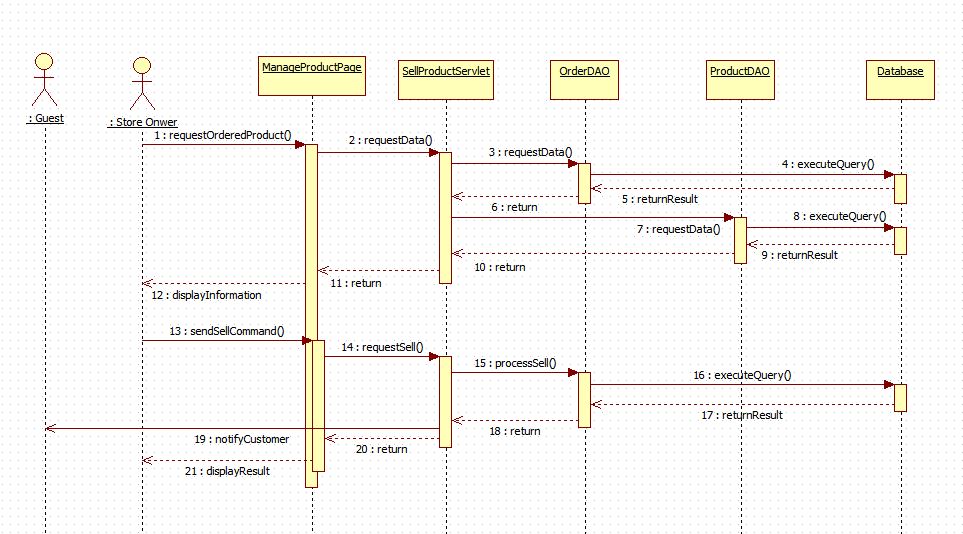


Figure 52: <Store Owner> Sell Product

##### <Store Owner> Send Product Price To Customer

**Summary**: This diagram shows how store owner send product price to customer who has ordered product.

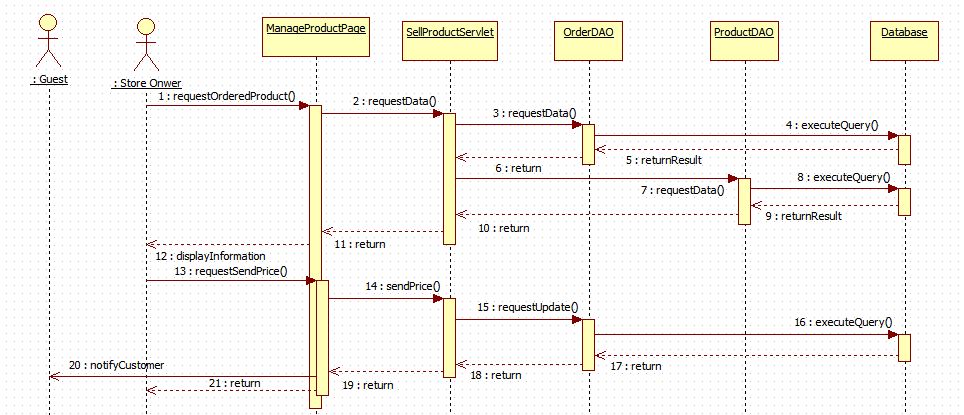


Figure 53: <Store Owner> Send Product Price To Customer

##### <Store Owner> Cancel Orders

**Summary**: This diagram shows how store owner cancel all orders which relates to their product.

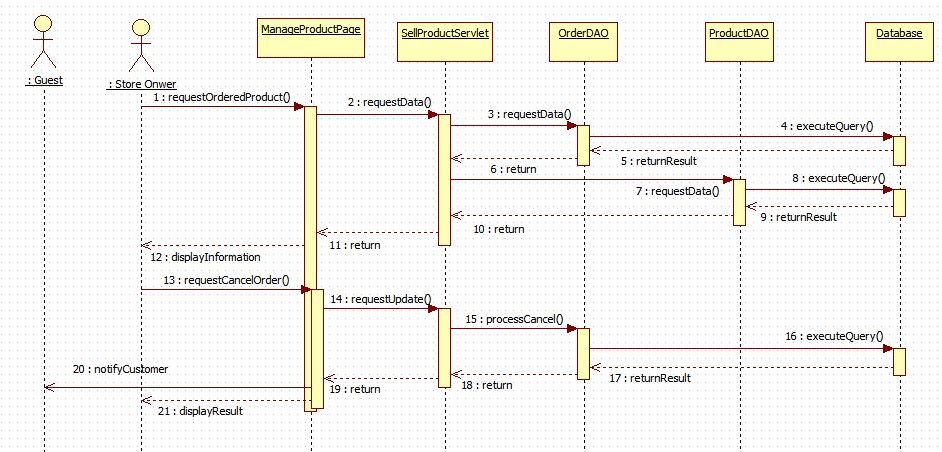


Figure 54: <Store Owner> Cancel Orders

##### <Store Owner> Extend Consignment Period

**Summary**: This diagram shows how store owner help consignor to extend consignment period after it’s over.

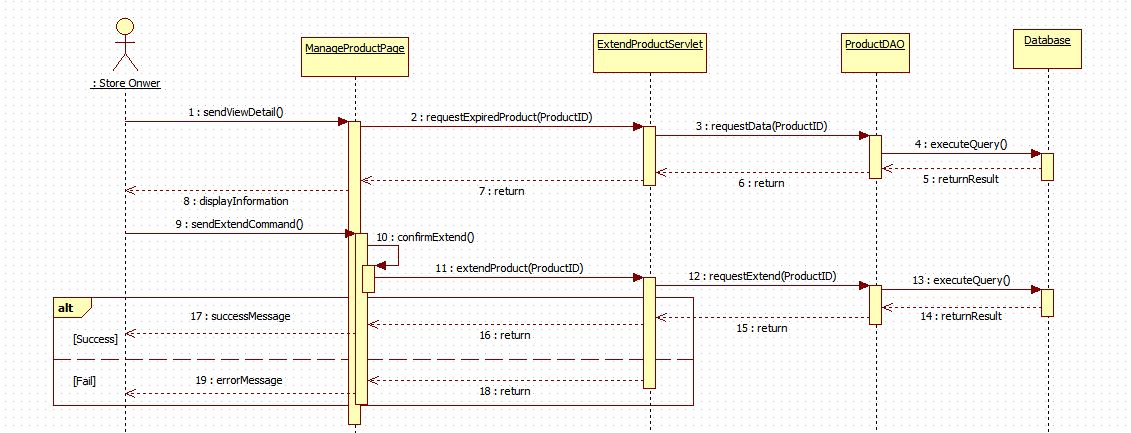


Figure 55: <Store Owner> Extend Consignment Period

##### <Store Owner> Update Product

**Summary**: This diagram shows how store owner update product information after publish product on web.

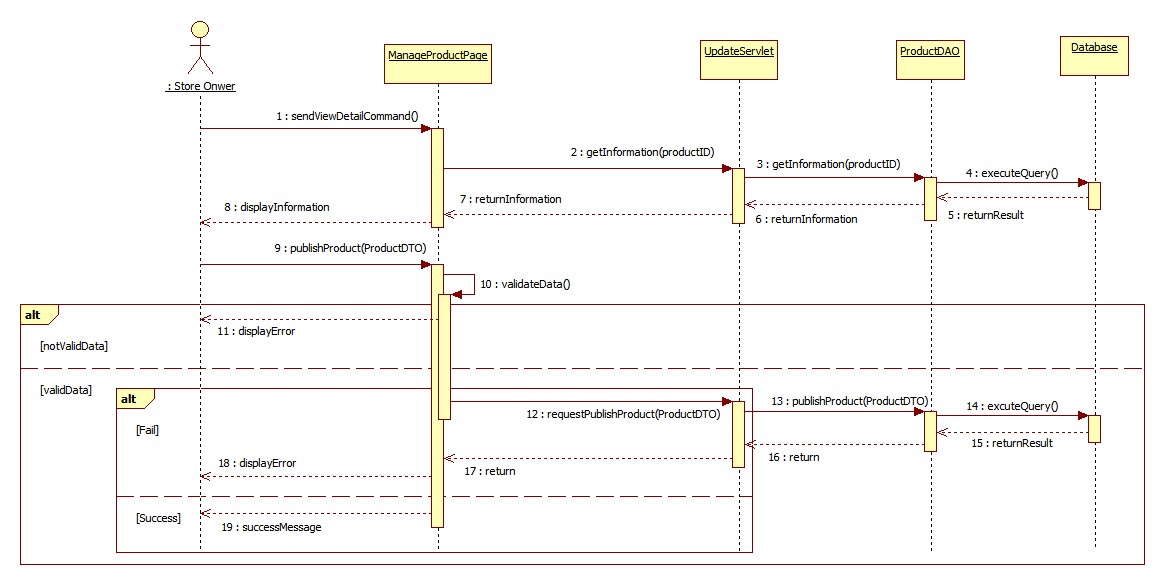


Figure 56: <Store Owner> Update Product

##### <Store Owner> View Available Product

**Summary**: This diagram shows how store owner view received product of their store.

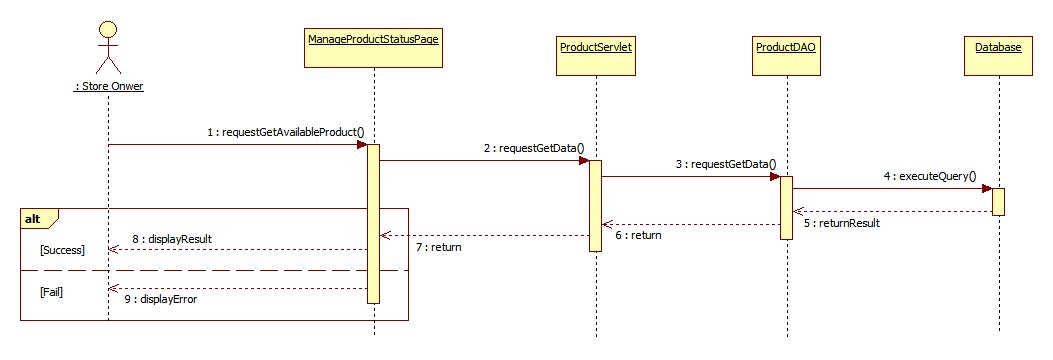


Figure 57: <Store Owner> View Available Product

##### <Store Owner> View Consignment Statistic

**Summary**: This diagram shows how store owner view consignment statistic of their store.

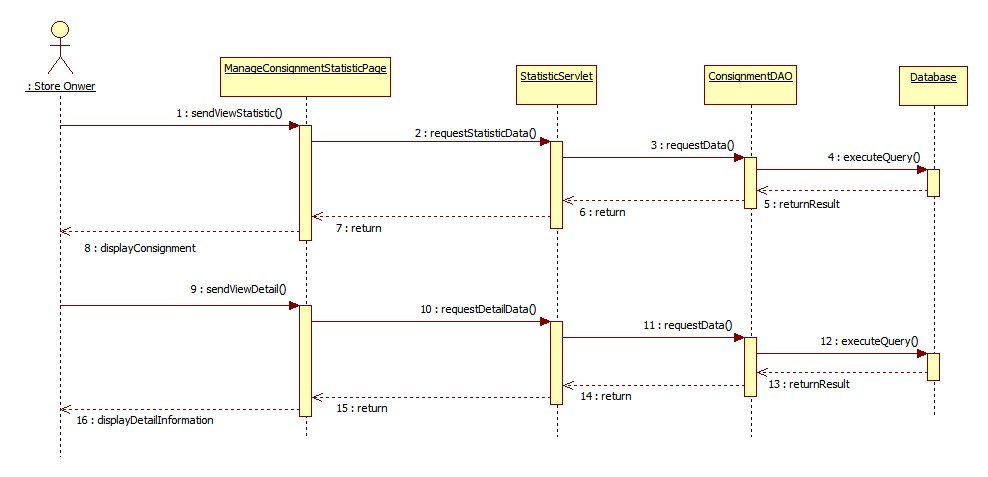


Figure 58: <Store Owner> View Consignment Statistic

##### <Store Owner> Refuse Request

**Summary**: This diagram shows how store owner refuse a consignment request from consignor.

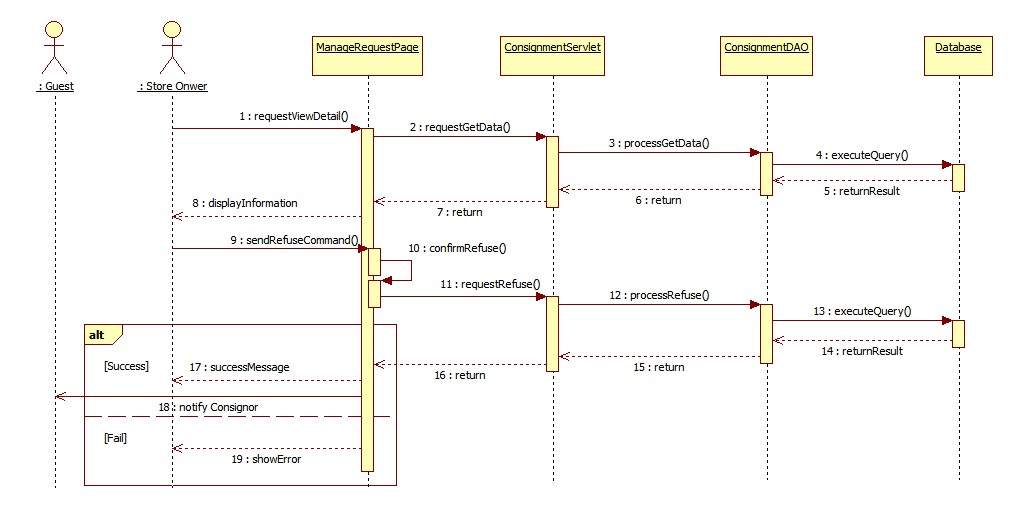


Figure 59: <Store Owner> Refuse Request

##### <Store Owner> Accept Product

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

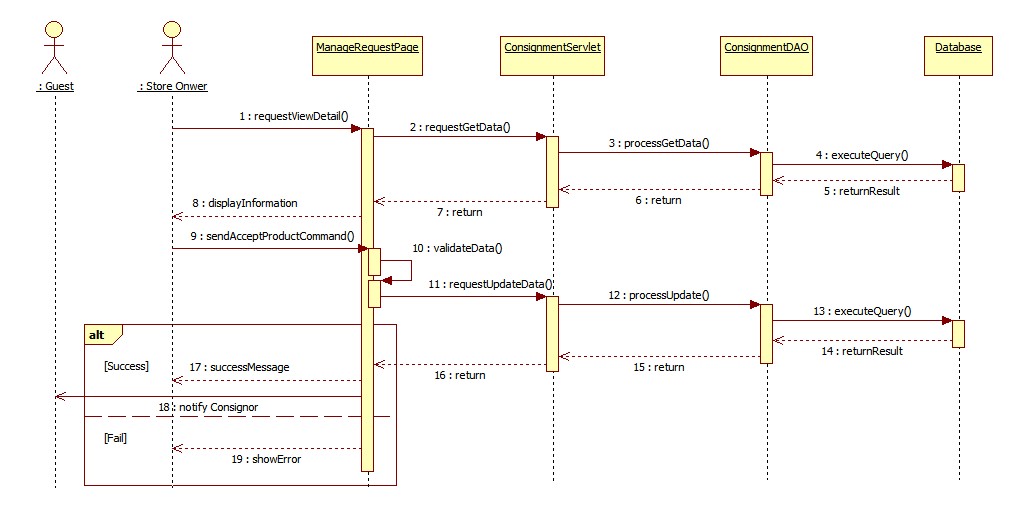


Figure 60: <Store Owner> Accept Product

##### <Store Owner> Refuse Product

**Summary**: This diagram shows how store owner refuses product.

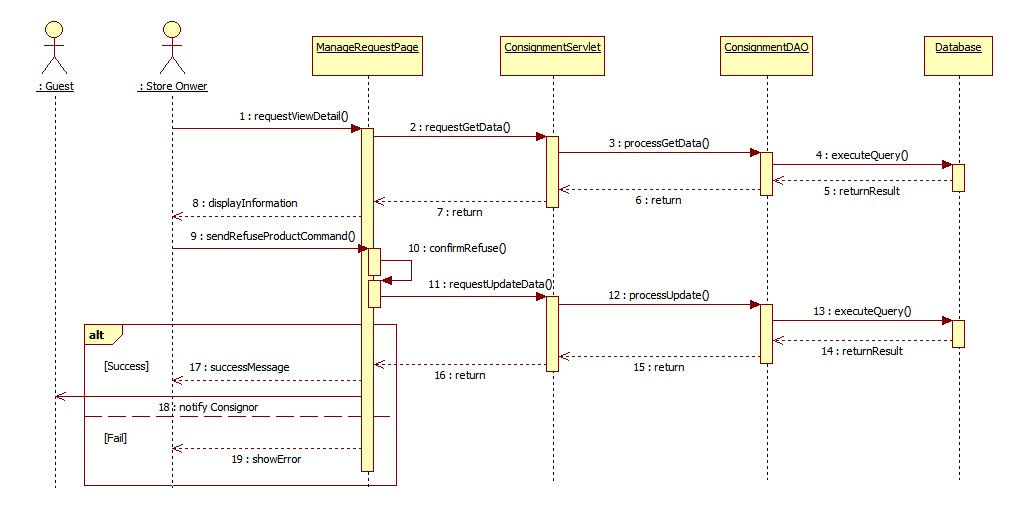


Figure 61: <Store Owner> Refuse Product

##### <Store Owner> Update Request

**Summary**: This diagram shows how store owner update consignment information and notify to consignor.

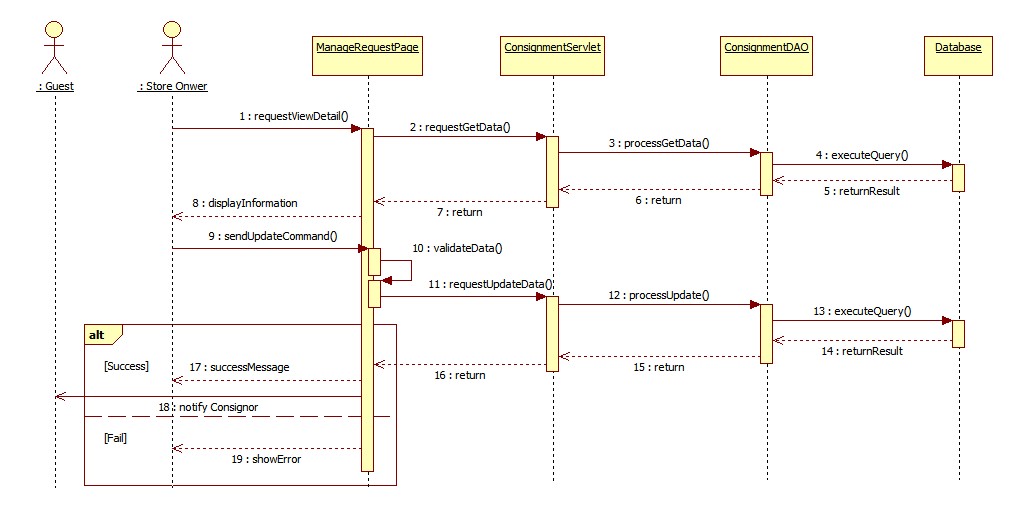


Figure 62: <Store Owner> Update Request

##### <System> Search Product Online

**Summary**: This diagram shows how system find product with information that was given by consignor.

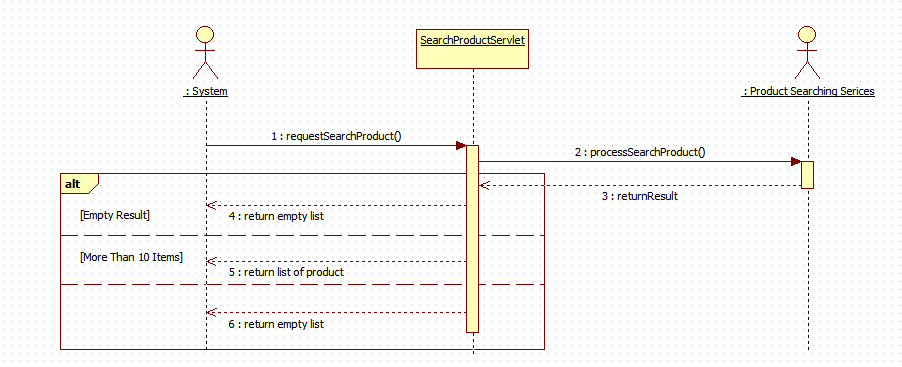
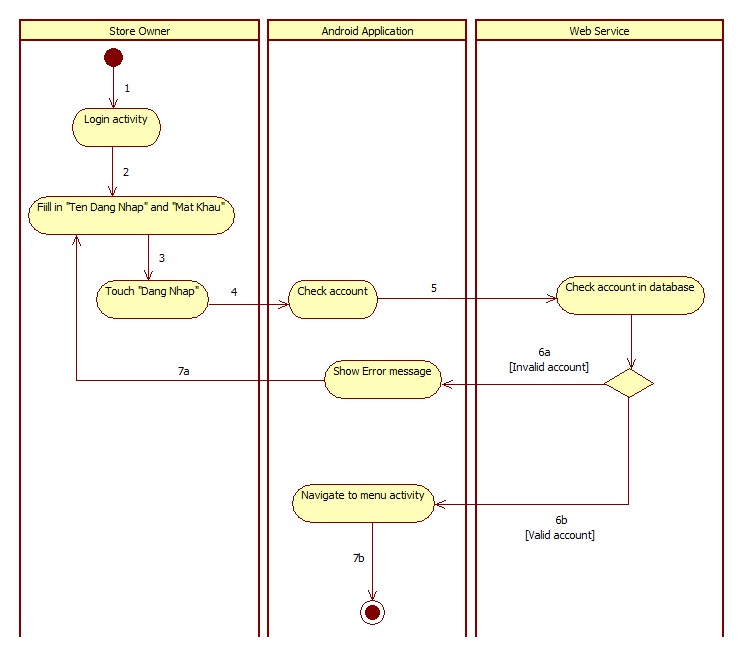


Figure 63: <System> Search Product Online

#### Activity Diagram

##### <StoreOwner> Login

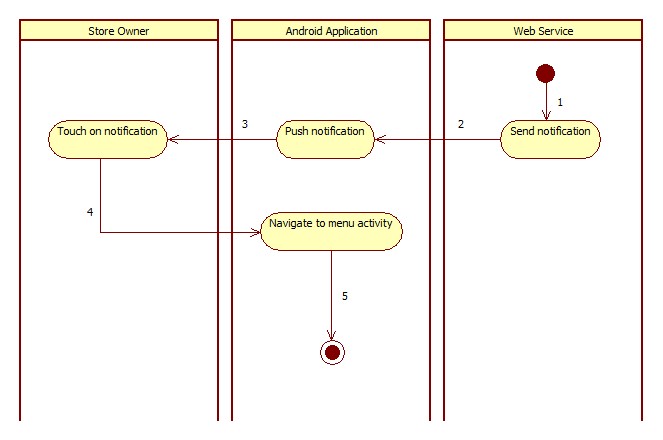
**Summary:** This diagram shows how Store Owner login into system

****

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Store Owner touch “Đăng Xuất” tab | 3. Show confirm dialog |
| 4a. Store Owner touch “Hủy” button | 4b. Store Owner touch “Xác Nhận” button | 5a. Dismiss dialog |
| 5b. call logout() and navigate to login activity | 6a. Finish | 6b. Finish |

##### <StoreOwner> Receive Nofitication

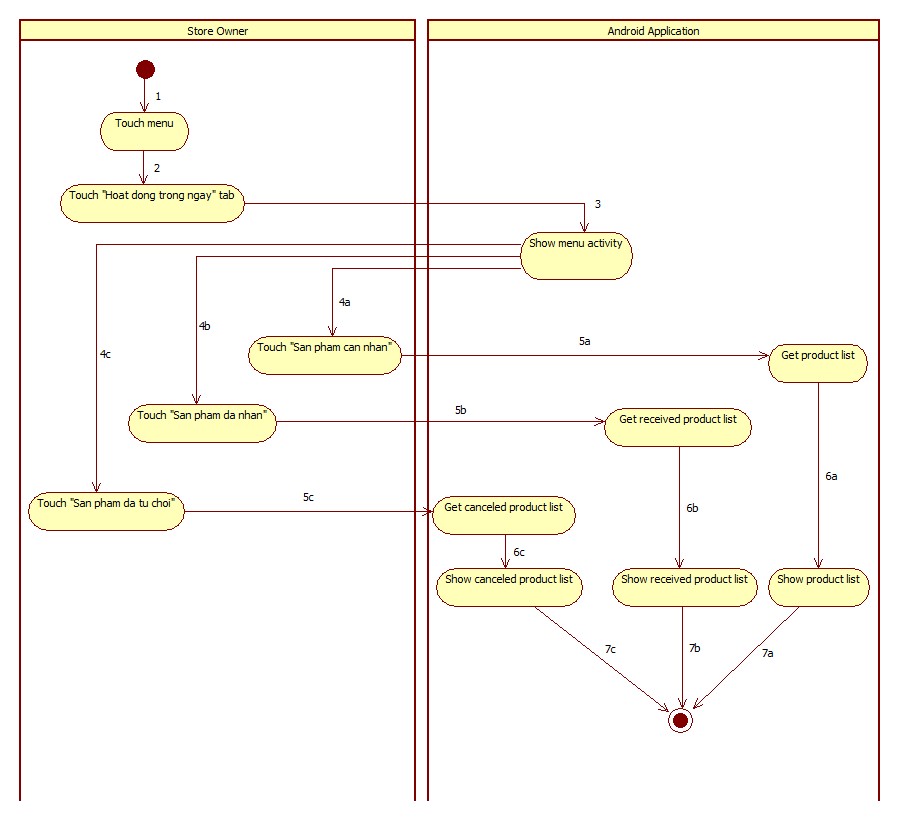
**Summary:** This diagram shows how Store Owner receive notification

****

|  |  |  |
| --- | --- | --- |
| 1. Send Notification | 2. Push Notification | 3. Store Owner touch on notification |
| 4. Navigate to menu activity | 5. Finish |  |

##### <StoreOwner> View Task

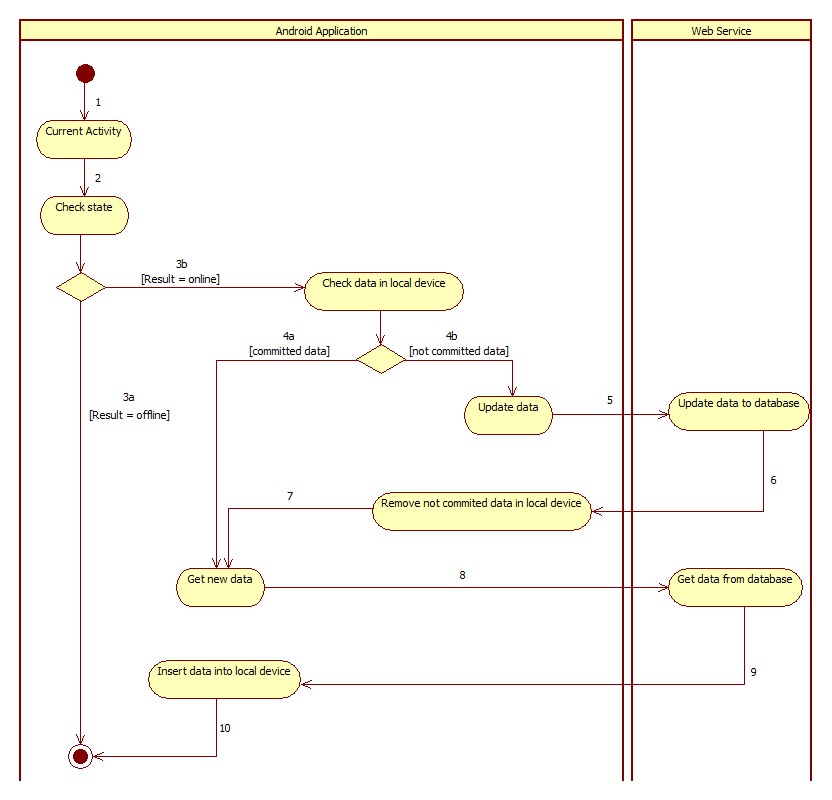
**Summary:** This diagram shows how Store Owner receive notification

****

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Touch tab | 3. Call menu activity |
| 4a. Store Owner touch “Sản phẩm cần nhận” | 4b. Store Owner touch “Sản phẩm đã nhận” | 4b. Store Owner touch “Sản phẩm đã từ chối” |
| 5a. Call getProduct() | 5b. Call getReceivedProduct() | 5c. Call getCanceledProduct() |
| 6a. Show product list | 6b. Show received product list | 6c. Show canceled product list |
| 7a. Finish | 7b. Finish | 7c. Finish |

##### <StoreOwner> Auto Synchronize

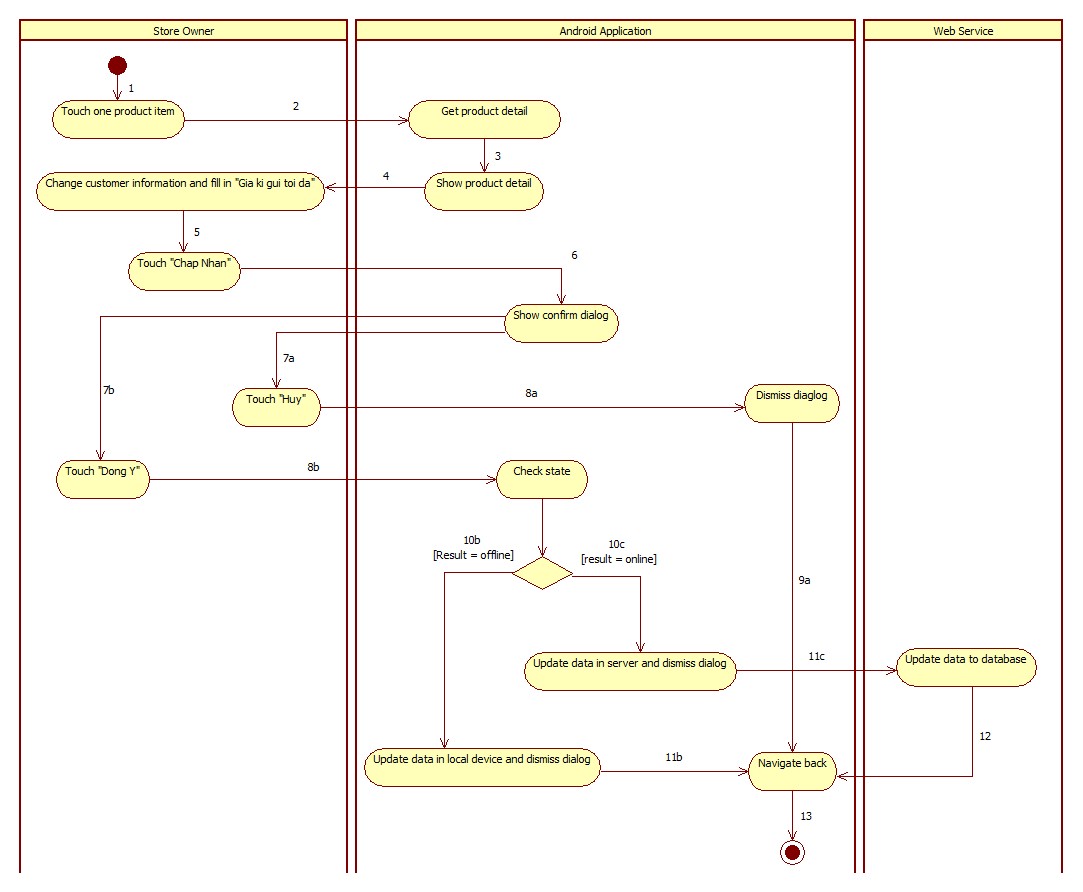
**Summary:** This diagram shows how Mobile Application auto synchronize data.

****

|  |  |  |
| --- | --- | --- |
| 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

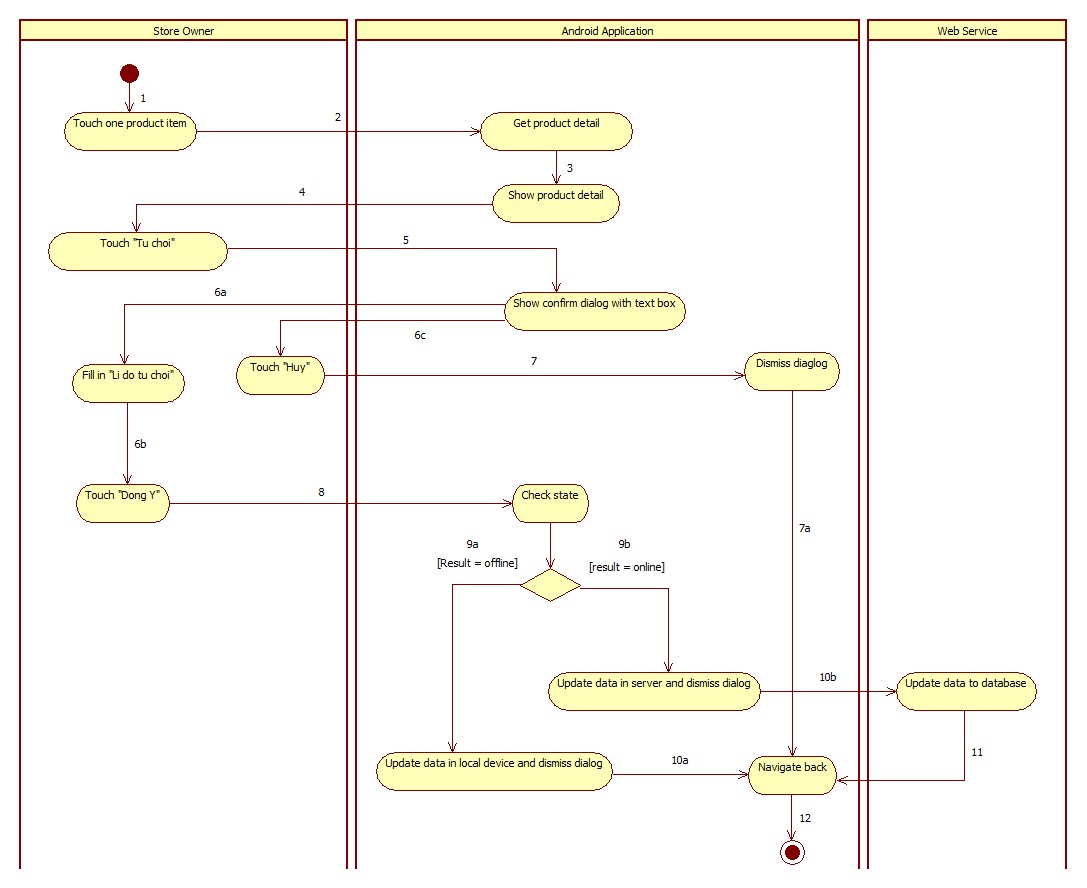
**Summary:** This diagram shows how StoreOwner receive product.



|  |  |  |
| --- | --- | --- |
| 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 |  |

##### <StoreOwner> Cancel Product

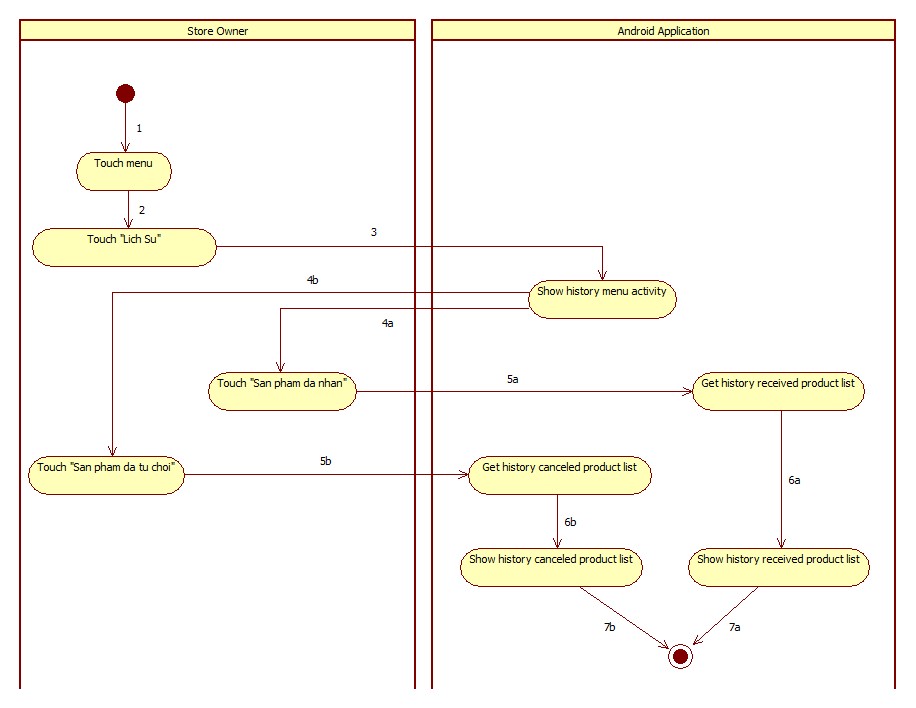
**Summary:** This diagram shows how StoreOwner cancel product.



|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Call getProductDetail() | 3. Show product detail |
| 4. Store Owner touch “Từ chối” button | 5. Show confirm dialog with text box | 6a. Store Owner enter “Lí do từ chối”. |
| 6b. Store Owner touch “Đồng ý” button. | 6c. Store Owner touch “Hủy” button to cancel action. | 7. Dismiss confirm dialog |
| 7a. Navigate back | 8. Call checkState() | 9a. Update data in local device and dismiss dialog. |
| 9b. Call service to update data in server and dismiss dialog. | 10a. Navigate back | 10b. Send data to web service |
| 11. Navigate back | 12. Finish |  |

##### <StoreOwner> View History

**Summary:** This diagram shows how StoreOwner view history

****

|  |  |  |
| --- | --- | --- |
| 1. Start | 2. Touch tab | 3. Call history menu activity |
| 4a. Store Owner touch “Sản phẩm đã nhận” | 4b. Store Owner touch “Sản phẩm đã từ chối” | 5a. Call getHistoryReceivedProduct() |
| 5c. Call getHistoryCanceledProduct() | 6a. show history received product list | 6b. show history canceled product list |
| 7a. Finish | 7b. Finish |  |

## User Interface Design

### Guest/Customer Interface Design

#### Login

### StoreOwner Interface Design

### Admin Interface Design

## Database Design

### Logical Diagram

### Data Dictionary

## Algorithms

# Report No. 5 System Implement & Test Description

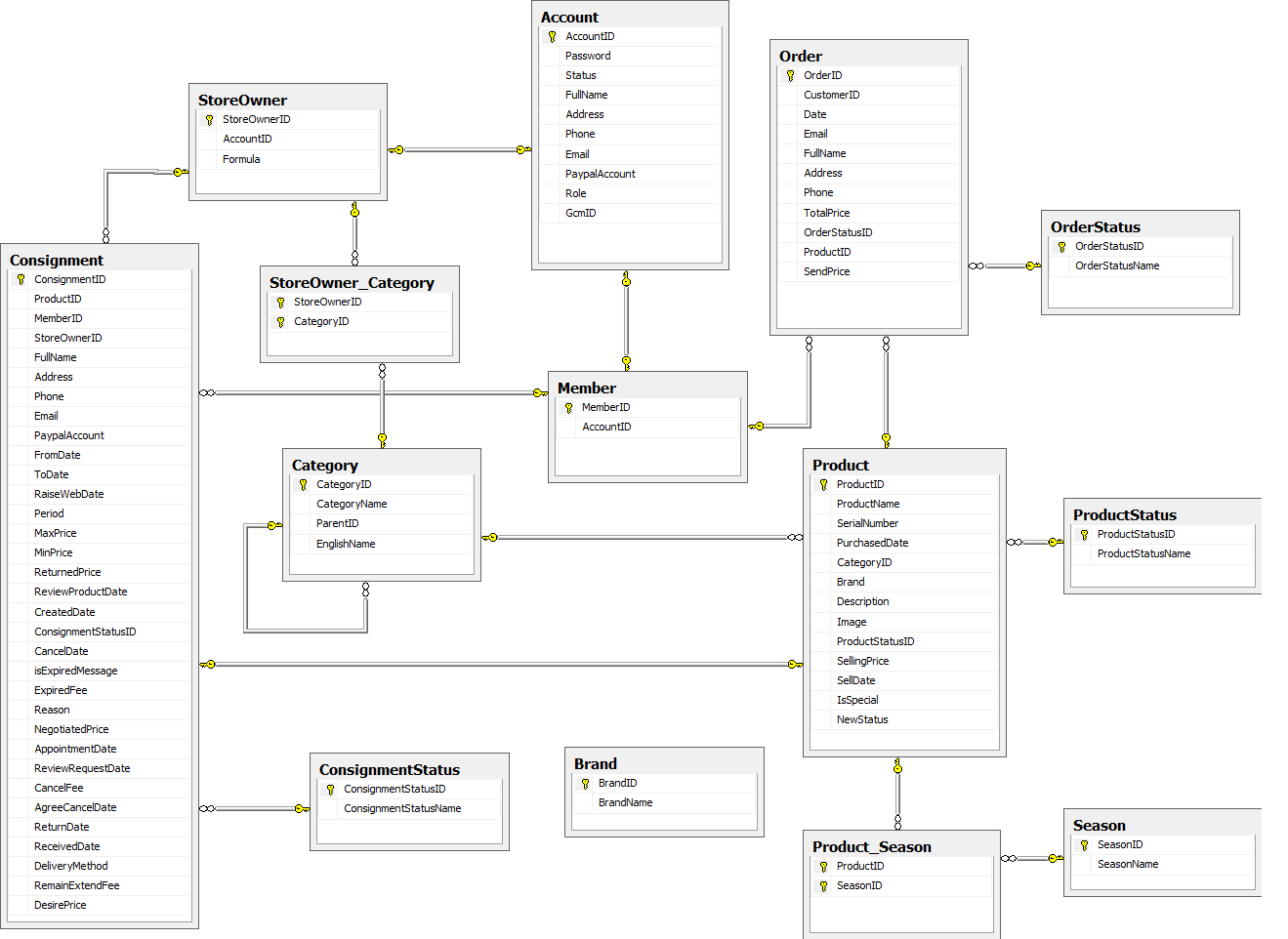
## Introduction

### Overview

### Test Approach

## Database Relationship Diagram

### Physical Diagram



### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| Account | Describe all accounts in the system. |
| StoreOwner | Describe all storeowner profiles in the system. |
| Member | Describe all member profiles in the system. |
| Consignment | Describe all consignments in the system. |
| Product | Describe all products in the system. |
| Order | Describe all orders in the system |
| Category | Describe all product categories |
| StoreOwner\_Category | Describe all product categories that storeowner can accept |
| Season | Describe all seasons in the system |
| Product\_Season | Describe all seasons of product |
| ProductStatus | Describe all statuses of product |
| ConsignmentStatus | Describe all statuses of consignment |
| OrderStatus | Describe all statuses of order |
| Brand | Describe brands in the system |

## Performance Measurement

## Test Plan

### Features to be tested

### Features not to be tested

## System Testing Test Case

### StoreOwner Core Flow

### Guest Core Flow

Reference:

1. Tech in Asia, September 2014, Viet Nam’s smartphone users number *<https://www.techinasia.com/oops-vietnam-22-million-smartphone-users-33-million>*
2. The Guardian, February 2015, location-based dating app user percentage *<http://www.theguardian.com/technology/2015/feb/17/mobile-dating-apps-tinder-two-thirds-men>*
3. Sun Microsystems, 1999, Java Style Coding Convention <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>