



MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Tên đề tài

Nhóm số 3	
Group members	Tên các thành viên trong nhóm – Mã số sinh viên
Supervisor	Giảng viên hướng dẫn
Ext. Supervisor	N/A
Capstone Project code	Mã đề tài

-Ho Chi Minh City, *Ngày bắt đầu làm-*

This page is intentionally left blank

Table of Contents

Table of Contents	3
List of Tables.....	4
Definitions, Acronyms, and Abbreviations.....	6
A. Report No. 1 Introduction	7
1. Project Information	7
2. Introduction	7
3. Current Situation	7
4. Problem Definition	7
5. Proposed Solution	8
5.1 Feature functions	8
5.2 Advantages and disadvantages	8
6. Functional Requirements	9
6.1 Name Card Management	Error! Bookmark not defined.
6.2 Event	Error! Bookmark not defined.
6.3 Searching	Error! Bookmark not defined.
6.4 Suggestion	Error! Bookmark not defined.
6.5 User Management	Error! Bookmark not defined.
7. Role and Responsibility	9

List of Tables

Table 1: Roles and Responsibilities9

List of Figures

Figure 1: Modified Waterfall Development Model.....**Error! Bookmark not defined.**

Definitions, Acronyms, and Abbreviations

Miêu tả từ viết tắt hay các term dùng trong tài liệu thuyết minh bên dưới

Name	Definition
Từ viết tắt	Định nghĩa

A. Report No. 1 Introduction

1. Project Information

- Project name: <Tên đề tài>
- Project Code: <Mã đề tài>
- Product Type: <Sản phẩm (web app, desktop app , mobile app)>
- Start Date: <Ngày bắt đầu>
- End Date: <Ngày kết thúc>

2. Introduction

<Nhập đề: giới thiệu sơ nét về đề tài, có thể ghi các vấn đề cần giải quyết, các giải pháp, các công nghệ dẫn đến nhu cầu của đề tài ở tầm khái quát, tổng quan>

Ví dụ:

In this document, we introduce a solution for motorbike insurance company. Current insurance company systems have some problems like delayed in renew contracts for customer or inconvenient in checking insurance card validation process. Based on our researches and analysis, we proposed a solution for insurance company in Vietnam and other developed countries.

We build a system, which help the insurance companies to solve current problems. In the process of analysis, we believe the NFC cards is capable to resolve the problem by using NFC card to save information about insurance contract. NFC cards are convenient to manage the contract information and checking, validating process. Beside of that we also provide an information system to manage NFC cards so that insurance companies will manage the contracts easier.

This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows. We hope the system and our solution will help resolve the problems from insurance companies in Vietnam and other developed countries.

3. Current Situation

<Mô tả về hệ thống hiện tại trong thực tế hoặc hành vi của người dùng hiện tại>

Ví dụ

When participating in traffic, vehicle owners are required to have compulsory insurance (according to Article 6, Decree on compulsory insurance for civil liability of motor vehicle owners, Decree No. 103/2008/ND-CP by Vietnam Government). Therefore, vehicle owners buy insurance from insurance companies or its agents. They pay insurance premium by cash or in online website and receive an insurance certificate with a term of one year, the term can be shorter in some specific situation. When their insurance out of date, they must buy a new insurance, old certificate will be useless. Traffic police will read insurance certificate to check traffic participants.

4. Problem Definition

<Định nghĩa vấn đề: nêu ra các khó khăn, khuyết điểm, hạn chế ở hệ thống hiện tại>

Ví dụ

Below are disadvantages of current situation:

- *Forget insurance's expired date: Vehicle owners usually keeps their insurance certificate in wallet or somewhere on their vehicle. However, except in cases of necessity, people are not often check their insurance so they could forget its expired date. An expired insurance is not good while it be revealed by traffic officers and could get worse in case of traffic accident.*
- *Hard for traffic officers to check and verify insurance:*
- *...*

5. Proposed Solution

<Giới thiệu về giải pháp mà nhóm đưa ra để giải quyết vấn đề>

Ví dụ:

Our proposed solution is to build an insurance NFC card system named "MIC system" to resolve the current situations and compatible with current laws, we also design the system to be scalable so we can deploy this system to a multiple insurance services company in future plan.

MIC system includes a web application and two mobile applications with following functions:

5.1 Feature functions

<Nêu ra các tính năng cốt lõi, các vai trò cốt lõi trong giải pháp mà nhóm đề xuất, chỉ nên nêu các tính năng chủ chốt giải quyết bài toán, không phải liệt kê toàn bộ tính năng>

Ví dụ

- *Web application:*
 - *Register insurance: user can register a new insurance card with on website using online payment. A staff will contact the user to create contract and sends an insurance NFC card to him/her. If users already have a NFC card, they can use the website to renew current contract.*
 - *Check card information: ...*
 - *...*
- *Insurance card printer (mobile app):*
 - *Simulating NFC card printer: staff can print NFC card....*
- *...*

5.2 Advantages and disadvantages

<Phân tích ưu và khuyết điểm của giải pháp của nhóm đề xuất>

- Advantages:

<Liệt kê ưu điểm>

Ví dụ

- *The interaction between the insured one and the insurance company: the insured one and the company now are easier to communicate through the website when each person has an account.*

- ...

- Disadvantages:

<Liệt kê khuyết điểm>

Ví dụ

- *Currently not consistent with the law of Vietnam about insurance card issues.*

- ...

- **Notes:** Có thể phân tích điểm vượt trội hay khuyết điểm của giải pháp sẽ được thực hiện so với hệ thống đang có sẵn

6. Functional Requirements

Function requirements of the system are listed as below:

<Liệt kê các tính năng theo gom nhóm cụ thể: tìm kiếm, gợi ý, quản lý tài khoản>

Ví dụ

- *User component:*

- *New contract request*

- ...

- *Staff component*

- *Create new contracts*

- ...

- ...

7. Role and Responsibility

Liệt kê danh sách và vai trò theo mô hình phần mềm mà nhóm sẽ lựa chọn trong phần tiếp theo. Các thành phần nên đặc tả vào bảng để dễ nhìn

Ví dụ

No	Full Name	Role	Position	Contact
1	Kiều Trọng Khánh	Project Manager	Supervisor	khanhkt@fpt.edu.vn
2	Đinh Quang Trung	Developer	Leader	trungdqse60994@fpt.edu.vn
3	Nguyễn Hữu Phúc	Developer	Member	phucnhse60749@fpt.edu.vn
4	Phùng Quang Minh Trí	Developer	Member	triphqmse60746@fpt.edu.vn
5	Nguyễn Chí Kha	Developer	Member	khanc60351@fpt.edu.vn

Table 1: Roles and Responsibilities

B. Report No.2 Software Project Management Plan

1. Problem Definition

1.1 Name of this Capstone Project

- <Tên đề tài kèm mã>

Ví dụ

- *Official name:* Insurance Card
- *Vietnamese name:* Thẻ bảo hiểm
- *Abbreviation:* MIC

1.2 Problem Abstract

<Tổng quan về giải pháp của nhóm sẽ thực hiện với dự án, **tuyệt đối không sao chép ở phần 1** mà diễn giải lại cho phù hợp dưới góc nhìn của quản trị dự án theo khía cạnh đang lên kế hoạch cho giải pháp mà đã đề ra trong phần introduction>

Ví dụ

As current in Viet Nam customer use Motor Insurance Certificate Paper when they get problems with their motor. Using the Motor Insurance Certificate Paper is inconvenient, for example, it can be wet or to insert or update the information in to insurance certificate paper is complicate. So we use the NFC card we call it is insurance card to handle it. Insurance company supplies the NFC card when the customer buy insurance. The card contains the information of customer, if the customer joins with many insurance service they just use only one card.

We provide a software to check the validation of card, the expired date of card and insurance services that customer joined. We also provide other advantages that can help save time and costs in some process of company. For example, the software can automatic extend the insurance service, update the information about accidents of motor. In addition, we also provide a system software to manage the information of customer via some insurance card we bought, this software will deploy at insurance company

1.3 Project Overview

1.3.1 Current Situation

- <Phân tích vấn đề khi đề tài được triển khai, các ưu/khuyết điểm của đề tài: có thể là khảo sát thực tế từ người dùng khi lấy yêu cầu, hoặc các thống kê mà nhóm đã nghiên cứu, so sánh với quy trình của hệ thống hiện tại>

Ví dụ

Below are the problems encountered in this project:

- **Security:** currently, there is few possible problems encountered with NFC tags, as NFC tags can be counterfeited, attacked during data transmission caused data loss, data corruption.

- **Server crash:** all the needed data is stored in the server. So if server crash, all the devices cannot get card information.
- **Absence of team members:** team members can get sick or unexpected problems.
- **Currently not consistent with the law of Vietnam about insurance card issues.**
- ...

1.3.2 The Proposed System

<Trình bày giải pháp cụ thể của nhóm tổng quát từ cách tiếp cận đến công nghệ, qui trình sẽ áp dụng >

Ví dụ

According to the technology researches, we found out that the NFC technology is very capable of resolve the current situations in insurance companies. We can use a feature of NFC tag to resolve the security problem from NFC card. The basic idea is to use a NFC tag (or NFC “card”) which contains a unique card ID as an insurance card instead of paper card currently.

We also build a high available web server to maintain the main system to work 24/7 to make sure that if mobile applications need access to the information there will be always available.

We assign responsibility in vertical to make sure if any member in this problem cannot continue to work in our team there will be the least harmful to the project processes.

To resolve problem from Vietnam laws of insurance for motorbike, we support the insurance companies to propose new law sections about using technology devices to work with insurance certificate paper to make our system work legally in current situation.

Our system includes three main subsystems: an online website for company’s staffs, a mobile application for police officers and a mobile application to simulate the card printer

1.3.2.1 Web Site

- <Liệt kê tính năng chính theo gom nhóm chức năng/vai trò>

Ví dụ

Website is a common communication portal for insurance company’s staffs and users (customers). Website provide following features:

- **For users (customers):**
 - Users can register new insurance card with online payment.
 - ...
- **For staffs:**
 - ...

Beside above, website system also provides an API interface for two mobile applications to retrieve, update data from mobile applications.

1.3.2.2 Mobile Application

- <Liệt kê tính năng chính theo gom nhóm chức năng/vai trò>

Ví dụ

This is a simulating application to simulate the work of Card Printer. In reality, the company who deploy this system need to have a NFC Card Printer to write information about the insurance company and customer information into an NFC card. However, our system currently only support this as a simulating application. This application is used by company's staffs and do followings:

- *Retrieves insurance contract information and write data to a physical NFC card.*
- *...*

...

1.3.3 Boundaries of the System

<Xác định nội dung cụ thể phạm vi đề tài sẽ thực hiện - fix scope để tập trung thuyết minh trong các phần tiếp theo - nội dung này căn cứ vào các nội dung trong phiếu nhận đề tài>

Ví dụ

This section supposes that the government laws in local area supports the method of using NFC cards as insurance cards, and accept NFC insurance cards are legal.

- *Every company who has Information System infrastructure can deploy this system.*
- *Companies who deployed this system has to equip enough devices for the system to run, includes:*
 - *Computer system with internet connection.*
 - *...*

1.3.4 Future Plans

<Mô tả kế hoạch phát triển của đề tài để từ đó trong thiết kế từ conceptual, erd, đến mô hình phát triển phần mềm và thiết kế kiến trúc phải được chuẩn bị để có thể mở rộng trong tương lai để chứng tỏ khả năng lý tưởng của đề tài>

Ví dụ

Current system only can deploy to a company which provides single service: motor insurance card, we call this is the Isolated Single Service Model. We design the system to make it easy to scale to 2 bigger models:

- *Isolated Multiple Service Model: system can be deployed to one company which provide multiple insurance services such as motor insurance, health insurance, assets insurance...*
- *Distributed Multiple Service Model:.....*
- *...*

1.3.5 Development Environment

1.3.5.1 Hardware requirements <Yêu cầu phần cứng>

<Mô tả các yêu cầu phần cứng mà nhóm sẽ thực hiện testing và triển khai cho sản phẩm. Nên mô tả dưới dạng bảng>

Ví dụ:

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® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer Memory	1GB RAM	2GB or more

Table 2: Hardware Requirement for Server

...

1.3.5.2 Software requirements <Yêu cầu phần mềm>

Mô tả các yêu cầu phần mềm mà nhóm sẽ áp dụng trong phát triển sản phẩm. Nên mô tả dưới dạng bảng biểu

Ví dụ:

Software	Name / Version	Description
Operating system	Window Server 2008	Operating system and platform for development
Environment	Java EE 6	Specification for developing web application
Modeling tool	Microsoft Visio 2013	Used to implement website and web service
IDE	Netbeans 7.2.1, IntelliJ IDEA 14.1	Programming tools
DBMS	MySQL 5.6	Used to create & manage the database for system
Source control	TortoiseSVN 1.8.11	Used for source control
Web browser	Chrome 42 or above	Testing browser

2. Project organization

2.1 Software Process Model

<Mô tả về mô hình phát triển mà nhóm lựa chọn, có ảnh hưởng tới mục 3 Project management plan>

<Các hình vẽ về mô hình và nội dung mô tả cần phải được reference>

<Giải thích lý do lựa chọn mô hình dựa trên các nội dung liên quan đến đề tài và những nội dung đã được đề ra trong phần introduction, project plan>

Ví dụ

This project is developed under waterfall model. We apply customized waterfall model to capable with current situation in our team. We choose this model because the following reasons:

- *Based on researches and clarify Vietnam laws of insurance for motorbike and current system in insurance companies, the requirements of this project are stable, clear, fixed and well understood by all team members.*
- *This project use NFC technology, ...*
- *...*

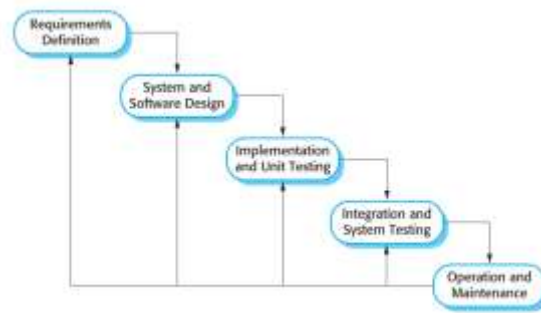


Figure 1: Waterfall model

Reference: Page 30, chapter 2, Software process model, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

2.2 Roles and responsibilities <Bảng phân chia vai trò>

<Mô tả vai trò của từng thành viên trong dự án và công việc phân công cụ thể cho từng thành viên. Nên mô tả dưới dạng bảng biểu>

Ví dụ

No	Full name	Role in Group	Responsibilities
1	Kiều Trọng Khánh	Project manager	<ul style="list-style-type: none"> Specify user requirement Control the development process Give out technique and business analysis support
2	Trần Nguyễn Đăng Khoa	Team Leader, BA, DEV, Tester	<ul style="list-style-type: none"> Managing process Designing database Clarifying requirements Prepare documents GUI Design Create test plan Coding Testing
...			<ul style="list-style-type: none">

Table 3: Roles and Responsibilities Details

2.3 Tools and Techniques

<Các công cụ sử dụng: chú ý ghi rõ phiên bản. Nên mô tả dưới dạng bảng biểu>

Ví dụ

Tool / Technique	Name / version
Frontend	HTML, CSS, JavaScript, jQuery, Bootstrap
Backend	JavaEE, Servlet, JSP, Hibernate
...	

3. Project Management Plan

3.1 Software development life cycle

<Mô tả cụ thể các công việc sẽ làm kèm theo phân bổ tài nguyên và đánh giá rủi ro. Chú ý các phase phải phù hợp với mục 2.1 Software Process Model ở trên. Nên mô tả dưới dạng bảng biểu và dùng trang giấy ngang để trình bày cho rõ ràng>

Ví dụ

<i>Phase</i>	<i>Description</i>	<i>Deliverables</i>	<i>Resource needed</i>	<i>Dependencies and Constrains</i>	<i>Risks</i>
Requirement Analysis	- Collect requirements from customer. - Identify and clarify requirements for the system in general.	- Introduction of proposed system. - Software requirement specification. - Project Task Plan. - Prototypes	20 man-days	N/A	- Missing requirement - Unclear scope of project - Lack of member share of understand
Design	- Architecture design for the system - Detail design using top-down break down - Choose Architecture style	- Software Design Document - Base code structure - Technology notes	20 man-days	Depend on "Requirement Analysis"	- Lack of experience. - Not fulfil requirement.
...					

Table 4: Software Development Life Cycle Detail

3.2 Phase Detail

<Mô tả cụ thể công việc trong các giai đoạn có chỉ định thành viên thực hiện tương ứng nội dung mô tả trong phần 3.1. Nên mô tả dưới dạng bảng biểu và dùng trang giấy ngang để trình bày cho rõ ràng>

Ví dụ

3.2.1 Phase 1: Requirement Analysis

<i>Task</i>	<i>Description</i>	<i>Author</i>
1. Collect requirements	Find which systems currently provide similar service, their strengths and weakness.	KhoaTND, HuyDN, TanNH
2. Identify and clarify main functions.	Define which main functions system should provide.	KhoaTND, HuyDN, TanNH
...		

Table 5: Phase 1: Requirement Analysis

3.3All Meeting Minutes

<Tạo folder trong svn và mỗi buổi gặp mặt giảng viên cần ghi nhận form meeting minute - và lưu trữ lại. Ghi đường dẫn trên svn vào phần này>

4. Coding Convention

<Mô tả tổng quan các code convention rule được nhóm áp dụng trong dự án được thực hiện, sau đó reference đến nơi tham chiếu

Không được phép reference không vì nếu như thế có thể xem như buộc trong project thực hiện phải áp dụng đúng như là nội dung references qui định>

C. Report No. 3 Software Requirement Specification

1. User Requirement Specification

<Liệt kê các yêu cầu về tính năng theo vai trò trong dự án>

Ví dụ

1.1 Guest Requirement

Guest is a person who doesn't have access to the system. Guest can use some functions in the system. To use all functions, guest must login. These are some functions guest can use:

- Register.
- Login.
- ...

1.2 Member Requirement

- ...

1.3 ...

2. System Requirement Specification

2.1 External Interface Requirement

2.1.1 User Interface

<Liệt kê các yêu cầu về trình bày cho người sử dụng>

Ví dụ

- General requirement for graphics user interface 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 will be taken into consideration:
 - UI for business web applications - Janko Jovanovic [Ref: <http://www.smashingmagazine.com/2010/02/25/designing-user-interfaces-for-business-web-applications/>]
 - Ten principles of effective web design – Vitaly Friedman [Ref: <http://www.smashingmagazine.com/2008/01/31/10-principles-of-effective-web-design/>]
 - Principles of mobile interface design – Jonathan Stark [Ref: <http://www.oreilly.com/pub/e/2144>]

2.1.2 Hardware Interface

<Liệt kê các yêu cầu phần cứng sử dụng trong dự án>

Ví dụ

- Smartphone with NFC support.

2.1.3 Software Interface

<Liệt kê các yêu cầu về phần mềm chú ý ghi rõ phiên bản cũng như kích thước màn hình>

Ví dụ

- *Web application: work with Firefox (v30 or above), Chromes (v14 or above), Internet Explorer (v10 or above) browse.*
- *Mobile application: Android operating system (v 4.0 or above).*

2.1.4 Communication Protocol

<Yêu cầu về giao tiếp giữa các thành phần trong ứng dụng>

Ví dụ

- *Use HTTP protocol 1.1 for communication between the web browser and the web server.*

2.2 System Overview Use Case

<Hình Overall Use case của hệ thống: chú ý sử dụng bộ kí hiệu phù hợp ý nghĩa và phiên bản UML sử dụng để ghi trong mô tả use case>

Ví dụ

Thông tin mô tả về đặc tả UML tham khảo tại <http://www.omg.org/spec/UML/2.0/>

Chú ý

- Các quan hệ giữa các use case và khi dùng **extend** phải ghi rõ **<extension point> và condition**
- Overview usecase phải thể hiện ràng buộc giữa các usecase trong hệ thống, tuyệt đối **không được liệt kê usecase**
- Nên sử dụng abstract usecase với nhóm chức năng có liên quan. Không nên sử dụng dạng **abstract usecase chỉ có một usecase, không sử dụng dạng abstract usecase có chứa thành phần abstract usecase**
- Khi mô tả usecase nên **chú ý tập trung chức năng, view** là các **thành phần phụ trợ (có thể nói là extend) không phải là chức năng chính** của hệ thống
- Cần phân biệt rõ **usecase là chức năng, qui trình**. Usecase **không phải là màn hình**, hay các **bước - step - trong quá trình xử lý**

Ví dụ

Figure 2: System Overview Use Case

2.3 List of Use Case

<Đặc tả chi tiết Use case theo từng role>

<Tách nhỏ thành phần usecase trong overview thành từng nhóm theo vai trò actor trong hệ thống đã được phân tích. Hình vẽ phải bao gồm luôn các usecase có quan hệ>

Ví dụ

2.3.1 <Guest>Overview Use Case

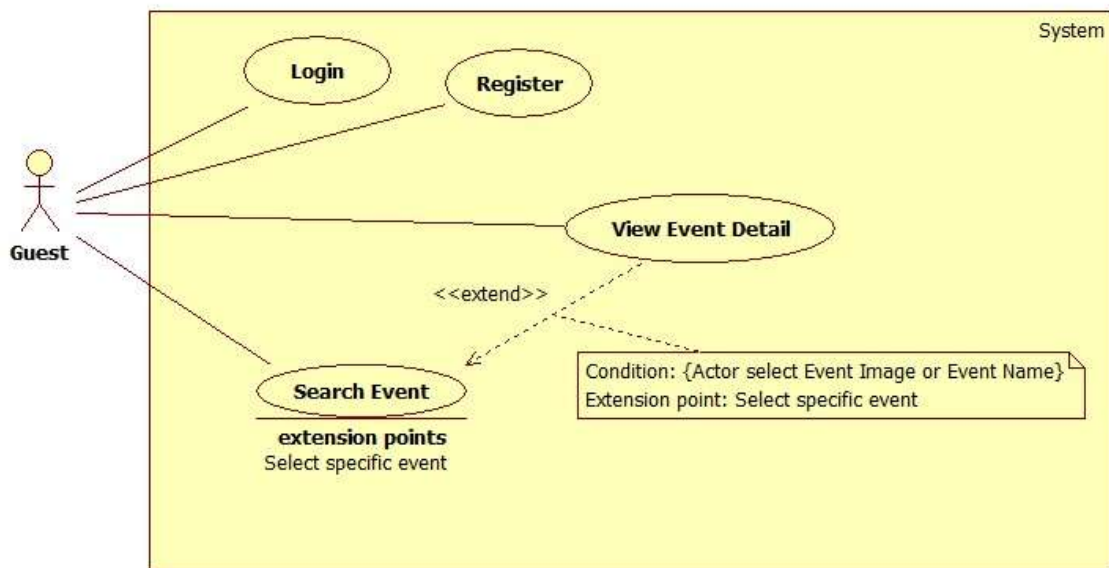


Figure 3: <Guest> Overview Use Case

<Tách riêng từng usecase để đặc tả trong usecase specification, lưu ý nếu có quan hệ thì phải vẽ hình có luôn quan hệ>

Ví dụ

2.3.1.1 <Guest> Register Use Case Diagram

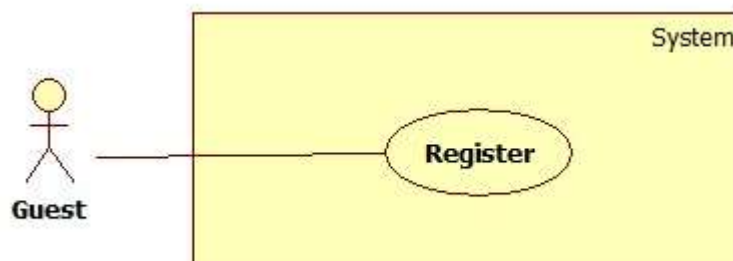


Figure 4: <Guest>Register Use Case Specification

GuideLine: Đây là giai đoạn **lấy requirement** nên các mô tả phải được diễn đạt theo ngôn ngữ của khách hàng, **không phải là nơi mô tả màn hình giao diện khi ứng dụng đã hoàn tất**. Ngoài ra, đây chính là **nơi thể hiện rõ vai trò lấy requirement với phương pháp ethnography - observe** để chuẩn bị thông tin cho thiết kế và thực hiện sản phẩm. Các **nội dung trong phần này** chính là phần **thông tin để hình thành** nên các **thực thể trong conceptual diagram**

USE CASE – <UC number>												
Use Case No.	Đánh số UC	Use Case Version	2.0									
Use Case Name	Tên UC											
Author	Người thiết kế, hiện thực											
Date	Ngày viết	Priority	Mức độ quan trọng trong dự án. Core flow thì đánh là High và giảm dần đến Normal									
<p>Actor:</p> <ul style="list-style-type: none"> - <Actor sẽ thực hiện use case> <p>Summary:</p> <ul style="list-style-type: none"> - <Tóm tắt về tính năng của use case> <p>Goal:</p> <ul style="list-style-type: none"> - <Mục đích của use case: kết quả khi usecase kết thúc thành công> <p>Triggers:</p> <ul style="list-style-type: none"> - <Bước làm use case được kích hoạt> <p>Preconditions:</p> <ul style="list-style-type: none"> - <Xác định các ràng buộc phải đạt được trước khi chức năng được thực hiện, thông thường là role của actor, trạng thái yêu cầu của dữ liệu, các ràng buộc về toàn vẹn dữ liệu hay qui trình> - Ví dụ: để cancel một hóa đơn thì precondition là <ul style="list-style-type: none"> o <i>User phải là một customer</i> o <i>Hóa đơn vẫn đang trong tình trạng chưa hết thời hạn hủy của hệ thống là 3 ngày</i> <p>Post Conditions:</p> <ul style="list-style-type: none"> - <Trạng thái sau khi tiến hành bắt buộc phải có 2 trạng thái cho success và fail. Vì vậy khi ghi phải có đủ và phần fail bắt buộc xuất hiện trong exception scenario> - Success: Khi thành công thì tình trạng hệ thống thế nào đối với hệ thống và đối với người dùng - Fail: Khi có lỗi xảy ra thì hệ thống sẽ xử lý thế nào để đảm bảo usability cho người dùng và toàn vẹn dữ liệu cho hệ thống <p>Main Success Scenario: <Hướng xử lý chính của hệ thống></p> <table border="1"> <thead> <tr> <th>Step</th> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td>-</td> </tr> <tr> <td>2</td> <td></td> <td></td> </tr> </tbody> </table> <p>Alternative Scenario: <Hướng xử lý khác trong tình huống dữ liệu cụ thể như</p>				Step	Actor Action	System Response	1		-	2		
Step	Actor Action	System Response										
1		-										
2												

mệnh đề if hoặc lựa chọn khác của người dùng trong quá trình main flow được diễn ra>

No	Actor Action	System Response
1		

Exceptions: Gồm các tình huống xử lý ngoại lệ cũng như xử lý các exception do người dùng gây ra khi nhập liệu

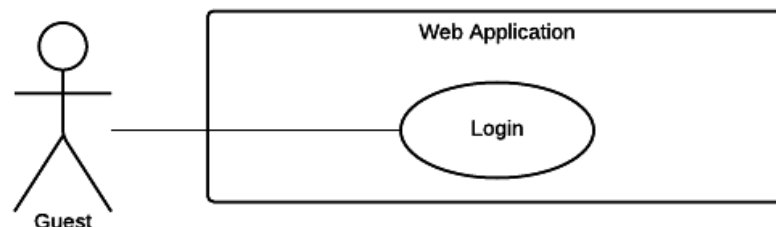
No	Actor Action	System Response

Relationships: Mỗi quan hệ với các Use case khác nếu có trong quá trình xử lý, tuy nhiên nó không phải là abstract usecase

Business Rules:

- Thành phần mô tả các yêu cầu về mặt nghiệp vụ của use case.
- Tất cả các giả định về nghiệp vụ nếu có phải được ghi vào
- Chú ý tới sự chuyển đổi về trạng thái của dữ liệu cũng phải được ghi tại đây
- Các định nghĩa cũng cần làm rõ (sản phẩm nổi bật, sản phẩm sắp có là sản phẩm thể nào trong hệ thống)
- Các ràng buộc dữ liệu dưới hệ thống, các rule liên quan đến toàn vẹn dữ liệu
- Các quy trình, activities, quá trình chuyển đổi trạng thái của hệ thống

Ví dụ



USE CASE – WG01			
Use Case No.	WG01	Use Case Version	2.0
Use Case Name	Login		
Author	TrungDQ		
Date	27/05/2015	Priority	Normal
Actor: <ul style="list-style-type: none"> - Guest Summary: <ul style="list-style-type: none"> - This use case allows guest to log in the system. Goal: <ul style="list-style-type: none"> - Guest can log in the system. Triggers: <ul style="list-style-type: none"> - Guest sends the login command. Preconditions: <ul style="list-style-type: none"> - N/A Post Conditions:			

- Success: Guest login the system.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest goes to login view.	System requires identity information from Guest: - Email or customer code: free text input - Password: free text input
2	Guest inputs information.	
3	Guest sends command to login to system	Guest will login system with their specific role [Alternative 1] [Exception 1]

Alternative Scenario:

Step	Actor Action	System Response
1	Guest enter wrong identity information.	Wrong identity information, System shows error message.

Exceptions:

Step	Actor Action	System Response
1		System show message the "System is busy" when the internet is lost

Relationships: N/A

Business Rules:

- Password are encrypted before being sent to server.
- After login to system, guest will be redirected to specific view based on their role on the system: staff or customer.
 - o If role is "Customer", the system will display to Customer view.
 - o If role is "Staff", the system will display to Staff Dashboard view.

Ví dụ

<Guest> Create new contract request

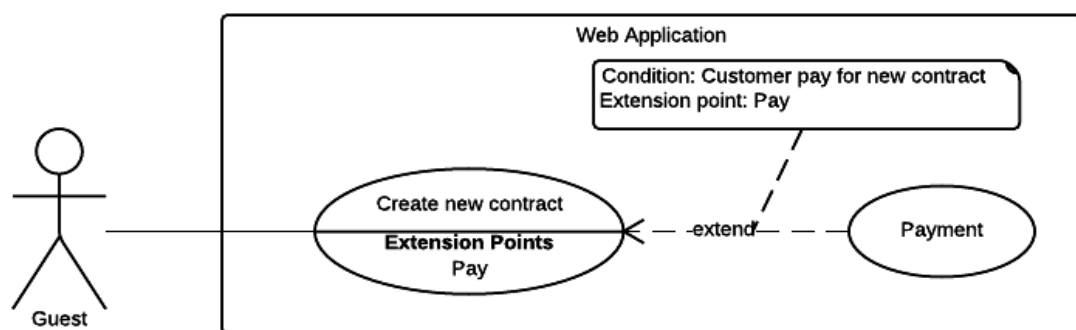


Figure 5 <Guest> Create new contract request

USE CASE – WG02			
Use Case No.	WG02	Use Case Version	2.0
Use Case Name	Create new contract request		
Author	TrungDQ		
Date	27/05/2015	Priority	Normal
Actor:			
- Guest			

Summary:

- This use case allows guest to create new contract request.

Goal:

- Guest can create new contract request.

Triggers:

- Guest sends command to create contract request.

Preconditions:

- N/A

Post Conditions:

- Success: New account and new contract will be created for guest.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest goes to new contract view.	System requires information from guest: Personal information <ul style="list-style-type: none">- Name: free text input, required, length 3 – 80.- Address: free text input, required, length 3 – 250.- Email: free text input, required, length 3 – 250.- Phone number: free text input, required, length 8 – 15.- Personal ID: free text input, length 8 – 15. Contract information (all information below are required) <ul style="list-style-type: none">- Contract's type: select one of the options.- Start date: date time input, required.- Contract term: text- Contract's fee: text Vehicle information <ul style="list-style-type: none">- Plate: free text input, required, length 4 – 15.- Brand: free text input, required, length 2 – 20.- Model code: free text input, length 2 – 20.- Vehicle type: free text input, length 2 – 20.- Color: free text input, length 2 – 20.- Engine: free text input, required, length 2 – 20.- Chassis: free text input, required, length 2 – 20.- Capacity: free text input, required, length 2 – 20.- Year of manufacture: number text input, value from 1900 to current year.- Weight: free text input, value from 1 – 1000, unit: kilogram- Seat capacity: free text input, value from 1 – 100. Security question <ul style="list-style-type: none">- Answer: free text input, required, length 1 – 10
2	Guest inputs information.	
3	Guest sends command to create new contract request.	System validate information, display contract details and request for confirmation. [Exception 1, 2, 3]
4	Guest sends command to create new contract request.	Add new account and new contract information to the system. Show successful message and ask user to process payment.
5	Guest sends command	Display new view let user select one of following

	to process payment	payment gateways: - PayPal payment gateway. - Direct payment. And show guest the fee: Contract's fee: text.
6	If user chooses PayPal gateway and sends confirm command. [Alternative 1]	Forward to PayPal payment view to process the payment.
7	User process the PayPal payment	If payment succeed: Show message created successful. [Exception 4]

Alternative Scenario:

No	Actor Action	System Response
1	If user chooses direct payment method	Show company address map.

Exceptions:

No	Actor Action	System Response
1	Guest sends command to create new contract request	System shows error message to ask user input missing required fields.
2	Guest's email is existed in the system	Show message to notify guest that their email is existed in the system.
3	Guest's vehicle plate is existed in the system	Show message to notify guest that their vehicle is existed in the system.
4	If payment failed	Show message to notify user that payment failed and the renew request has been aborted.

Relationships: Payment

Business Rules:

- New customer account and new contract will be created in the system with inputted information.
- The initial status of contract will be set to "Pending".
- When customer completed payment process:
 - + if the contract's start date has come, contract's status would change from "Pending" to "No Card".
 - + If start date is not come yet, the contract status is not changed.
- Staff will receive a notification about new contract request, they verify contract's information and issue a card for this contract, in this case, contract's status would change from "No Card" to "Ready".
- System must ensure has no duplicate customer or vehicle.
- An email contains customer code and password will be sent to user, user can use this information to login to the system later.
- Start date must not be earlier than the current date.
- Contract term is specified by the system.
- Contract types are loaded from system, contract type can be managed by system administrator.
- Contract price would be calculated from contract type and contract term.

Table 6 Use case WG02 - <Guest> Create new contract request

Ví dụ

<Customer> Cancel contract

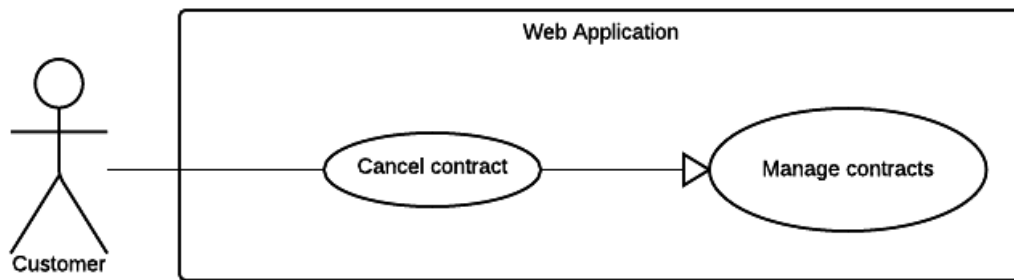


Figure 6 <Customer> Cancel contract

USE CASE – WC03			
Use Case No.	WC03	Use Case Version	2.0
Use Case Name	Cancel contract		
Author	TriPQM		
Date	27/05/2015	Priority	High
Actor: <ul style="list-style-type: none">- Customer.			
Summary: <ul style="list-style-type: none">- This use case helps user cancel their contract.			
Goal: <ul style="list-style-type: none">- Customer can cancel the contract.			
Triggers: <ul style="list-style-type: none">- Customer sends cancel contract request.			
Preconditions: <ul style="list-style-type: none">- User must login into the system with role Customer.- User's contract has not expired.- Customer's contract status must not be “Expired”, "Cancelled" or “Request cancel”.			
Post Conditions: <ul style="list-style-type: none">- Success: Send to the staff the cancel contract request.- Fail: Show error message.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	User goes to cancel contract view.	Display new view require user input some information: <ul style="list-style-type: none">- Reason to cancel the contract: can be optional selected from these values:<ul style="list-style-type: none">○ “Xe cơ giới bị thu hồi đăng ký và biển số theo quy định của pháp luật”○ “Xe cơ giới hết niên hạn sử dụng theo quy định của pháp luật”○ “Xe cơ giới bị mất được cơ quan công an xác nhận”○ “Xe cơ giới hỏng không sử dụng được hoặc bị phá hủy do tai nạn giao thông được cơ quan công an xác nhận”○ Other reason: free text input, required, length 1-250.	
2	User inputs information		
3	User sends cancel contract request command.	<ul style="list-style-type: none">- Change contract status.- Send request to the Staff. [Exception 1]	
Alternative Scenario: N/A			

Exceptions:

No	Actor Action	System Response
1	If user didn't check any reason to cancel contract	Show message to notify user that they have to choose the reason for cancel contract.

Relationships: N/A

Business Rules:

- Cancel contract request will be sent to the system with inputted information.
- System update status of the contract from “Pending”, “No Card” or “Ready” to “Request cancel”.
- A notification will be sent to staff after the process is completed.

Table 7 Use case WC03 - <Customer> Cancel contract

Ví dụ

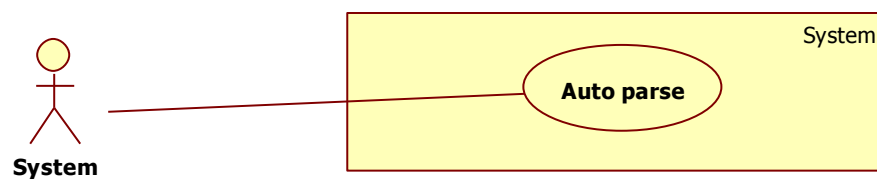


Figure 7: <System> Auto parse use case diagram
Use Case Specification

USE CASE – ARB08			
Use Case No.	ARB08	Use Case Version	2.0
Use Case Name	Auto parse		
Author	Pham Nguyen Bich Hien		
Date	30/05/2014	Priority	Normal
Actor: <ul style="list-style-type: none"> - System. 			
Summary: <ul style="list-style-type: none"> - System can parse resource automatically from many websites at specified time. 			
Goal: <ul style="list-style-type: none"> - Get resource from many websites. 			
Triggers: <ul style="list-style-type: none"> - The time hits configured time. 			
Preconditions: <ul style="list-style-type: none"> - Parse time has been configured. 			
Post Conditions: <ul style="list-style-type: none"> - Success: New data is inserted to storage. Log file is generated. - Fail: Nothing is changed in the storage. Log file is generated. 			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Server checks the current time. If it hits configured time, parse process starts.	<ul style="list-style-type: none"> - Send request to the parsed link. - Fetch data from the response based on the inputted XPath. 	

		<ul style="list-style-type: none">– Validate data [Exception 1].– If data is valid, insert to storage [Alternative 1].– Generate log file.
--	--	--

Alternative Scenario:

Step	Actor Action	System Response
1	Server checks the current time. If it hits configured time, parse process starts.	<ul style="list-style-type: none">– If fetched link resource is already in the storage, update its information.– Generate log file.

Exceptions:

No	Actor Action	System Response
1	Data is invalid.	<ul style="list-style-type: none">– Generate log file.

Relationships: N/A

Business Rules:

- If link resource exists in storage, do nothing.
- If link resource is not active, do nothing.
- Log file structure:

ARB LOG FILE

Tạo file lúc: {Created date}, {Created time}

STT	Link	Thời gian parse	Dạng dữ liệu	Tổng số sách nhận được	Insert thành công	Insert thất bại

Tổng thời gian parse dạng {Data type}: {Elapsed time}

Tổng thời gian parse: {Total elapsed time}

Tổng sản phẩm parse được: {Total parsed books}

-

Table 8: Auto parse use case specification table

3. Software System Attribute

<Mô tả non-functional requirement, các nội dung phải có dẫn chứng về việc đã đo đạc, có định lượng bằng các phương pháp, công cụ và phải hiểu về các nội dung đã ghi ra.>

3.1 Usability

3.2 Reliability

3.3 Availability

3.4 Security

3.5 Maintainability

3.6 Portability

3.7 Performance

.....

4. Conceptual Diagram

<Xác định các **thực thể** - **không cần có thuộc tính** - và **mối quan hệ** giữa chúng với nhau **thông qua các business rule, actor, các thành phần có mối quan hệ** để hình thành nên các thực thể thông qua các **mô tả trong usecase diagram và usecase specification** đã nêu ra ở trên>

Chú ý

- Chỉ sử dụng một tập kí hiệu và cần reference đến địa chỉ mô tả tập kí hiệu để sử dụng cho chính xác
- Các Diagram cần lớn rõ ràng, phải dàn trang cho phù hợp và nên dùng trang A3 để in
- Các thành phần trong diagram phải được thể hiện thông qua dictionary

Data Dictionary <Đặc tả các thực thể có trong hình>

Entity Data dictionary: describe content of all entities	
Entity Name	Description

Ví dụ

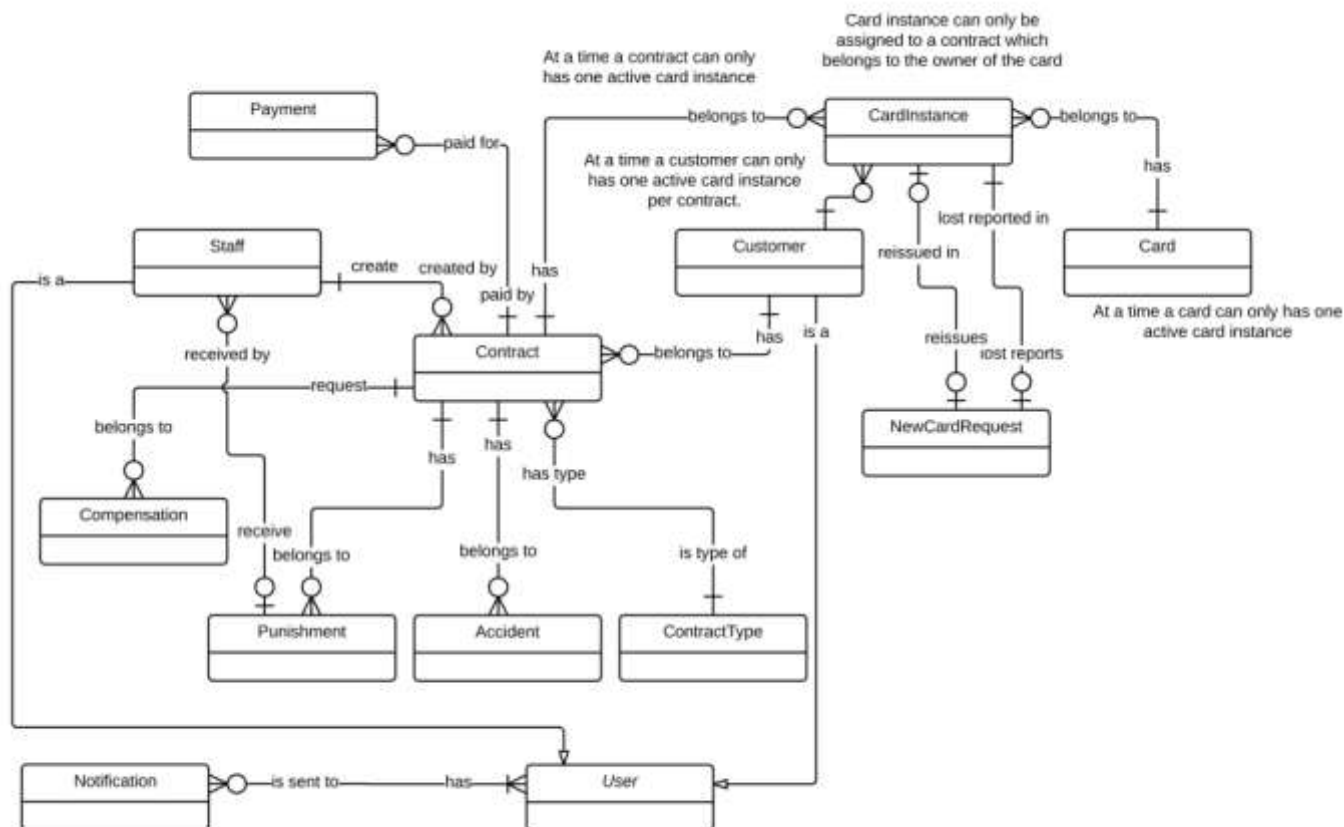


Figure 8 Conceptual diagram

Data Dictionary

<i>Entity Data dictionary: describe all content of all entities</i>	
<i>Entity Name</i>	<i>Description</i>
<i>User</i>	<i>Abstract entity describes a user in system</i>
<i>Customer</i>	<i>Contain the customer information.</i>
<i>Contract</i>	<i>Contain the contract information.</i>
<i>Card</i>	<i>Contain the card information</i>
<i>CardInstance</i>	<i>Represent a card assigned to a contract</i>
<i>Payment</i>	<i>Contain the payment information.</i>
<i>Staff</i>	<i>Contain the staff information.</i>
<i>Compensation</i>	<i>Contain the compensation information.</i>
<i>Punishment</i>	<i>Contain the punishment information.</i>
<i>Accident</i>	<i>Contain the accident information.</i>
<i>ContractType</i>	<i>Contain the contract type information.</i>
<i>NewCardRequest</i>	<i>Contain the new card request information.</i>
<i>Notification</i>	<i>Contain the notification information</i>

Table 9 Conceptual Diagram Data Dictionary

D.Report No. 4 Software Design Description

1. Design Overview

<Nội dung này tham khảo và có thể giữ nguyên và chỉ thay thế các phần phù hợp với đề án của nhóm. Nhóm có thể viết lại cho hay hơn>

- *This document describes the technical and user interface design of **MSSC System**. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.*
- *The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.*
- *The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.*
- *The database design describes the relationships between entities and details of each entity.*
- *Document overview:*
 - *Section 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 which includes class diagram, class explanation, and sequence diagram to details the application functions.*
 - *Section 5: describe screens design.*
 - *Section 6: describe a fully attributed ERD.*
 - *Section 7: describe algorithms.*

2. System Architectural Design

<Kiến trúc hệ thống mà nhóm xây dựng: sử dụng các pattern và reference đến nội dung và xem xét lựa chọn các diagram mang đầy đủ nội dung như concept, không sao chép, vay mượn và chế kí hiệu. Nếu dùng kí hiệu ngoài UML thì ghi chú giải kí hiệu ngay cạnh hình vẽ.>

<Mô tả kiến trúc của từng thành phần trong ứng dụng nếu có.>

Ví dụ

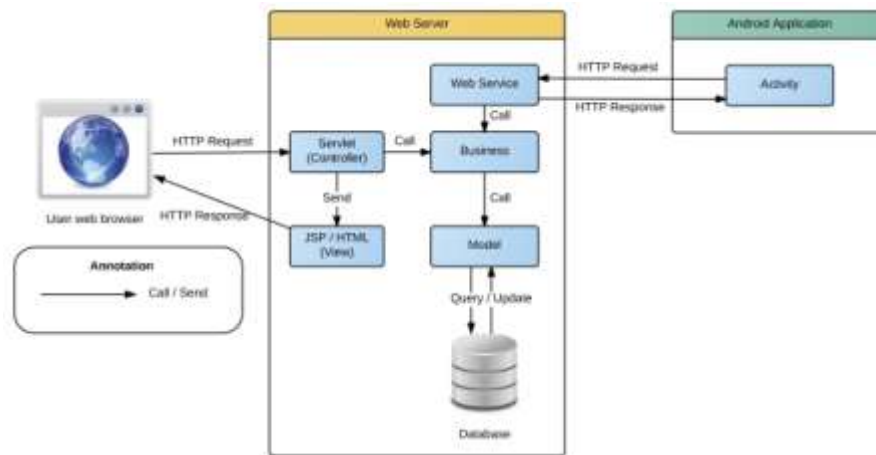


Figure 9 System architecture design

This diagram is referenced and modified from an original concept from: Chapter 6 Architecture Design, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

2.1 Web application architecture description

<Giải thích lý do tại sao lựa chọn mô hình này dựa trên SRS, Introduction, và project plan đã nêu ra ở các phần trên>

<Mô tả các thành phần của kiến trúc theo dạng bảng, và sự tương tác giữa các thành phần theo kiến trúc.>

Ví dụ

In Web Application, the system is developed under J2EE MVC architecture style. We choose this architecture for Web application because of following advantages:

- *Web app contains a Web service (public API for mobile app), with MVC architecture, we can separate business code with Controller and View, so we can use the business code in web service without repeat the code.*
- ...

This project follows MVC architecture with following components:

- *Servlet (Controller) is the parts of the application that acts like event handler to handles user interaction. Typically, controller read data from a request and calls appropriate Business's method then selects view to return to user.*
- ...

2.2 ...

3. Component Diagram

<Thể hiện việc chia hệ thống thành các component. Nội dung này dựa trên kiến trúc đã đề ra ở phần trên để chia cho phù hợp và đúng mô hình>

Ghi chú: Xem lại bộ quy ước kí hiệu của UML 2.0 trước khi vẽ các mối quan hệ cũng như hiểu rõ thiết kế để vẽ chính xác. Nếu tool không phù hợp thì nhóm nên dùng Paint để vẽ

<Mô tả từng thành phần trong hình vẽ theo bảng biểu bên dưới.>

Component dictionary: describe component	
Component Name	Description

Ví dụ

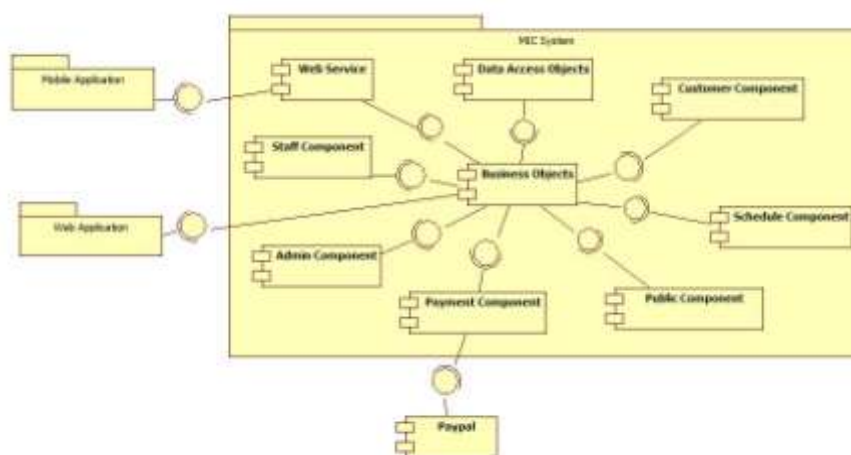


Figure 10 Component Diagram

Component Dictionary: Describes components	
Web Application	Web application package: View, Controller
Mobile Application	Mobile application package
PayPal	Handle payment process with PayPal API
Payment Component	Component to handle payment process
Web Service	Provide API for mobile applications to interact with the system.
Staff Component	Component to handle staff activities in the system
Customer Component	Component to handle customer activities in the system
Public Component	Component to handle guest activities in the system
Admin Component	Component to handle admin activities in the system
Schedule Component	Component to handle scheduler in the system
Business Objects	Common objects to handle domain business operations for each components
Data Access Objects	Component to handle interaction between the system and database

Table 10 Component Dictionary

4. Detailed Description

4.1 Class Diagram

<Hình thiết kế class diagram: tham khảo các mối quan hệ giữa các lớp trong đặc tả UML, nắm rõ về dependency, association, composition, aggregation, inheritance. Bên cạnh đó, cần xác định rõ cardinality giữa các quan hệ với nhau. Đây là dạng conceptual class diagram, do vậy, cần căn cứ trên conceptual diagram và nội dung xây dựng object cần thiết khi lập trình và xây dựng ứng dụng trong lúc viết chương trình>

<Mô tả từng thành phần class theo bảng biểu bên dưới.>

Class dictionary: describe Class	
Class Name	Description

Ví dụ

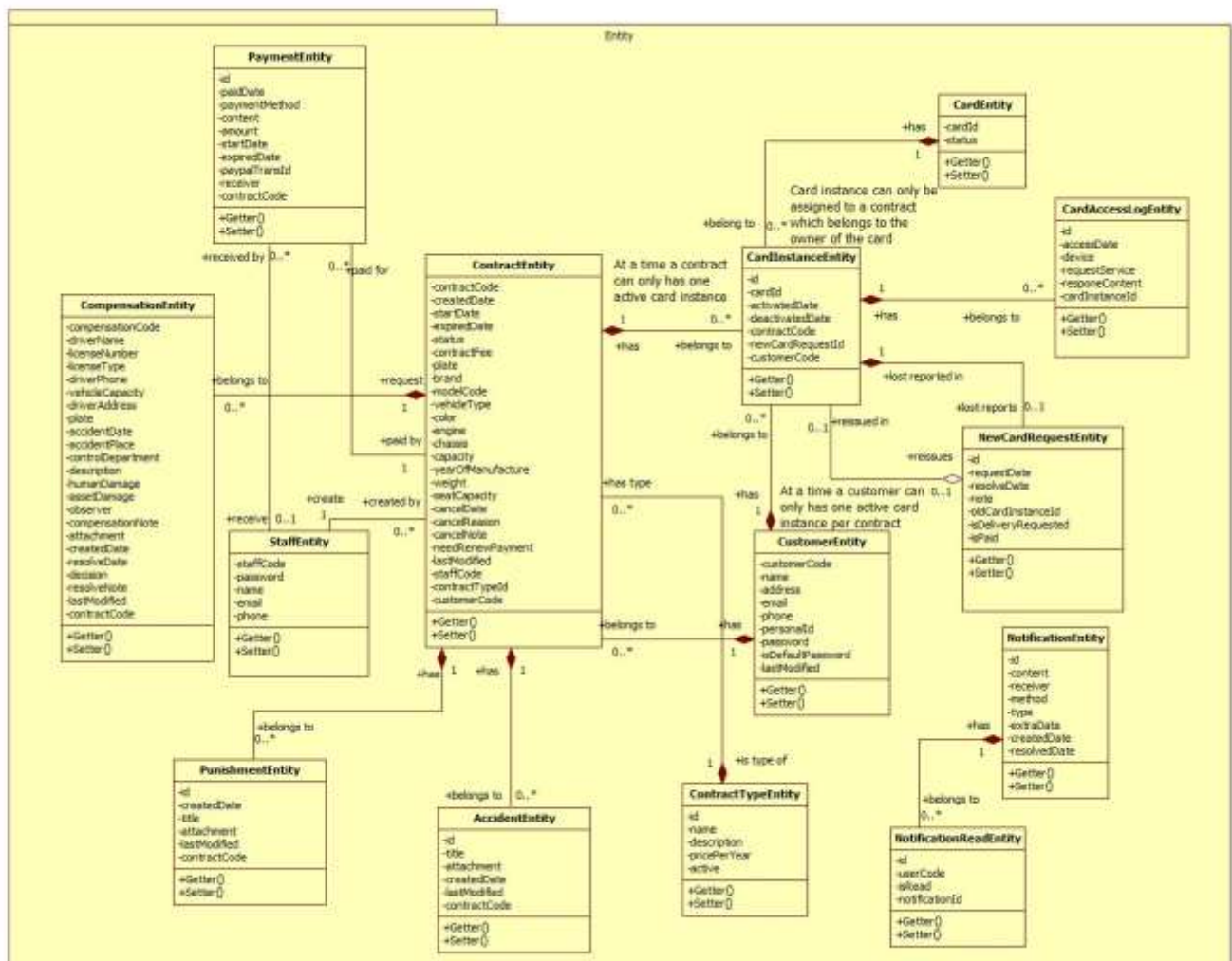


Figure 11 Class Diagram

Class dictionary: describe Class		
Class Name	Mapping column with Conceptual diagram	Description
PaymentEntity	Payment	Contain the payment information.
CardEntity	Card	Contain the card information.
CardInstanceEntity	CardInstance	Contain the card instance information
CustomerEntity	Customer	Contain the customer information.
ContractEntity	Contract	Contain the contract information.
StaffEntity	Staff	Contain the staff information.
CompensationEntity	Compensation	Contain the compensation information.
PunishmentEntity	Punishment	Contain the punishment information.
AccidentEntity	Accident	Contain the accident information.
ContractTypeEntity	ContractType	Contain the contract type information.
NewCardRequestEntity	NewCardRequest	Contain the new card request information.
CardAccessLogEntity	N/A	Not exist in conceptual diagram. But needed in class diagram to contain the card access log information.
NotificationEntity	N/A	Not exist in conceptual diagram. But needed in class diagram to contain the notification information.
NotificationReadEntity	N/A	Not exist in conceptual diagram. But needed in class diagram to know what notifications is read.

Table 11 Class dictionary

4.2 Class Diagram Explanation

<Mô tả các thành phần cụ thể cho các lớp đã được vẽ ra ở phần trên>

Ví dụ

4.2.1 Role

Attribute

Attribute	Type	Visibility	Description
RoleID	int	Private	Unique identifier of a role
Name	string	Private	Role name

Method

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

4.2.2 ...

4.3 Interaction Diagram

4.3.x Tên Interaction Diagram

<Sử dụng **sequence diagram** là chủ yếu để trình bày nội này. Sequence diagram cần kết hợp giữa các class đã trình bày ở trên kết hợp với các kiến trúc đã được thuyết minh để có mô hình phù hợp. Đối với ứng dụng **điện thoại di động** thì nên sử dụng **activity diagram**>

Summary: <Nên có phần tóm tắt trước diagram để trình bày về mục đích của diagram trước khi thể hiện hình vẽ>.

Ví dụ

4.3.1.1 Create new contract

Summary: this diagram show process of staff creates new contract

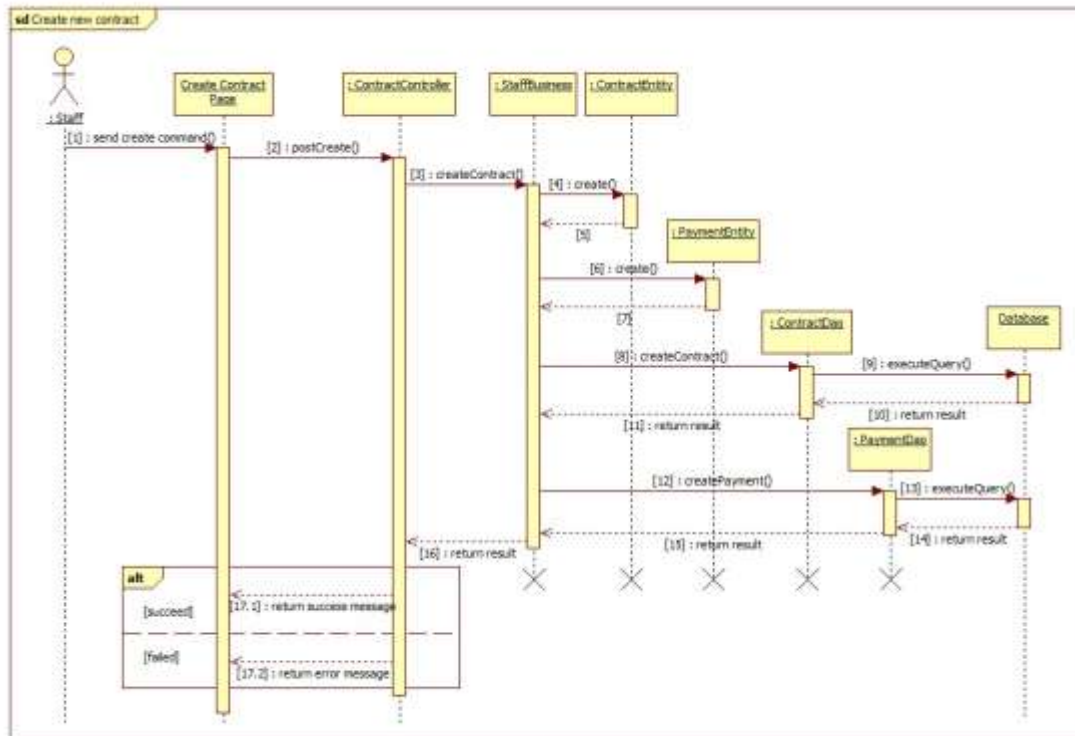


Figure 12 Sequence diagram - <Staff> Create new contract

4.3.1.2 <Member> View Friend List

Summary: This diagram shows how member views all contacts that include MSSC contacts and android cell phone contacts.

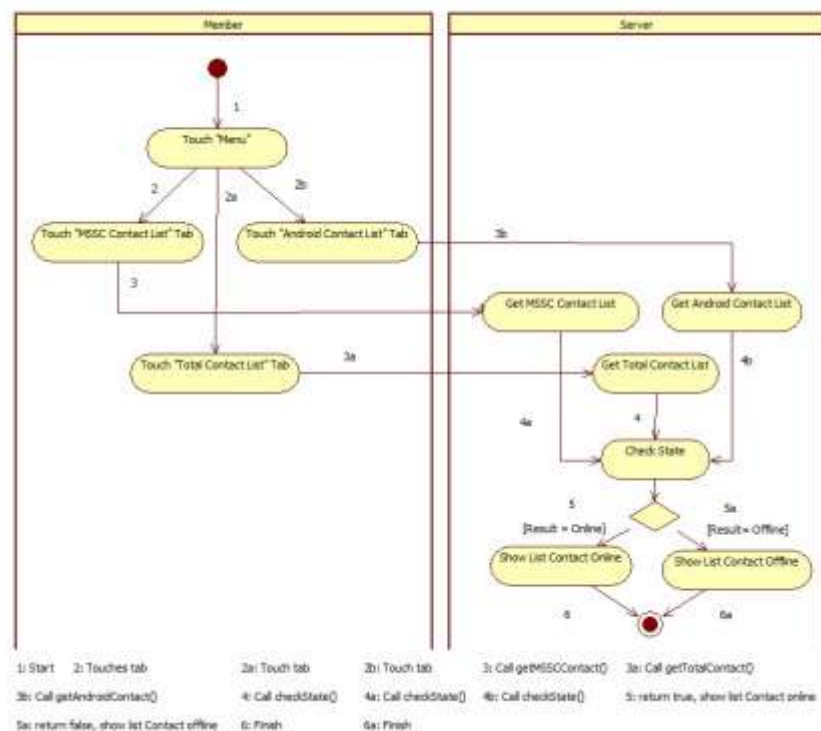


Figure 13: <Member> View Friend List

5. Interface

5.1 Component interface

<Mô tả các interface như của web service hay các signature của core flow được sử dụng trong hệ thống>

Nội dung được đặc tả theo dạng bảng như sau

Signature	Description	Input	Output	Output Format	Exception
Tên hàm	Mô tả mục đích	Tham số truyền	Kết xuất khi hàm xử lý xong	Kiểu dữ liệu	Xử lý lỗi

Ví dụ

Web Service Interface

Signature	Description	Input	Output	Output Format	Exception
<i>public ResponseObject getCheckConnection(R r)</i>	<i>Check server status</i>	<i>Request object r</i>	<i>Json Boolean the status of server</i>	<i>Boolean</i>	<i>JsonProcessingException</i>

...					
-----	--	--	--	--	--

5.2 User Interface Design

<Chụp và mô tả màn hình>.

Lưu ý phải đánh số đặc tả các control trên giao diện cùng với các thành phần trong ràng buộc

Ví dụ

5.3 Guest Interface Design

5.3.1 Login



Figure 14: Login

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Username	Fill user name	No	Yes	Textbox	String	N/A
2	Password	Fill password	No	Yes	Password	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Signin	Log-in into the system	N/A	Transfer to home page

6. Database Design

6.1 Entity relationship diagram (ERD)

<Thiết kế ERD. Được suy ra và hình thành từ conceptual diagram, class diagram và quá trình hình thành architectural>

6.2 Data Dictionary

<Mô tả về các thực thể>

Entity Data dictionary: describe content of all entities	
Entity Name	Description

<Mô tả các thành phần bên trong thực thể>

Entity name	Attributes	Description	Domain	Null
Tên	Thuộc tính 1 {PK}	Mô tả	Kiểu dữ liệu	Y/N

Table 12: Detail Data Dictionary

* Business integrity constraint:

<Mô tả các ràng buộc về toàn vẹn dữ liệu để đảm bảo nghiệp vụ>

7. Algorithms

<Các thành phần thuật toán - các giải pháp để giải quyết phần core flow mà nhóm đã áp dụng>

Chú ý

- Không nhất thiết phải là thuật toán nổi tiếng mà có thể là cách tổ chức dữ liệu cũng như giải thuật do nhóm đang thực hiện ở bên trong hệ thống: ghi rõ bản chất, phân tích về độ phức tạp, nếu tham khảo phải ghi rõ nguồn
- Cách giải quyết hay cách áp dụng các qui trình nghiệp vụ hay cách chuyển đổi bài toán khi làm bằng tay - chưa áp dụng máy tính và chương trình để cho thấy việc áp dụng giải bài toán hay giải quyết vấn đề rồi chuyển đổi cách đó sang thành chương trình máy tính

Ví dụ

7.1 Document Breakdown

7.1.1 Definition

Document breakdown is the way to break the document into many small parts. Each part has its own title and contents of it. And the final data has tree structure.

7.1.2 Define Problem

All content of document is quite difficult for manage so we must re-construct structure of document for using.

7.1.3 Solution

To solve this problem, we should follow these steps:

- Convert (save) document DOCX file as html type by using Microsoft Word save as Web Filtered.
- Import both html file and directory that includes all pictures of document.
- Using xpath to get data of html file as we need, include h1, h2, h3, ..., image, text content, ..
- Save them with structure as below:
-TitleA: contentA

---TitleA1: contentA1
-----TitleA1.1: contentA1.1
-----TitleA1.2: contentA1.2
---TitleA2: contentA2

7.1.4 Complexity

- In total, the complexity of this algorithm is N^4 .

7.1.5 Flowchart

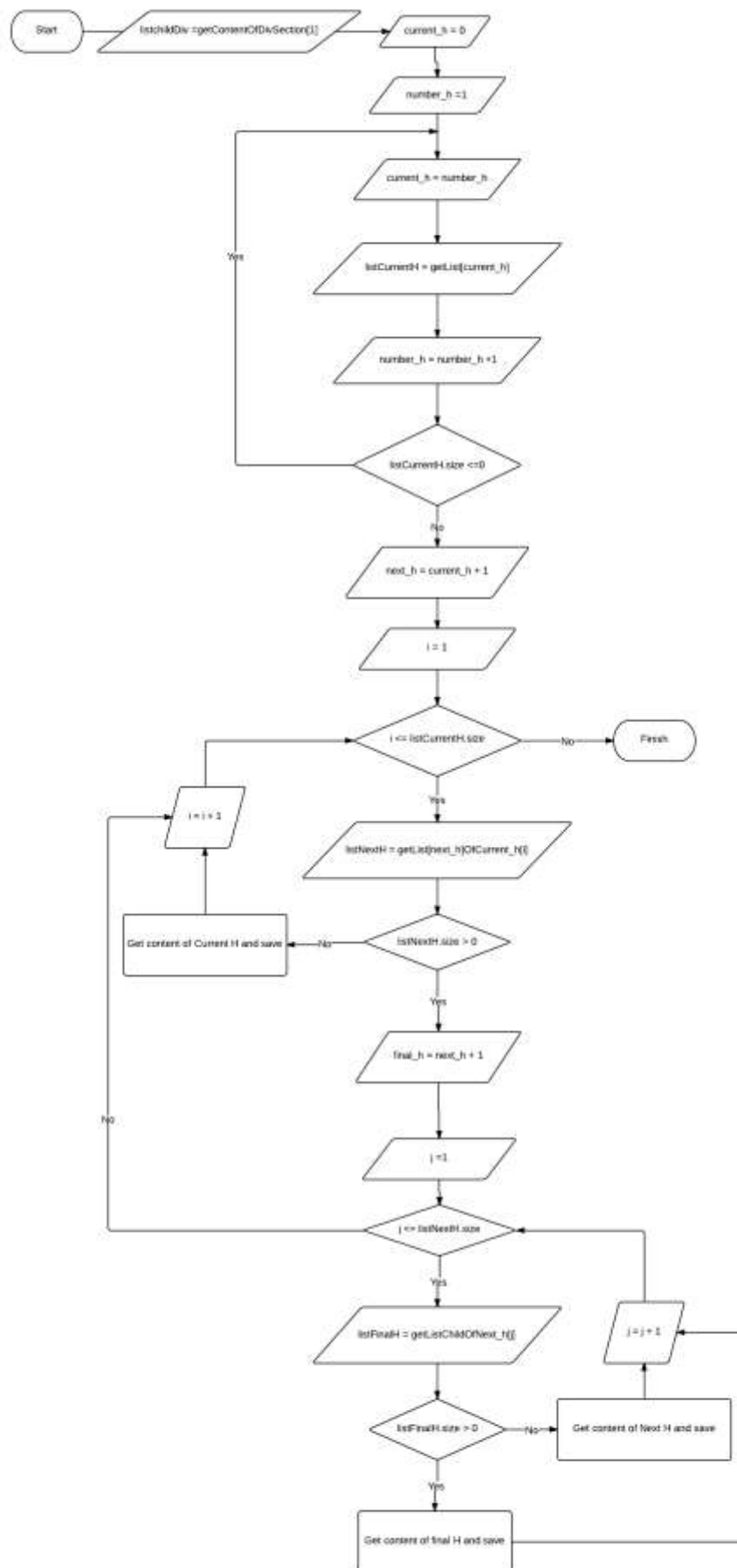


Figure 15: Breakdown document flow chart

7.2 String Comparison

7.2.1 Define Problem

Given two strings. Calculate their matching percent.

7.2.2 Requirement

- Robustness to changes of word order: two strings which contain the same words, but in a different order, should be recognised as being similar.
- Language independence: the algorithm should work not only in English, but in many different languages.

7.2.3 Solution

- If a string contains many words, break it into a list of words.
- For each word, we find out how many adjacent character pairs are contained in it.
- Create a function $\text{pairs}(s)$ which returns a list of adjacent character pairs of string s .
- Then, we use below formula to calculate matching percent.

$$\text{similarity}(s1, s2) = \frac{2 \times |\text{pairs}(s1) \cap \text{pairs}(s2)|}{|\text{pairs}(s1)| + |\text{pairs}(s2)|}$$

7.2.4 Example

Calculate the matching percent of 2 strings: France and French.

- Upper case 2 strings:
 - + France \rightarrow FRANCE.
 - + French \rightarrow FRENCH.
- Break string into list of adjacent character pairs:
 - + FRANCE $\rightarrow \{FR, RA, AN, NC, CE\}$
 - + FRENCH $\rightarrow \{FR, RE, EN, NC, CH\}$
- Calculate its matching percent.

$$\begin{aligned}\text{similarity}(\text{FRANCE}, \text{FRENCH}) &= \frac{2 \times |\{FR, NC\}|}{|\{FR, RA, AN, NC, CE\}| + |\{FR, RE, EN, NC, CH\}|} \\ &= \frac{2 \times 2}{5 + 5} = \frac{4}{10} = 0.4\end{aligned}$$

E. System Implementation & Test

1. Introduction

1.1 Overview

<Mô tả tổng quát mục đích test chủ yếu với thời gian và scope và số lượng nhân lực thì nhóm áp dụng phương pháp gì cho việc test>

Ví dụ

This section provides in detail all necessary information about implementation information and testing procedure of MSSC includes test plans, test cases, test result and risks estimations.

1.2 Test Approach

<Phương pháp kiểm thử của nhóm : black box, white box ...>

2. Database Relationship Diagram

2.1 Physical Diagram

<Vẽ database khi cài đặt vật lý trên các RDBMS: chú ý bố cục cũng như kích thước cho dễ đọc>

2.2 Data Dictionary

<Mô tả thành phần theo bảng biểu bên dưới>

Data dictionary: describe content of all tables	
Table Name	Description
Tên	Explanation

<Mô tả thành phần chi tiết>

Entity name	Attributes	Description	Domain	Null
Tên	Thuộc tính 1 {PK}	Mô tả	Kiểu dữ liệu	Y/N

Table 13: Attribute Data Dictionary

3. Performance Measures

<Cách nhóm ước lượng việc đo đạc hệ thống>

Ví dụ

3.1 Clustering Performance

- Clustering is performed by running K Mean Algorithm which has complexity of : $O(n * k * I * d)$
 - n : number of points
 - k : number of cluster
 - I : number of iteration
 - d : number of attributes (3)

Clustering take almost the time of process that we can ignore the time needed to load data from database, digitalize data.

The speed of clustering will vary and increase dramatically when n increase. The purpose of this project is not about optimizing K-Mean Algorithm so it is accepted to let the process run till it completes. Moreover, the clustering is designed to run by staff, wait time is acceptable.

4. Test Plan

<Đưa ra kế hoạch test>

Ví dụ

The purpose of this section is to verify and ensure that MSSC meets its design specification and other requirements from user. The following part will describe which features to be tested and which will not.

4.1 Features to be tested

<Tính năng sẽ kiểm thử>

4.2 Features not to be tested

<Tính năng sẽ không kiểm thử>

5. System Testing Test Case

<Nên vẽ các workflow tính năng sẽ test để dễ hình dung, chú ý dàn trang in ngang, chú ý đánh số, ngày tháng, kết quả, không sao chép>

Ví dụ

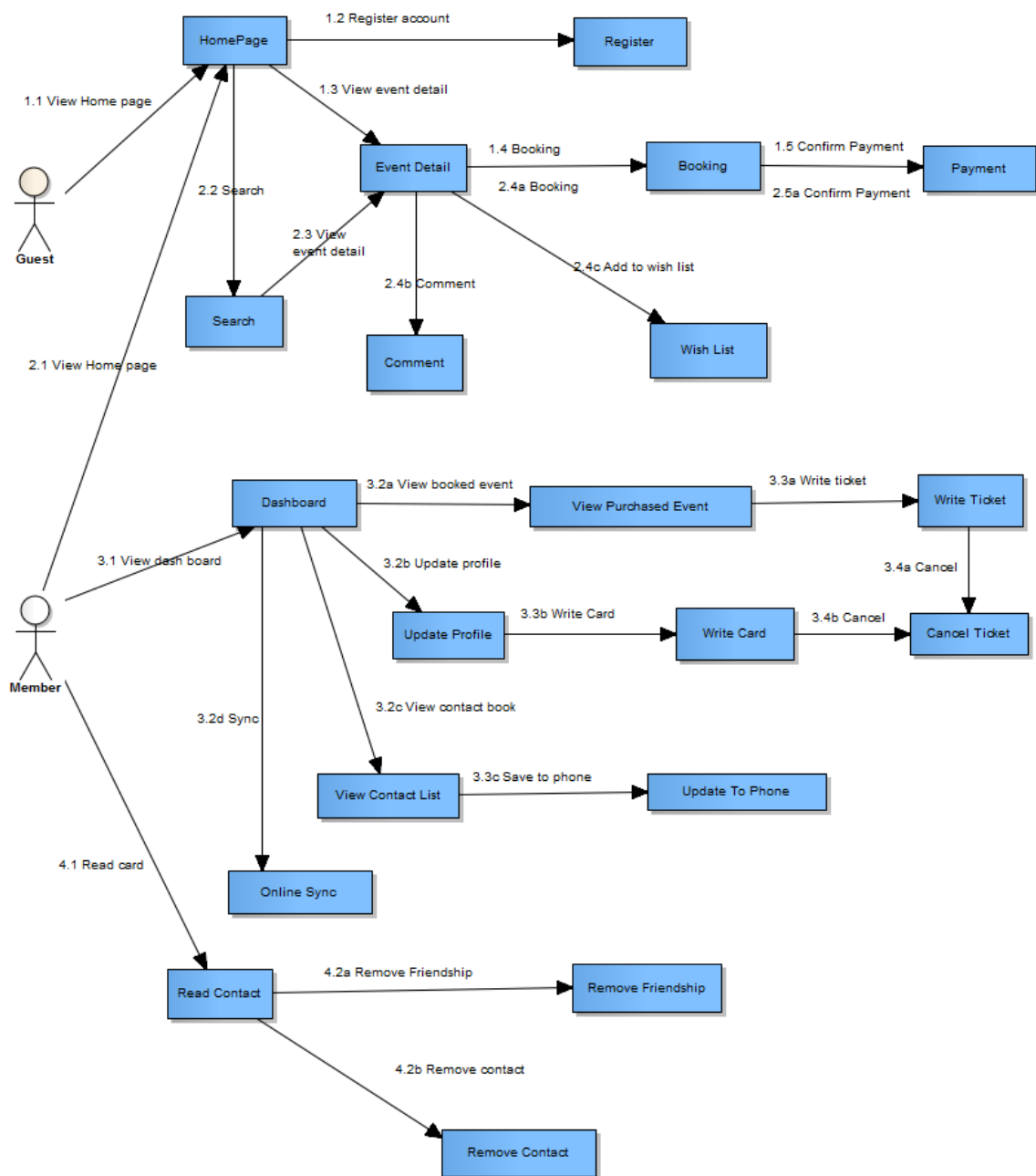


Figure 16: Guest, Member Core Flow

5.1 Guest Test Case

5.1.1 Search Event

<i>ID</i>	<i>Test Case Description</i>	<i>Test Case Procedure</i>	<i>Expected output</i>	<i>Inter-test Case Dependence</i>	<i>Result</i>	<i>Test Date</i>	<i>Note</i>

F. Software User's Manual

1. Installation Guide

1.1 Setting up environment at server side

The following software must be installed into the server machine:

1.1.1 Hardware requirements

<Yêu cầu phần cứng server, chú ý xem lại các report trước để nhất quán>

1.1.2 Software requirements

<Yêu cầu phần mềm server, chú ý xem lại các report trước để nhất quán>

1.2 Deployment at server side

<Mô tả quá trình triển khai lên server thực tế, gợi ý có thể gồm các bước sau, chú ý khi làm phải chụp hình cụ thể để hướng dẫn cũng như so sánh kết quả thành công>

1.2.1 Prepare deployment package

1.2.2 Configure Server before deploy

1.2.3 Deploy web application on server

1.3 Setting up the environment at client side

1.3.1 Setting up for computer

<Ghi rõ phiên bản tối thiểu để sử dụng>

2. User Guide

<Viết hướng dẫn sử dụng cho người dùng>

G. Appendix

<Các thành phần tham khảo của tài liệu chú ý tham khảo thêm cách ghi tại <http://www.khoahocviet.info/meresci/vi/meresci03d4.html>>