|  |  |
| --- | --- |
| logo ngan.png | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Office Rental Service |
|  |
| |  |  | | --- | --- | | **Group 6** | | | **Group Members** | Lê Xuân Tiến – Team Leader – SE60897  Nguyễn Vũ Hoàng Quốc – Team Member – SE61112  Trương Tiến Thành – Team Member – SE61052  Trần Lê Tuấn – Team Member – 60350 | | **Supervisor** | Mr. Nguyễn Trọng Tài | | **Ext Supervisor** | N/A | | **Capstone Project Code** | ORS | |
| - Hồ Chí Minh City, May 2015 - |

*This page is intentionally left blank*

Table of Contents

[A. Introduction 8](#_Toc419298696)

[1. Project Information 8](#_Toc419298697)

[2. Introduction 8](#_Toc419298698)

[3. Current Situation 8](#_Toc419298699)

[4. Problem Definition 8](#_Toc419298700)

[5. Proposed Solution 8](#_Toc419298701)

[5.1 Feature functions 8](#_Toc419298702)

[5.2 Advantage and disadvantage 8](#_Toc419298703)

[6. Functional Requirements 8](#_Toc419298704)

[7. Role and Responsibility 9](#_Toc419298705)

[B. Project Management Plan (PMP) 10](#_Toc419298706)

[1. Problem Definition 10](#_Toc419298709)

[1.1 Name of this Capstone Project 10](#_Toc419298713)

[1.2 Problem Abstract 10](#_Toc419298714)

[1.3 Project Overview 10](#_Toc419298715)

[2. Project organization 11](#_Toc419298716)

[2.1 Software Process Model 11](#_Toc419298719)

[2.2 Roles and Responsibilities 12](#_Toc419298720)

[2.3 Tools and Techniques 13](#_Toc419298721)

[3. Project management plan 13](#_Toc419298722)

[2.3.1 Tasks 13](#_Toc419298724)

[2.3.2 Task sheet 16](#_Toc419298725)

[2.3.3 All Meeting Minutes 16](#_Toc419298726)

[4. Coding Convention 16](#_Toc419298727)

[C. Software Requirements Specifications (SRS) 17](#_Toc419298728)

[1. User Requirement Specification 17](#_Toc419298729)

[1.1 Guest requirement 17](#_Toc419298730)

[2. System Requirement Specification 17](#_Toc419298731)

[2.1 External Interface Requirements 17](#_Toc419298732)

[2.2 System Overview Use Case: 18](#_Toc419298733)

[2.1 System Overview Use Case: 19](#_Toc419298734)

[3. Software System Attribute 21](#_Toc419298735)

[3.1 Reliability 21](#_Toc419298736)

[3.2 Availability 21](#_Toc419298737)

[3.3 Security 21](#_Toc419298738)

[3.4 Maintainability 21](#_Toc419298739)

[3.5 Portability 22](#_Toc419298740)

[3.6 Performance 22](#_Toc419298741)

[4. Conceptual Diagram 22](#_Toc419298742)

[5. Other material (if any) 24](#_Toc419298743)

[D. Software Design Description (SDD) 25](#_Toc419298744)

[1. Design Overview 25](#_Toc419298745)

[2. System Architectural Design 26](#_Toc419298746)

[2.1 Choice of System Architecture 26](#_Toc419298747)

[3. Component Diagram 26](#_Toc419298748)

[4. Detailed Description Explanation 26](#_Toc419298749)

[4.1 Class Diagram 26](#_Toc419298750)

[4.2 Class Diagram Explanation 28](#_Toc419298751)

[4.3 Interaction Diagram 28](#_Toc419298752)

[5. User Interface Design 29](#_Toc419298753)

[5.1 Login 29](#_Toc419298754)

[6. Database Design 31](#_Toc419298755)

[6.1 Logical Diagram 31](#_Toc419298756)

[6.2 Data Dictionary 33](#_Toc419298757)

[7. Algorithms 34](#_Toc419298758)

[7.1 Suggest products which rented by customers who rented current product 34](#_Toc419298759)

[E. System Implementation & Test (SIT) 36](#_Toc419298760)

[1. Introduction 36](#_Toc419298761)

[1.1 System Overview 36](#_Toc419298762)

[1.2 Test Approach 36](#_Toc419298763)

[2. Database Relationship Diagram 36](#_Toc419298764)

[2.1 Physical Diagram 36](#_Toc419298765)

[2.2 Data Dictionary 36](#_Toc419298766)

[3. Performance Measures 36](#_Toc419298767)

[3.1 Clustering Performance 36](#_Toc419298768)

[4. Test Plan 36](#_Toc419298769)

[4.1 Features to be tested 36](#_Toc419298770)

[4.2 Features not to be tested 36](#_Toc419298771)

[5. System Testing Test Case 37](#_Toc419298772)

[5.1 Guest Test Case 37](#_Toc419298773)

[6. Other material 38](#_Toc419298774)

[F. Software User’s Manual 39](#_Toc419298775)

[1. Installation Guide 39](#_Toc419298776)

[1.1 Setting up environment at server side 39](#_Toc419298777)

[1.2 Deployment at server side 39](#_Toc419298778)

[1.3 Client side environment setting 39](#_Toc419298779)

[2. User’s Guide 39](#_Toc419298780)

[2.1 Admin –Staff Login 40](#_Toc419298781)

[G. Appendix 40](#_Toc419298782)

**List of Figures**

[**Figure 1: Waterfall Model** 14](#_Toc419298675)

[*Figure 2: Use Case Overview* 20](#_Toc419298676)

[*Figure 3: <Guest>Overview Use Case* 21](#_Toc419298677)

[*Figure 4: <Guest> Register* 21](#_Toc419298678)

[*Figure 68: Entity Relationship Diagram* 25](#_Toc419298679)

[*Figure 71: Component Diagram* 28](#_Toc419298680)

[*Figure 72: Class Diagram* 29](#_Toc419298681)

[*Figure 73: Add Exchangeable Toy Sequence Diagram* 31](#_Toc419298682)

[*Figure 106: Login* 31](#_Toc419298683)

[**Figure 157: Physical Database Diagram** 34](#_Toc419298684)

[**Figure 158: Extract Deployment Package** 41](#_Toc419298685)

[**Figure 12: Admin - Staff Login Page** 42](#_Toc419298686)

**List Of Tables**

[Table 1: Roles and Responsibilities 9](#_Toc419298687)

[Table 2: Hardware Requirement for Server 11](#_Toc419298688)

[Table 3: Hardware Requirement for Web User 11](#_Toc419298689)

[*Table 4: <Staff>Cancel Exchange* 21](#_Toc419298690)

[*Table 54: Entity Data Dictionary* 24](#_Toc419298691)

[*Table 55: Entity Data Dictionary* 33](#_Toc419298692)

[*Table 56: Data Dictionary of Role* 34](#_Toc419298693)

[*Table 77: Basic Search Test\_Case* 37](#_Toc419298694)

[**Table 9: Login Step** 40](#_Toc419298695)

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| ORD | Office Rental Service |
| SRS | Software Requirement Specification |
|  |  |

# Report No 1 - Introduction

## Project Information

* Project name: **Office Rental Service**
* Project Code: **ORS**
* Product Type: **Website Application**
* Start Date: **May 11th, 2015**
* End Date: **August 22nd, 2015**

## Introduction

Nowadays, a lot of companies are established every day. When new companies are established or some companies want to expand their companies, they need to find where they put their offices. To find a suitable office for them, they need you to add some equipment they need, repair some equipment… All of that thing will get you a lot of time to do it and it is very complex to make contract with those companies. With this system, you can easily to manage your offices, equipment and contracts and it takes you less time than now.

## Current Situation

In Vietnam, to find a suitable office, company would rather consider the following actions:

* If company has large resource, and require a specific requirement and equipment, they would build a new office. This will takes lots of time and money, only adapted for foreign companies.
* Go to a broker company, fill in the form and ask for help to find a suitable office. This usually takes some days to find and cost fee for broker companies.
* Search the internet for office. There are some website in Vietnam has information about the office for rent. But the information is usually not richly or sometimes outdated.

## Problem Definition

To build a new office requires a lot of time, effort, and spend a lot of costs. Besides, there are many new office building has been built recently, but had not been rented, and many company has unused office space. The matching between the demand and the supply is critical required. But using tradition approach for searching office, we have some limitations:

* There are some classifieds website that put the office information. However, the office information is lack, the search criteria is not very effective, and the information about the amenities is hardly found.
* Every company choose for themselves the best office suite with their company: location, area, office facilities, finding a best match will be take time, costs. This work usually made by a third party broker company, and it will take days to finish and we would pay a fee to that company.
* When company want to rent the office or have a rented office but conditions where infrastructure does not guarantee, the company business will be very difficult to contact the lessors to consider repairing office suite company conditions, causing company to consume resources.

## Proposed Solution

The website support customer to select appropriate and process office rental online. The website also support manager to track their business.

### Feature functions

* Our website offers many methods for customer to search office (by size, price, place, amenities…). When they find approtiate office, our staff will arrange to meet them at that office. After check around, we can make a contract right away. The contract will be tracked by our website for further support.
* If customers request repair something in their office, the website will notify our staff to make sure they will fix that problem as soon as posible.
* Our system also supports the addition of devices such as power sockets, fax machines… at the request of customers.

### Advantage and disadvantage

The advantages and disadvantages of the proposed solution:

* Advantages:
  + User friendly and specialized interface.
  + Providing search engines criteria: price, location, amenities...
  + Providing full information about the office. Support map for customers
  + Easy to make an rental for appointment
  + When a deal is made, the contract is tracked by website, so the office state is update in real time for further search.
  + Can request for repair, or add new equipment based on the contract.
  + The repair request status will be update by notification system
  + The repair request is easily manage by automatic assign system
  + The contract making and repair status update is manage by mobile system so the staff will easy to update when go to the offices.
* Disadvantages:
  + The partner who own the office must contact us to post their information.
  + The mobile system require internet connection.
  + The function of notification system is limited.
  + The matching solution based on location and needs will take more time than search by keyword in traditional approach

## Functional Requirements

The functional requirements of the system are based on four main actor as below:

* Guess
  + Search for office base on places, category, amenities
  + View office detail
  + Register information to make an appointment
  + Login
* Customer:
  + Review office
  + Request rental appointment
  + Request repair
* Admin:
  + View list/Create/Update/Delete Manager.
  + View list/Create/Update/Delete Staff.
* Manager:
  + Approve/Cancel/Delete contract
  + Assign repair
  + View Statistic
* Staff:
  + Create contract
  + View list/Create/Update/Delete office
  + Approve/Delete rental
  + Update status request repair.
  + Approve/Delete review

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Nguyễn Trọng Tài | Project Manager | Instructor | [taint@fpt.edu.vn](mailto:taint@fpt.edu.vn) |
| 2 | Lê Xuân Tiến | Developer | Team Leader | [tienlxse60897@fpt.edu.vn](mailto:tienlxse60897@fpt.edu.vn) |
| 3 | Nguyễn Vũ Hoàng Quốc | Developer | Team Member | [quocnvhse61112@fpt.edu.vn](mailto:quocnvhse61112@fpt.edu.vn) |
| 4 | Trương Tiến Thành | Developer | Team Member | [thanhttse61052@fpt.edu.vn](mailto:thanhttse61052@fpt.edu.vn) |
| 5 | Trần Lê Tuấn | Developer | Team Member | [tuantl60350@fpt.edu.vn](mailto:tuantl60350@fpt.edu.vn) |

Table : Roles and Responsibilities

# Project Management Plan (PMP)



## Problem Definition



### Name of this Capstone Project

* **Official name**: Office Rental Service
* **Vietnamese name:** Dịch vụ cho thuê văn phòng
* **Abbreviation:** ORS

### Problem Abstract

Almost medium and small companies don’t build a new office. They usually look up on internet to rent office. But they can’t find a suitable office for them. Or the price to hire a third party broker company is too high. Besides, the office they rent doesn’t provide the necessary equipment and the time to repair is too long. So, our system will solve that problem. It not only helps you find a suitable office but also has other useful services.

### Project Overview

#### Current Situation

There are some current websites such as vanphongthue.com.vn, rongbay.com, timvanphong.vn, etc. All of them have some advantages and disadvantages.

* Advantages:
  + Friendly and specialized interface.
  + Providing search engines criteria.
  + Direct consultation with the manager.
* Disadvantages:
  + Only supply direct meeting, not arrange for a meeting.
  + Not regularly updated information.
  + Not support map.
  + Few pictures describing office.
  + Not provide equipment.

#### The Proposed System

The system will help customers find suitable office. If you don’t find a suitable office, you can make a request. When our system find some office that meet your requirement, it will send email for you. If you need repair office, our system will send staff to fix that problem as soon as possible. Moreover, our system also provide rental equipment for you.

In more detail, the system would contain following features:

##### Website

* Guess can search office and view detail. If they want to make appointment, they need to register.
* Customer can review office, request appointment, request office (if they can’t find ones), request rental equipment and request repair.
* Admin can manage staff and manager.
* Manager can create and delete contract. They also can assign staff to repair for customer and view statistic of system.
* Staff can view list, create, update and delete office. When staff finish repair for customer, they can report to manager. They also can approve or delete rental equipment.

##### Mobile Application

* Staff can check appointment task and update status request repair.

#### Boundaries of the System

* The system is intended for customers who want to rent office (such as businessman, medium or small company,…).
* The language of the system is Vietnamese
* The complete product includes:
  + The website, for admin, staff, customers and guest to interactive with the system.
  + Mobile application for staff to check task and update status request repair.
  + All the process involved document.

#### Development Environment

* **Hardware requirement for server computer**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (2Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window server 2008 | Window server 2012 |
| Computer Processor | Intel® Celeron® Processor 2957U (1.40 GHz) | Intel® Core™ i3-3210 Processor |
| Computer Memory | 1GB RAM | 4GB or more |

Table : Hardware Requirement for Server

* **Hardware Requirement for Web User**

|  |  |  |
| --- | --- | --- |
| **Web** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | 2Mbps | 4Mbps |
| **Web Browser** | Firefox, Chrome, IE 8 | Firefox, Chrome, IE 9 |

Table : Hardware Requirement for Web User

All computers must be connected to the Internet.

* **Hardware requirement for mobile app**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | 2Mbps | 4 Mbps |
| Operating System | Android 4.0 | Android 4.4 |
| Hardware |  |  |
| Memory | 10MB | 10MB |

Table : Requirement for mobile app

* **Software requirements**
  + Operating system: Windows 7, or above;
  + Framework: Hibernate with Java Persistence;
  + Modeling Tool: Visual Paradigm Community;
  + IDE: Intellji Idea;
  + DBMS: Microsoft SQL Server;
  + Source Control: GitHub, Git for windows, Tortoise Git.

## Project organization



### Software Process Model

We choose waterfall model.

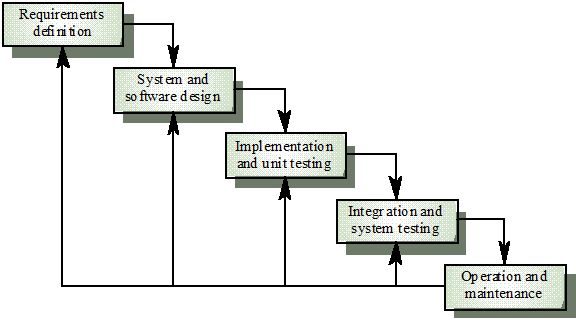


Figure : Software process model

This model is easy to manage and understand. For our project, we don’t have a lot of time so we use this model to help us release our project on time. This model uses for short project and it suitable for our project, which is small with 4 months and requirements are easy to clear.

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Nguyễn Trọng Tài | Supervisor/ Project manager | * Specify user requirement. * Control the development process. * Support technical and business analysis. * Review document and product application. |
| **2** | Lê Xuân Tiến | Team Leader, Developer, Tester | * Monitor process * Create project plan and distribute tasks * Clarify requirements. * Design database. * Prepare documents. * GUI Design. * Create code guide and form. * Coding. * Testing. * Deploy final product. |
| **3** | Nguyễn Vũ Hoàng Quốc | Developer, Tester | * Clarify requirements. * Prepare documents. * Review Database. * GUI Design. * Create test cases. * Coding. * Testing. |
| **4** | Trương Tiến Thành | Developer, Tester | * Clarify requirements. * Prepare documents. * Review Database. * GUI Design. * Create test cases. * Coding. * Testing. |
| **5** | Trần Lê Tuấn | Developer, Tester | * Clarify requirements. * Prepare documents. * Review Database. * GUI Design. * Create test cases. * Coding. * Testing. |

Table : Role and Responsibilities

### Tools and Techniques

* *Front-end*: CSS3, HTML5, JavaScript, JQuery, Ajax, AngularJS, Java Android, PhoneGap.
* *Back-end*: Java 1.7, Hibernate, Java Persistence.
* *Web Server*: Tomcat 7.
* *Developing Tool*: JetBrains IntelliJ IDEA 14, Eclipse.
* *Database Management System:* Microsoft SQL Server 2008.
* *Source Control:* Git-1.9.5-preview, TortoiseGit-1.8.14.0.
* *Modeling Tool*: Visual Paradigm 12.0 Commnunity.
* *Document Tool*: Microsoft Office 2013.

## Project management plan



### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource**  **needed** | **Dependencies**  **and Constrains** | **Risks** |
| **Requirement**  **Analysis** | - Collect requirements.  - Analyze requirements  -Identify and clarify requirements for the system in general. | -Introduction of proposed system. (report 1)  -Project Task Plan. (report 2)  -Software requirement specification. (report 3) | 40 man-days | N/A | - Missing requirement  - Unclear scope of project  - Lack of member share  - Time resource not well managed |
| **Design** | - Design database  - Architecture design for the system  - Detail design using top-down break down  - Choose Architecture style | - Software Design Document  - Base code structure  - Architecture notes  (report 4) | 74 man-days | Depend on “Requirement Analysis” | - Lack of experience.  - Not fulfil requirement. |
| **Implement** | - Implement all system functions | - Source code  - Final deploy packet | 120 man-days | - Software requirement specification.  - Software Design Document | - Lack of member experience.  - Different of member skill |
| **Testing** | - Create test plan  - Create test cases  - Perform unit test cases and system integration test | - Test plan  - Test cases  - Test report  (report 5) | 80 man-days | - Coding is finished  - Based on SRS | - Lack of experience leads to lack of test cases  - Time schedule |
| **Operation and maintenance** | - Create user’s manual, includes server deployment and end-user manual | - Report 6: System User’s Manual | 20 man-days | - Testing is finished  - Based on SRS | - Lack of experience can lead to hard-to-understand document for user  - Time deadline |

Table : Software development lifecycle

### Phase Detail

Below are all the major tasks that need to be performed sequentially during the development of the system.

#### Phase 1: Requirement Analysis

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Collect and analyze requirements** | Find the traditional approach for the service, and current IS system for office rental services. Analyze their strengths and weaknesses | TienLX, QuocNVH, ThanhTT, TuanTL |
| **2. Identify and clarify main functions.** | Define main flows which system will build to adapt, and main functions for system. | TienLX, QuocNVH, ThanhTT, TuanTL |
| **3. Create project management plan** | Define the overview of the project  Create project plan | TienLX, QuocNVH |

Table : Phase 1: Requirement Analysis

#### Phase 2: Design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Defined overall system design** | Design basis system architecture design | TienLX |
| **2. Create entity-relationship diagram** | From the defined use cases, create the ERD for the system | TienLX, QuocNVH |
| **3. Create logical database diagram** | Define the logical database diagram based on ERD | ThanhTT, TuanTL |
| **4. Create class diagrams and definition** | Create the classes with attributes and functions, along with its definition | TienLX, QuocNVH, ThanhTT, TuanTL |
| **5. Create interaction diagrams** | Include sequences diagrams and activity diagrams | TienLX, QuocNVH, ThanhTT, TuanTL |
| **6. Create interface design** | Design the user interface for the system applications | TienLX, QuocNVH, ThanhTT, TuanTL |
| **7. Define main algorithms** | Analyze and define the required algorithms for the system | TienLX, QuocNVH, ThanhTT, TuanTL |

Table : Phase 2: Design

#### Phase 3: Implement

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Design physical database** | Define the physical database design on SQL Server, and create database dictionary | TienLX, QuocNVH, ThanhTT, TuanTL |
| **2. Define base structure** | Define the base structure for the program | TienLX |
| **3. Coding admin functions** | Include manage account | TuanTL |
| **4. Coding manager functions** | Include manage repair list and manage contract list | ThanhTT |
| **5. Coding staff functions** | Include manage office | TienLX |
| **6. Coding customer functions** |  |  |
| **6.1. Coding request office flow** | Include search office, request office, request appointment | BE: QuocNVH  FE: TienLX |
| **6.2. Coding contract signing flow** | Include the assign flow for manager and view assigned flow for staff | ThanhTT |
| **6.3. Coding contract extend/cancel flow** | Include request extend contract and cancel contract | QuocNVH |
| **6.4. Coding request repair flow** | Include request repair office for customers | TuanTL |
| **7. Coding additional mobile app** | Include view assigned job for staff, view assigned and update the repair job for staff | BE: QuocNVH, TuanTL  FE: TienLX, ThanhTT |

Table : Phase 3: Implement

#### Phase 4: Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Create test plan** | Create the test plan document for the system | TienLX |
| **2. Create unit testing** | Create the test cases for each functions describe in the implementation |  |
| **2.1. For admin functions** | Create test cases for functions in manage account | TienLX |
| **2.2. For manager functions** | Create test cases for functions in manage repair and contract | TuanTL |
| **2.3. For staff functions** | Create test cases for functions in manage office | ThanhTT |
| **2.4. For customer functions** | Create test cases for functions for customer | QuocNVH |
| **2.5. For mobile app** | Create test cases for functions for mobile app | TuanTL |
| **3. Testing and fix bugs** | Base on the test cases, test the functions and fix bugs | TienLX, QuocNVH, ThanhTT, TuanTL |
| **4. System integration test** | Test the overall of the system, measure the performance of the system | TienLX, QuocNVH, ThanhTT, TuanTL |

Table : Phase 4: Testing

#### Phase 5: Operation and maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Create installation guide** | Define the system requirement for the server and client. document installation guild | TienLX, QuocNVH |
| **2. Create User guide** | Define the user manual for operating the system | TienLX, QuocNVH, ThanhTT, TuanTL |
| **3. Maintenance and fix bugs** | Tracking the system and fix the remains bugs while installation | ThanhTT, TuanTL |

Table : Phase 5: Operation and maintenance

### All Meeting Minutes

Refer to Meeting Minutes folder.

## Coding Convention

**Indentation:**

* Four spaces should be used as the unit of indentation
* Tabs must be set exactly every 8 spaces
* Avoid lines longer than 80 characters
* Break after a comma.

**Declarations:**

* One declaration per line
* Put declarations only at the beginning of blocks
* No space between a method name and the parenthesis "(" starting its parameter list
* Open brace "{" appears at the end of the same line as the declaration statement
* Closing brace "}" starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the "}" should appear immediately after the "{"

**White Space:**

* Between sections of a source file
* Between class and interface definitions
* Between methods

**Naming Conventions:**

* Class names should be noun
* Methods should be verbs

Reference: <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

# Software Requirements Specifications (SRS)

## User Requirement Specification

The system has five actors including: guest, customer, staff, admin and system.

|  |  |
| --- | --- |
| **Actor** | **Description** |
| General User | Person who want to view and refer toys for children. |
| Guest | Person join to website but not login into system. |
| Customer | Person who want to rent toys and logged into system. |
| Admin | Administrator, who monitor and configured the system. |
| Staff | Person who manage the business of system such as: manager toys, orders, reservation…. |
| System | System is also an actor, do every automatic functions according to set up. |

### Guest requirement

* **Register:** Guest need to register to become member of system.
* **Login**: Guest uses email and password to login into the system to rent or exchange toys.

## System Requirement Specification

### External Interface Requirements

#### User Interfaces

* User interface must be user friendly, easy to use.
* All functions are arranged and displayed clearly, don’t make user confuse when using.
* User interface should be attractive and colorful to be appropriate for toy’s service purpose.
* All the alert, notification, warning, display error shouldn’t make user feel afraid, confuse or annoyed.

#### Hardware Interfaces

* The system will use the standard hardware and data communications resources.

#### Software Interfaces

* The website run in Firefox and Chrome browsers.

#### Communications Protocol

* The website uses:
* HTTP/HTTPS protocol for communication between the web browser and the web server.
* TCP/IP network protocol for communication with HTTP protocol

### System Overview Use Case:

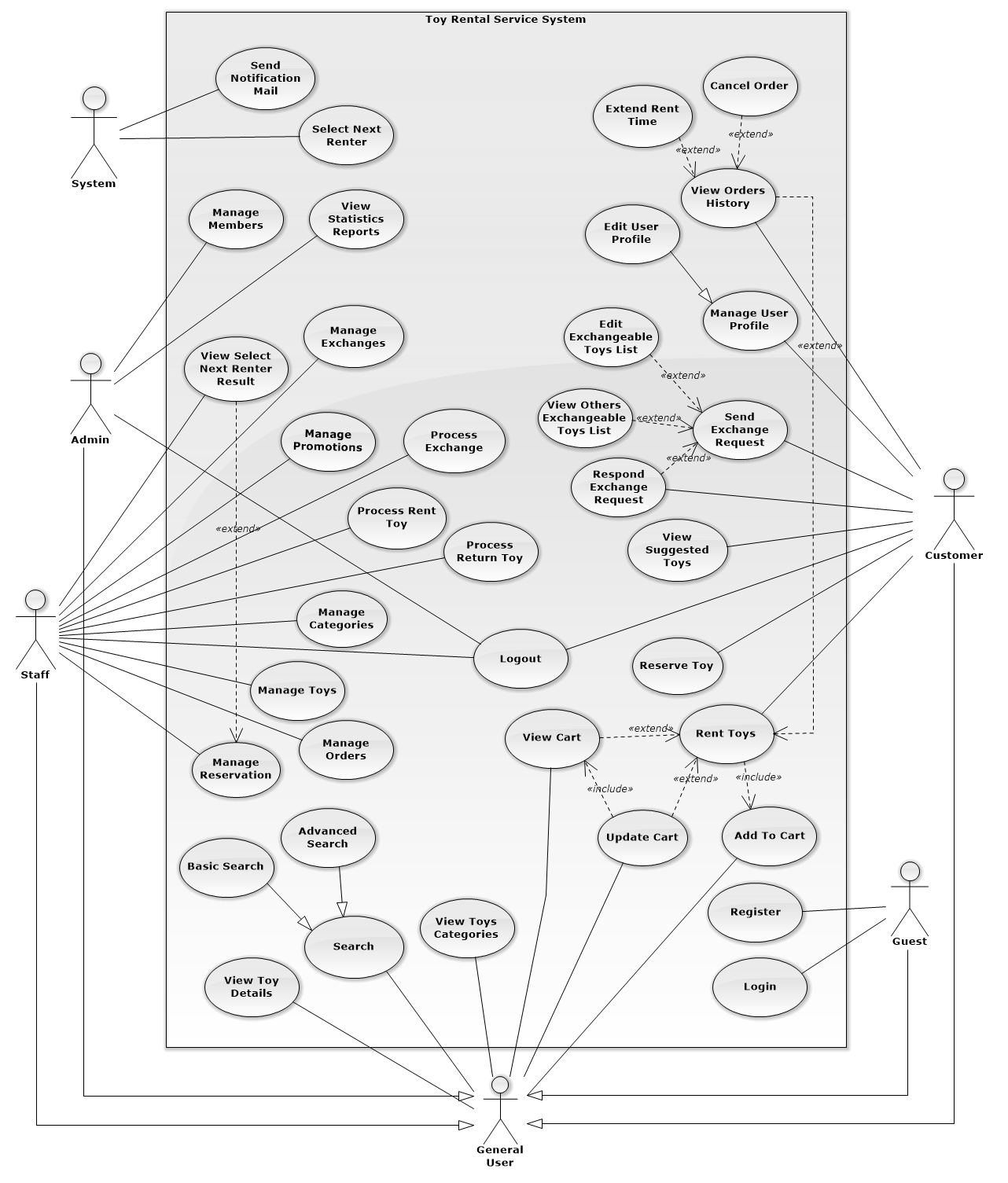


Figure : Use Case Overview

### System Overview Use Case:

##### **<Guest>Overview Use Case**

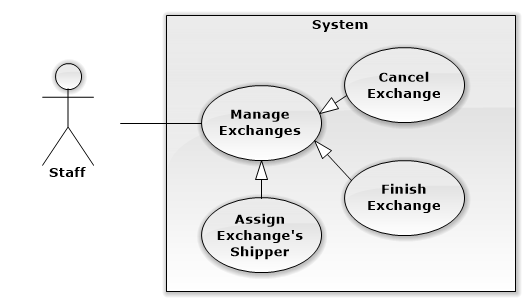


Figure : <Guest>Overview Use Case

##### **<Guest> Register**

**Use Case Diagram**

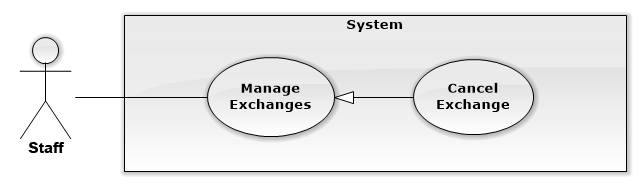


Figure : <Guest> Register

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CANCEL EXCHANGE** | | | |
| **Use Case No.** | UC001 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel Exchange | | |
| **Author** | Nguyễn Ngọc Tường Vy | | |
| **Date** | 13/09/2014 | **Priority** | Normal |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to cancel exchange.   **Goal:**   * Staff can cancel an exchange process.   **Triggers:**   * From the sidebar:   + Select “Quản lý trao đổi” menu.   + Select “Trao đổi đã xác nhận” or “Trao đổi đang tiến hành” sub menu   + Clicks “Hủy” button on details of exchange he want to cancel   **Preconditions:**   * User must log in the system with Staff role. * Staff wants to cancel exchange.   **Post Conditions:**   * **Success:** Exchange’s status is update to “cancelled”. * **Fail:** Exchange’s status isn’t updated in database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Clicks “Hủy” button on details of exchange he want to cancel | Display warning popup to confirm Staff’s action:   * “Bạn có chắc chắn muốn hủy yêu cầu trao đổi này không?” message * “Có” button * “Không” button | | 2 | Clicks “Có” button to confirm [Alternative 1] | Update that exchange’s status to “cancelled”.  Display success message: “Yêu cầu trao đổi đã được hủy”  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Clicks “Không” button to cancel. | Close warning popup and do nothing. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot update exchange’s status due to database connection. | Display error message: “Không thể hủy trao đổi. Vui lòng thử lại sau” |   **Relationships:** Manage Exchanges  **Business Rules:**   * Exchange with finished status, which staff has finished exchange cannot be canceled. * Staff should notify to customers before cancel their exchange. | | | |

Table : <Staff>Cancel Exchange

## Software System Attribute

### Reliability

* Mean Time Between Failures is more than 6 months.
* Accuracy: 100%.
* Critical bugs: not any

### Availability

* N/A

### Security

* All input data should be validated before saving to database
* All privacy information, such as password, should be encrypted to ensure security.
* Roles permission should be specify clearly and user should be authenticated and authorized when accessing to the system.

### Maintainability

* All program files shall include comments concerning authorship and date of last change.
* The system should be divided into modules and code should be easy to maintain and extend.

### Portability

* N/A

### Performance

* All requests should be response in no more than 1 minute.

## Conceptual Diagram

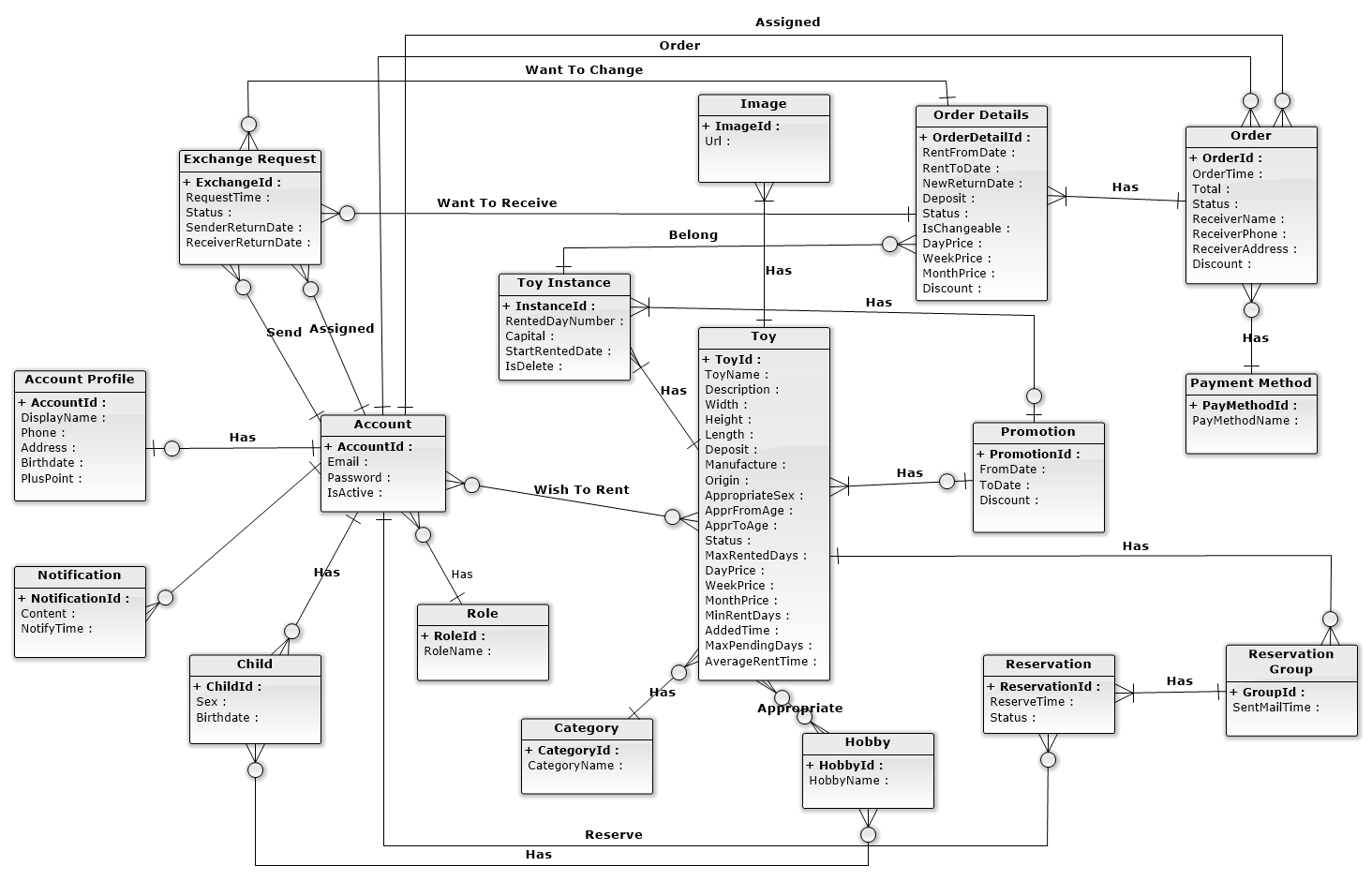


Figure : Entity Relationship Diagram

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| Account | Describe account of user of system. |
| Role | Describe role of user. One user has one role. |
| Child | Describe child of customers in system. |
| Hobby | Describe all normal hobbies of real child. |
| AccountProfile | Describe information of account in system. |
| Notification | Describe all of user’s notification of the system. |
| Toy | Describe all information of toys in system. |
| Image | Describe all image of toys in system. |
| ToyInstance | Describe all instances of a toy in system. |
| Category | Describe all kind of toys in system. |
| Promotion | Describe all information of promotion in system. |
| OrderDetails | Describe all details of an order in system. |
| Order | Describe all orders of the renter in system. |
| PaymentMethod | Describe all payment method of customer in system. |
| Reservation | Describe all toys booking of customer in system. |
| ReservationGroup | Describe all booking group of each toy in system. |
| ExchangeRequest | Describe toys exchanges of customer in system. |

Table : Entity Data Dictionary

## Other material (if any)

N/A

# Software Design Description (SDD)

## Design Overview

This document describes the technical and user interface design of the Children's Toy Rental Service Website. Document includes the system architectural design, component diagram, detailed description of the components, user interface, and database design.

The system architectural design describes the overall architecture of the system, general description of the model used, including detailed information for each component, the function of each component in the system, description how the system works and the way that sub-components communicate together.

The component diagram show the physical structure of the command line in accordance with the concept of part code. Contains information about the class or classes that logic execution, component diagram also shows the dependencies between the components together.

The detailed design describes includes class diagrams, class diagram explanations, and sequence diagrams of all use cases. Class diagram describes the static view of system with class concepts and the relationships between them. Sequence diagram describes in detail the interactions between objects.

All diagram using UML 2.0

The user interface design describes the layout of the system.

The database design describes detail entities and relationships between them.

Document overview:

* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagrams that describe the connection and integration of the system.
* Section 4: gives the detail design description include class diagram, class explanation
* Section 5: sequence diagram to details the application functions.
* Section 6: overview some main user interface of system.
* Section 7: describe database design and details of database with data dictionary.
* Section 8: describe in details all algorithms used in the system.

## System Architectural Design

### Choice of System Architecture

The system is developed using MVC combine with 3-tiers architechture.

## Component Diagram

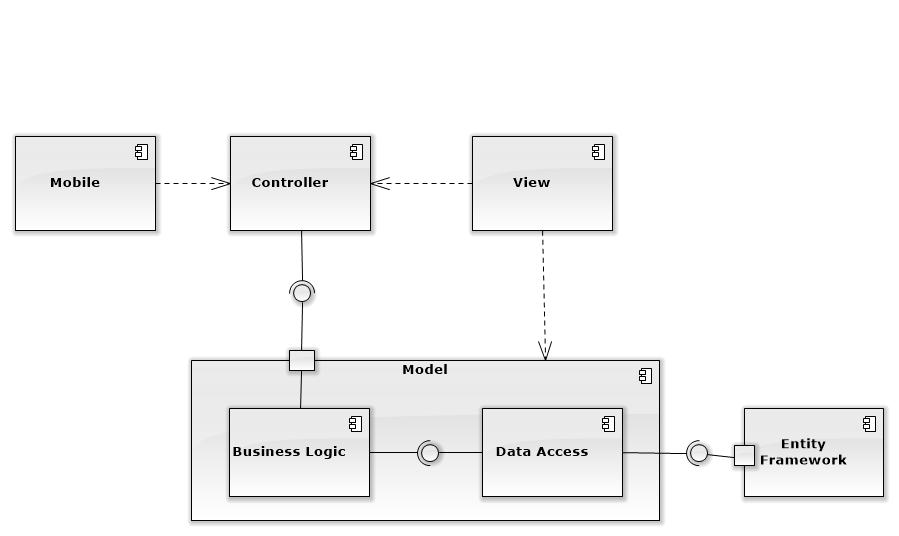


Figure : Component Diagram

## Detailed Description Explanation

### Class Diagram

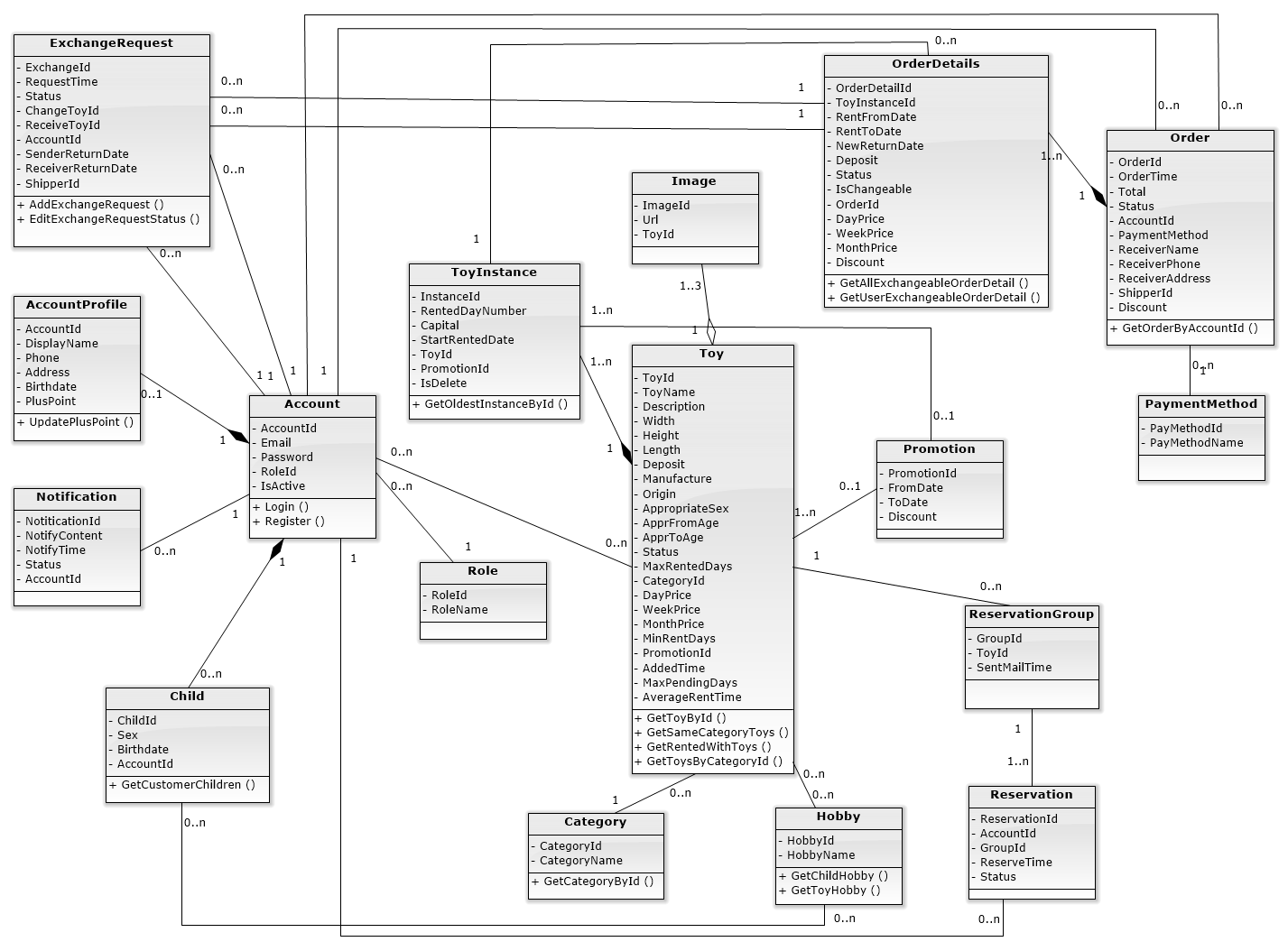


Figure : Class Diagram

### Class Diagram Explanation

#### OrderDetails

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| OrderDetailId | Int | Public | Unique ID of each order detail |
| ToyInstanceId | Int | Public | ID of rented toy instance |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| GetAllExchangeableOrderDetail | List<OrderDetail> | Public | Get all order details that can be exchanged |
| GetUserExchangeableOrderDetail | List<OrderDetail> | Public | Get all order detail of current user that can be exchanged |

### Interaction Diagram

#### Add Exchangeable Toy

**Summary:** <Nên có phần tóm tắt trước 1 diagram nói về mục

đích của diagram trước khi thể hiện hình vẽ>.

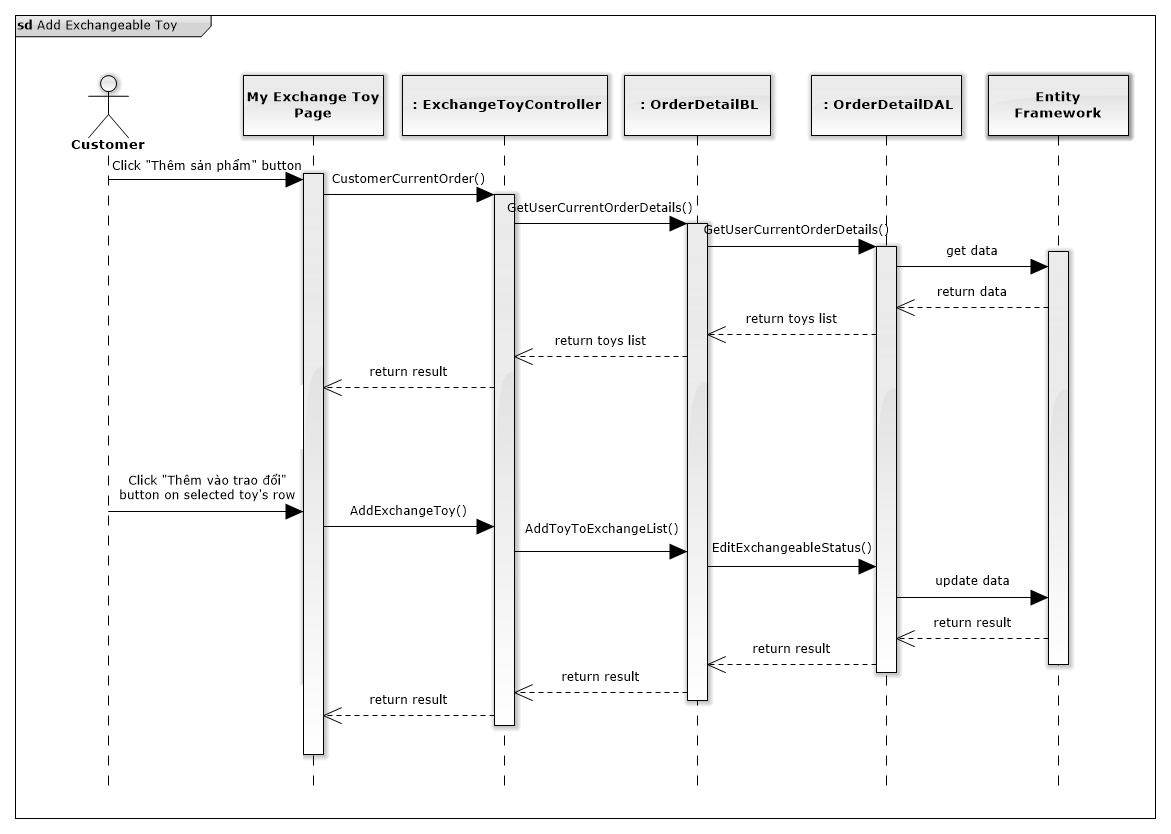


Figure : Add Exchangeable Toy Sequence Diagram

## User Interface Design

### Login

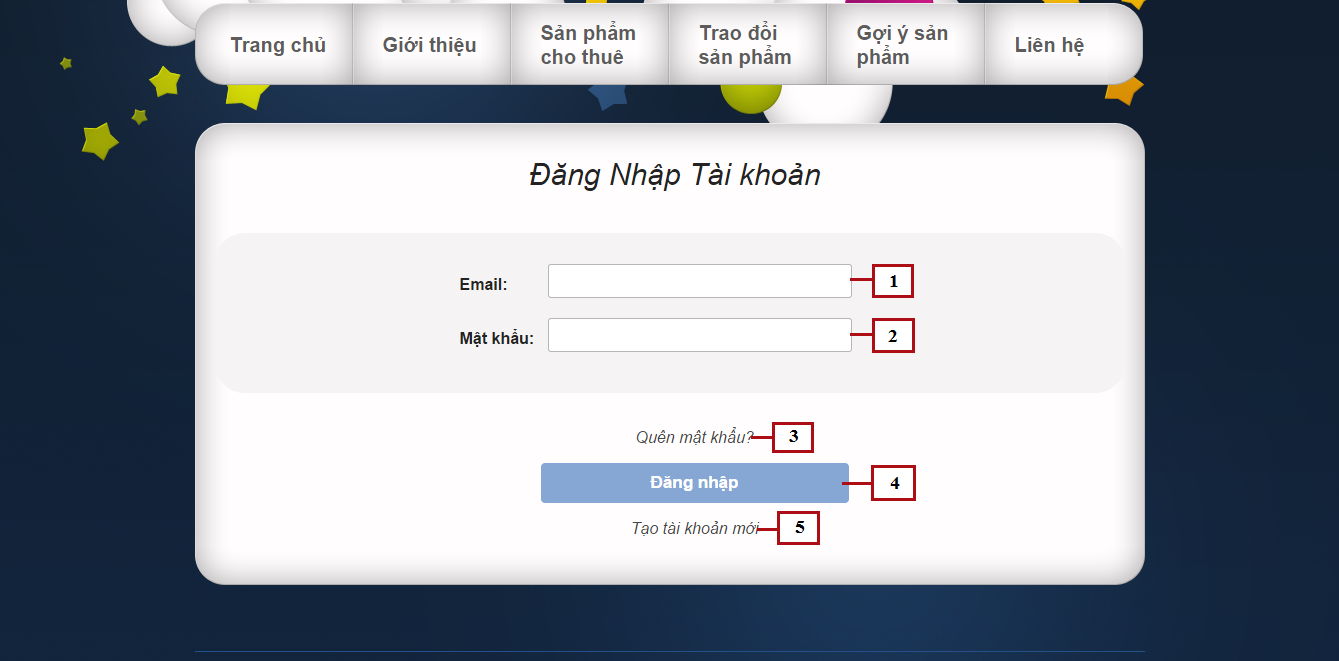


Figure : Login

**Fields**

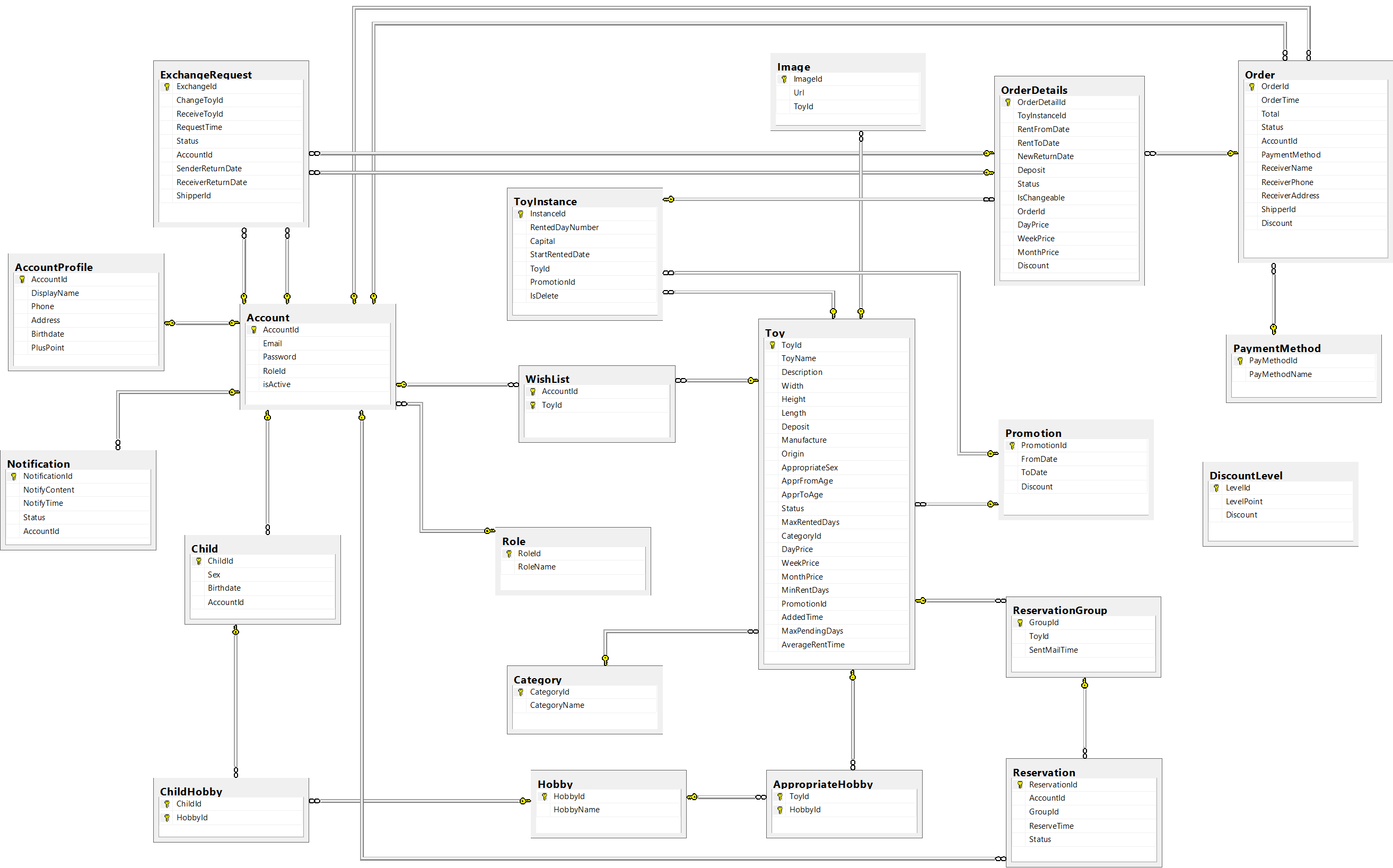
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Email | Fill user email | No | Yes | Textbox | String | N/A |
| 2 | Password | Fill password | No | Yes | Password | String | [8-32] |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Forget password | Click when forget password, to request new password | N/A | N/A |
| 3 | Login | Click to login into the system | Yes | Transfer to home page |
| 5 | Register | Click to register a new account | N/A | Transfer to register page |

## Database Design

### Logical Diagram



**Figure 10: Physical Database Diagram**

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| Account | Describe account of user of system. |
| Role | Describe role of user. One user has one role. |
| Child | Describe child of customers in system. |
| ChildHobby | Describe all hobbies of each child. |
| Hobby | Describe all normal hobbies of real child. |
| AccountProfile | Describe information of account in system. |
| Notification | Describe all of user’s notification of the system. |
| Toy | Describe all information of toys in system. |
| Image | Describe all image of toys in system. |
| ToyInstance | Describe all instances of a toy in system. |
| Category | Describe all kind of toys in system. |
| AppropriateHobby | Describe suitable hobbies of toys in system. |
| WishList | Describe favorite toys list of customer in system. |
| Promotion | Describe all information of promotion in system. |
| OrderDetails | Describe all details of an order in system. |
| Order | Describe all order of the renter in system. |
| PaymentMethod | Describe all payment of customer in system. |
| Reservation | Describe all toys booking of customer in system. |
| ReservationGroup | Describe all booking group of each toy in system. |
| DiscountLevel | Describe percent of discount based on plus point of customer |
| ExchangeRequest | Describe all toys exchange of customer in system. |

Table : Entity Data Dictionary

#### Table Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Role | RoleID {PK} | Unique identifier of a role, auto increment. | Integer | No |
| RoleName | Role name of each role | Varchar(10) | No |
| Unique Key: RoleName | | | | |

Table : Data Dictionary of Role

## Algorithms

### Suggest products which rented by customers who rented current product

#### Requirement:

Suggest related product for current viewed product by getting products most rent by customers who rented current product.

#### Solution:

1. Get all customers who rented current product
2. Get all products rented by these customers which is different from current product
3. Sort product list in order of most rented
4. Get top 5 products to suggest for customer

#### Example:

Current product: Gấu bông teddy

1. Customer who rented this product: Customer 1, Customer 2, Customer 3
2. Product these customers had rented:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Customer 1 | Customer 2 | Customer 3 |
| Gấu bông teddy | X | X | X |
| Cún đốm | X | X |  |
| Nhà thả 12 khối | X |  |  |
| Bàn nhạc trẻ em |  | X | X |
| Bập bênh puppy | X | X | X |
| Xe tập đi |  | X |  |

1. Sorted list of most rented: Bập bênh puppy, Cún đốm, Bàn nhạc trẻ em, Nhà thả 12 khối, Xe tập đi
2. Top 5 products: Bập bênh puppy, Cún đốm, Bàn nhạc trẻ em, Nhà thả 12 khối, Xe tập đi

# System Implementation & Test (SIT)

## Introduction

### System Overview

This section provides in detail all necessary information about test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases of CTS system.

### Test Approach

Goal: Check all the features in CTS system and record remain bug to fix.

Type: White box testing, Black-box Testing.

Size: System Component.

Technique: Check list.

## Database Relationship Diagram

### Physical Diagram

### Data Dictionary

## Performance Measures

### Clustering Performance

## Test Plan

### Features to be tested

We will carry out test based on core workflow of system. All main functions will be tested carefully and clearly following phases.

* Guest: add to cart, view cart, update cart, view product detail, basic search, and advance search, register

### Features not to be tested

* Login, Logout.

## System Testing Test Case

### Guest Test Case

#### Search Event

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Pre-Condition** | **Test Case Procedure** | **Expected output** | **Result** | **Test Date** |
| BS01 | Basic Search with valid inputted key word | N/A | 1 – Input “bup be” into search fields.  2 – Press enter key | - After step 2, System redirected to search page and display 3 results:   1. Thú Bông\_Disney Alice in Wonderland. 2. Cún đốm (Búp bê & Thú bông category). 3. Gấu bông teddy (Búp bê & Thú bông category) | Passed | 30/11/2014 |
| BS02 | Basic Search with wrong key word | N/A | 1 – Input “abc” into search fields.  2 – Press enter key | - After step 2, System redirected to search page and display message “Hiện tại không có sản phẩm mà bạn muốn tìm” | Passed | 30/11/2014 |

Table : Basic Search Test\_Case

## Other material

N/A

# Software User’s Manual

## Installation Guide

### Setting up environment at server side

The following software must be installed into the server machine:

#### Hardware requirements

Server computer for deploying with the minimum configuration:

* CPU Intel® Core 2 Duo
* 2GB RAM.
* 30GB of hard disk.
* Internet

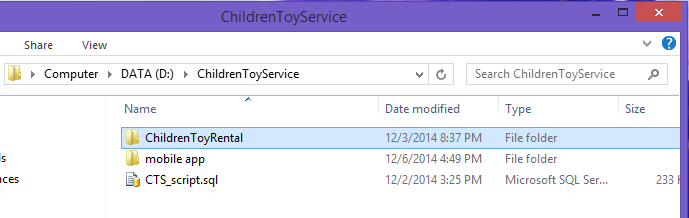
#### Software requirements

* Web Server: Internet Information System – IIS 7.
* Operating system: Windows Server 2008, or above
* SQL Server 2012 Express, or above: used to create and manage database of web application.

### Deployment at server side

#### Prepare deployment package

* Extract the deployment package to a folder on the server.
* For example: D:\ChildrenToyService



**Figure 11: Extract Deployment Package**

#### Database Deployment

### Client side environment setting

Client side should have one of these following browser:

* Google Chrome
* Firefox
* Internet Explorer

## User’s Guide

This user guide describes how to use the application based on the process of children toy rental service website

### Admin –Staff Login

**Figure 12: Admin - Staff Login Page**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Enter “Email” và “Mật khẩu”  (E.g: Email: admin@gmail.com. Mật khẩu: 12345678) |
| 2 | Click “Đăng nhập” |

**Table 17: Login Step**

# Appendix