**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Warning system on parameters of the devices in the family**

|  |  |
| --- | --- |
| **Group 1** | |
| **Group members** | **Võ Phúc Hải – SE61582**  **Tôn Thất Minh Trí – SE61679**  **Nguyễn Phú Ngọc Trai – SE61060**  **Nguyễn Kim Cường – SE90180** |
| **Supervisor** | **Nguyễn Đức Lợi** |
| **Ext. Supervisor** | N/A |
| **Capstone Project**  **code** | **APMS** |

-Ho Chi Minh City, **September 8th 2017**-

*This page is intentionally left blank*

## Table of Contents

[Table of Contents 3](#_bookmark0)

[List of Tables 4](#_bookmark1)

[Definitions, Acronyms, and Abbreviations 6](#_bookmark2)

1. [Report No. 1 Introduction 7](#_bookmark3)
   1. [Project Information 7](#_bookmark4)
   2. [Introduction 7](#_bookmark5)
   3. [Current Situation 7](#_bookmark6)
   4. [Problem Definition 7](#_bookmark7)
   5. [Proposed Solution 8](#_bookmark8)
      1. [Feature functions 8](#_bookmark9)
      2. [Advantages and disadvantages 8](#_bookmark10)
   6. [Functional Requirements 9](#_bookmark11)
      1. Name Card Management 10
      2. Event 11
      3. Searching 12
      4. Suggestion 13
      5. User Management 14
   7. Role and Responsibility 15

## List of Tables

[Table 1: Roles and Responsibilities 9](#_bookmark13)

## 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 |

1. **Introduction**

## Project Information

* + - * Project name: **Warning system on parameters of the devices in the family**
      * Project Code: **APMS**
      * Product Type: **Internet of Things (IoT)**
      * Start Date: **8th September, 2017**
      * End Date: **11th December, 2017**

## Introduction

## The system will use sensors mounted on the devices in house and sends parameters of these devices over the internet to report for users via mobile devices. Our system will auto check devices and send an error message to users when problems, errors occur from electrical appliances in house.

## We build a system using IOT technology to transform the parameters data to remote sever through internet network. We build mobile app for checking and take warning from system.

## This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows.

## Current Situation

## Nowadays, the large systems and premium devices is equipped with the early warning system for devices/system's safety. The warning system includes sensors which measure parameters of device’s mainboard and environment.

## Next, the system compares sensor’s parameter with parameter which have been set. If any parameters exceeded the allowed level, the system would send warning message and suggested solutions to user’s phone via internet. Then, the user could decide what to do to protect the device and asset.

## A lot of warning system is widely used in developed countries, such as: America, Japan, Europe, … But in Vietnam, this system is not popular, most of them are imported expensive systems. Example: the warning system for Base Transceiver Station in telecommunication section, …

## Problem Definition

## The warning system usually apply in industrial area, still not familiar with home people. The current warning system has some disadvantages for family:

## Accessibility: the cost for install the system is very expensive. Usually heavy and not compact. There are not many retailers for this type of device.

## Connectivity: the system still using wire for connecting sensor to CCU. It is hard for maintenance.

## Handling problems: when device have problems, people have to manually check individual device for problems. Many people have to hire electrician or go to service center to fix the device.

## Proposed Solution

* The system follow status of devices is a necessary.
* Deploy more sensors on devices to increase management devices ability.
* Deploy environment sensors outside devices.
* Send warning for user when devices go wrong to have a solution to solve a problem fast.
* Wireless connection bring convenience for user.
* Set up a center system take parameters from sensors.
* Sending a warning to smartphone.

## Functional Requirements

* Measure component:
* Define device input’s voltage, current.
* Identify the electricity’s problem.
* Measure device’s temperature.
* Detect gas leakage by gas sensor.
* Passing signals over RF component:
* Signal of embedded sensor in devices.
* Data communication using RF carrier.
* Directional antennas and the influence of the environment.
* Passing data over internet component:
* Received data from device’s sensors.
* Data communication using Wi-Fi module.
* Transmission Control Protocol via Internet, UART protocol communication.
* API server at server to received data.
* Controlling warning component:
* Server send data to smartphone with internet connection.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full name** | **Role** | **Position** | **Contact** |
|  | Nguyễn Đức Lợi | Project manager | Supervisor | [loind@fpt.edu.vn](mailto:loind@fpt.edu.vn) |
|  | Võ Phúc Hải | Developer | Leader | [Haivpse61582@fpt.edu.vn](mailto:Haivpse61582@fpt.edu.vn) |
|  | Tôn Thất Minh Trí | Developer | Member | [Trittmse61679@fpt.edu.vn](mailto:Trittmse61679@fpt.edu.vn) |
|  | Nguyễn Phú Ngọc Trai | Developer | Member | [Trainpnse61060@fpt.edu.vn](mailto:Trainpnse61060@fpt.edu.vn) |
|  | Nguyễn Kim Cường | Developer | Member | [Cuongnkse90180@fpt.edu.vn](mailto:Cuongnkse90180@fpt.edu.vn) |

## Conclusion

* Research Configure and commands set microprocessor.
* Communication technique: How about RF interface; SPI, I2C, UART.
* Methods to transfer data from a micro processing over internet / RF.
* Research configure server for data received and data transmission.
* Research write android app for get warning from sensors.
* Use C/C++ embedded, Python, with main processor.
* Use software in design Schematic and PCB like OrCAD, Proteus or similar software.

1. **Software Project Management Plan**
2. **Problem Definition**
   1. **Name of this Capstone Project**

* Official name: Warning system on parameters of the device in the family
* Vietnamese: Hệ thống cảnh báo về thông số thiết bị trong gia đình
* Abbreviation: APMS
  1. **Problem Abstract**
* In Vietnam, the most of the electrical appliances in the home do not show any information about the device’s real status and environment. So, when something wrong happened, the user wouldn’t know why their devices is not working.
* The embedded sensors measured parameters of device’s environment such as voltage, current, … These parameters passed to server via wireless and internet. If something is abnormal, the specific warning would be sent to user’s cellphone. Then, they could decide what to do to protect devices and asset.
* This system has a central processor which collect sensor’s data from all distributed processors which embedded in devices to report detailed statistical information about device’s parameters in real-time.

### Project Overview

###### Current Situation

###### Advantages

###### Low cost.

###### Module easy to install.

###### Using easy.

###### Easy sent warning to user.

###### Not yet applied at home.

###### Disadvantages

###### The danger from high voltage electricity.

###### Support some type of devices.

###### Too many errors appear from house.

###### Too many devices to check.

###### The Proposed System

* + Warning system usually appear in high value and demand devices, especially in industrial area. Therefore, a properly warning device is essential for home living, thus, with a good price. Our proposed solution is to make a small device, with sensors built-in that attach to the household, communicate through RF modules. The RF modules provide wireless communication directly between Central Unit Control and sensor devices. There are also a monitor screen built in the Central Unit Control for easy to use. The CUC also send data through internet using Wi-fi to users’ phone.
  + The sensor unit device:
    - * Contain several sensors in a small package to detect changing parameters in the device it attach to.
      * Processing and send result to the Central Control Unit through RF module.
      * Self-checking device.
* The Central Control Unit:
  + - Central Control Unit have a small color screen to show the parameters from the sensor devices.
    - It also send information to user’s phone.
    - The mobile app
    - Beautiful and intuitive User Interface.
    - Receive data from the Central Control Unit through internet (Wi-fi)

###### Boundaries of the System

###### Advantages:

###### Easy to assemble.

###### Decreases time need to inspect device problems in the house.

###### Increase living quality.

###### Instant warning about the problem.

###### Easy for maintenance.

###### Reduce the risk of incidence happen in house.

###### Maximum and minimum the usage of electricity more effective and efficiency.

###### Disadvantages:

###### Intensive intervention in the system.

###### Don’t have web version for monitoring.

###### Future Plans

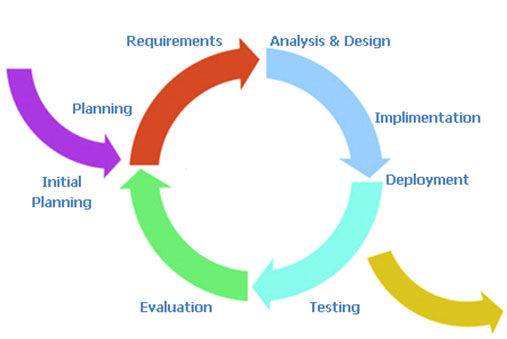
* There are no perfect solutions to problems, as well as there are no perfect systems. With the inexperience of our team members and the time constrains, our proposed solution and project contains many issues. Below are the problems encountered in this project:
* Hardware Knowledge: We are inexperienced with hardware. All the hardware components chosen to be used in this project is based on our familiar with them, or based on the shortest time we need to learn how to use them. So they are only the most appropriate, not the best choice for the project.
* Security: Currently, there is few possible problems encountered with RF, as RF is vulnerable to replay attack.
* Server crash: All the needed data for the app is stored in the server. So if server crash, all the devices cannot working properly.
* Our future plan is try to solve these problems one by one.

###### Development Environment

|  |  |
| --- | --- |
| **Component** | **Hardware** |
| Mainboard | Raspberry Pi 3 |
| Communication | USB Cable |
| Devices | Button |
| Motor | 5V |
| Receive/Transfer data | RF 433Mhz |
| Wifi | Port wifi on Raspberry Pi 3 |
| Power | DC |
| module sensor | Gas, energy, temperature… |
| Hard Drive | SD card |
| Processer | Arduino |

* + - 1. Hardware requirements
      2. Software requirements
* Windows XP/7/8/10: operating system for developing and deploying.
* SQL Server Express 2012: used to create and manage database for system.
* Visual Studio 2015: used to develop API.
* Arduino IDE: used to develop Arduino program.
* Proteus 8: used to drawing board with other hardware.
* Github & SourceTree: used for source control.
* StarUML: used to create models and diagrams.
* Slack: used for communication and meeting.
* C/C++: used for embedded module
* C#: used for web server
* Java: used for mobile application
* Python: programing for Raspberry Pi 3.

1. **Project organization**
   1. **Software Process Model**



* This project is developed under Iterative and incremental development model. We apply customized Iterative and incremental development model to capable with current situation in our team. We choose this model because of the following reasons:
* We are still inexperienced and by develop the system through iterations (repeated cycles) and incrementally (in small portions of time), we can learn from our mistakes and apply that knowledge on the next iteration.
* We are researching and developing the system at the same time, so using this model allow us more flexibility to adapt to changes.
* Working with embedded system hides a lot of problems that are unknown in the planning phase until it is too late. With Iterative and incremental development model, we test the system in small portion at a time, therefore reduce risk and build a feature rich and robust system.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Team roles | Responsibilities |
| 1 | Nguyễn Đức Lợi | Supervisors, Project Manager | * Specify user requirement * Advisor for ideas and solutions * Control the development process |
| 2 | Võ Phúc Hải | Team Leader,  BA, Developer,  Tester | * Managing process * Managing budge * Dividing tasks for * member * Create test plan * Clarifying requirement * Prepare document * Coding * Testing |
| 3 | Tôn Thất Minh Trí | Team Member  Developer,  Tester | * Create test plan * Clarifying requirement * Prepare document * Coding * Testing |
| 4 | Nguyễn Phú Ngọc Trai | Team Member  Developer,  Tester | * Create test plan * Clarifying requirement * Prepare document * Coding * Testing |
| 5 | Kim Cường | Team Member  Developer,  Tester | * Create test plan * Clarifying requirement * Prepare document * Coding * Testing |

|  |  |
| --- | --- |
| **Tools** | |
| Operating system | Windows OS, Linux/Ubuntu/Rasbian |
| Developing tool | Arduino IDE, Visual Studio, QT Creator, Notepad++ |
| Source control | Github |
| Communication tool | Slack |
| Model & Diagram | StarUML, Draw.io |
| **Programming language** | |
| Arduino | C++ |
| Raspberry | C++, Python |
| Application | C#, Python |

*Table 3: Roles and Responsibilities Details*

* 1. **Tools and Techniques**

1. **Project Management Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iteration** | **Scope** | **Evaluation** | **Activities** | **Estimated Duration** | **Assign Responsibilities** |
| Initial Iteration | Initial team workplace and identify project scope | A working team environment | * Set up environment * Set up Slack * Set up IDE | 5 days | HaiVP, TriTTM, TraiNPN, CuongNK |
| Iteration 1 | Identify boundaries of the system, planning software and hardware. Create a proof-of-concept prototype. | Report 1, Report 2 and a proof-ofconcept prototype | * Introduction document * Software and Hardware Project Management Plan document * Proof-ofconcept prototype | 15 days | HaiVP, TriTTM, TraiNPN, CuongNK |
| Iteration 2 | Produce an architectural prototype | Report 3, Report 4 and an architectural prototype | * Software and Hardware Requirement Specification document * Software and Hardware Design Description document * Architectural prototype | 15 days | HaiVP, TriTTM, TraiNPN, CuongNK |
| Iteration 3 | Build the product (up to beta release) | Report 5 and a working product (beta release) | * System Implementation and Test document * PCB * Application * Web API Server | 15 days | HaiVP, TriTTM, TraiNPN, CuongNK |
| Iteration 4 | Finish the product (full product release) | Report 6 and the completed product | * Software and Hardware User’s Manual document * Product demonstration model | 15 days | HaiVP, TriTTM, TraiNPN, CuongNK |
| Final Iteration | Prepare for Demo Day | Final Documentation, Presentation Slide | * Final Document * Mini Document * CD contains all source code * Presentation Slide | 5 days | HaiVP, TriTTM, TraiNPN, CuongNK |

*Table 4: Software Development Life Cycle Detail*

## Coding Convention

* C/C++ Convention:

C/C++: Using to develop program and solve algorithm on hardware.

* 1. Summary:
     1. Naming Convention:
        1. Using Pascal case for class name.
        2. Using Camel case for function, variable’s name.
     + Commenting Convention:
       1. Place the comment on the separate line with function.
       2. Place the comment at the end of the line, which has calculation formula.
* C# Convention
  1. C#: Using to develop Web API
  2. Summary:
     1. Naming Convention:
        1. Use Camel case for variable’s name.

Use Pascal case for class’s name, function’s name.

* + - 1. Global variable’s name must uppercase and separate by underscore.

1. **Software Requirement Specification**
   1. **User Requirement Specification**

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

Ví dụ

* 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. ***Member Requirement***
     + *...*

***1.3 ...***

* 1. **System Requirement Specification**
     1. **External Interface Requirement**
        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 businesss web applications - Janko Jovanovic [Ref:* [*http://www.smashingmagazine.com/2010/02/25/designing-user-interfaces-for- business-web-applications/*](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/*](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*](http://www.oreilly.com/pub/e/2144)*]*
      1. **Hardware Interface**

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

Ví dụ

* + - * *Smartphone with NFC support.*

###### 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).*

###### 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.*
    1. **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 usercase 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

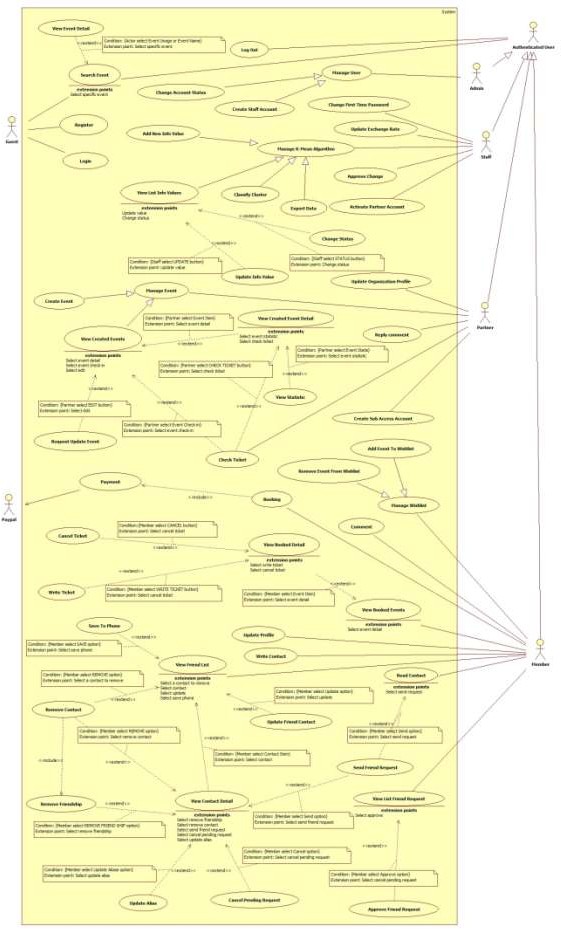


Figure 2: System Overview Use Case

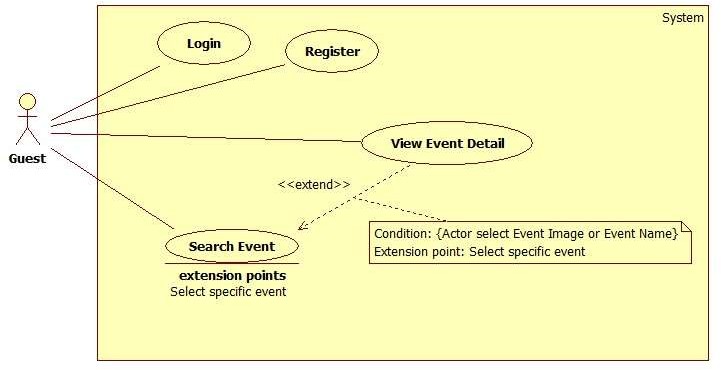
Ví dụ

### List of Use Case

<Đặc tả chi tiêt 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>

* + - 1. ***<Guest>Overview Use Case***



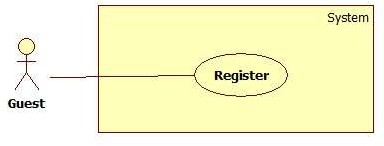
Ví dụ

***Figure 3: <Guest> Overview Use Case***

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

***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 - observate** 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**

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 |  | - |
| 2 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |
| **Actor:**   * <Actor sẽ thực hiện use case>   **Summary:**   * <Tóm tắt về tính năng của use case>   **Goal:**   * <Mục đích của use case: kết quả khi usecase kết thúc thành công>   **Triggers:**   * <Bước làm use case được kích hoạt>   **Preconditions:**   * <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*   + ***User phải là một customer***   + ***Hóa đơn vẫn đang trong tình trạng chưa hết thời hạn hủy của hệ thống là 3 ngày***   **Post Conditions:**   * < 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**   **Main Success Scenario: <Hướng xử lý chính của hệ thống>**  **Alternative Scenario: <Hướng xử lý khác trong tình huống dữ liệu cụ thể như** | | | |

|  |  |  |
| --- | --- | --- |
| No | Actor Action | System Response |
| 1 |  |  |

|  |  |  |
| --- | --- | --- |
| No | Actor Action | System Response |
|  |  |  |

Ví dụ

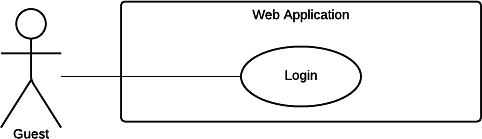
**mệnh đề if hoặc lựa chọn khác của người dùng trong quá trình main flow được diễn ra>**

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

**Relationships:** Mối 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 thế nào trong hệ thống)
* Các ràng buộc dữ liệu dưới hệ thống, các rule liên quan đến toàn vẹn dữ liệu
* Các qui trình, activities, quá trình chuyển đổi trạng thái của hệ thống



|  |  |  |  |
| --- | --- | --- | --- |
| *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:*   * *Guest Summary:* * *This use case allows guest to log in the system. Goal:* * *Guest can log in the system. Triggers:* * *Guest sends the login command. Preconditions:* * *N/A*   *Post Conditions:* | | | |

|  |  |  |
| --- | --- | --- |
| *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]* |

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

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

Ví dụ

* *Success: Guest login the system.*
* *Fail: Show error message. Main Success Scenario:*

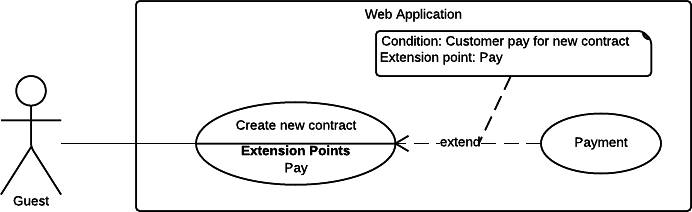
*Alternative Scenario:*

*Exceptions:*

*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.*
  + *If role is “Customer”, the system will display to Customer view.*
  + *If role is “Staff”, the system will display to Staff Dashboard view.*

***<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***   * *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)* * *Contract’s type: select one of the options.* * *Start date: date time input, required.* * *Contract term: text* * *Contract’s fee: text*   ***Vehicle information***   * *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***  - *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:*  *Exceptions:*  *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.* | | | | |

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

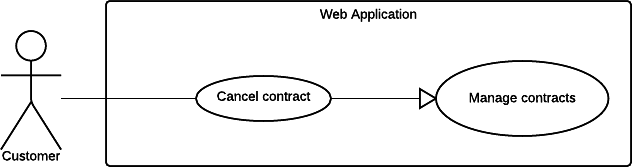
|  |  |  |
| --- | --- | --- |
| *No* | *Actor Action* | *System Response* |
| *1* | *Guest sends command to create*  *new contract request* | *System shows error message to ask user*  *input missing required fields.* |
| *2* | *Guest’s email is existed in the*  *system* | *Show message to notify guest 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.* |

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

Ví dụ

***<Customer> Cancel contract***

|  |  |  |
| --- | --- | --- |
| *Step* | *Actor Action* | *System Response* |
| *1* | *User goes to cancel contract view.* | *Display new view require user input some information:*   * *Reason to cancel the contract: can be optional selected from these values:*   + *“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 huỷ 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.* | * *Change contract status.* * *Send request to the Staff. [Exception 1]* |



***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:*   * *Customer.*   *Summary:*   * *This use case helps user cancel their contract. Goal:* * *Customer can cancel the contract. Triggers:* * *Customer sends cancel contract request. Preconditions:* * *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:* * *Success: Send to the staff the cancel contract request.* * *Fail: Show error message. Main Success Scenario:*   *Alternative Scenario: N/A* | | | |

|  |  |  |
| --- | --- | --- |
| *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.* |

***Table 7 Use case WC03 - <Customer> Cancel contract***

*Exceptions:*

*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.*

Ví dụ

**System**

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

System

**Auto parse**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***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:***   * *System.*   ***Summary:***   * *System can parse resource automatically from many websites at specified time.*   ***Goal:***   * *Get resource from many websites.*   ***Triggers:***   * *The time hits configured time.*   ***Preconditions:***   * *Parse time has been configured.*   ***Post Conditions:***   * ***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.* | | * *Send request to the parsed link.* * *Fetch data from the response based on the inputted XPaths.* | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | * *Validate data [Exception 1].* * *If data is valid, insert to storage [Alternative 1].* * *Generate log file.* |  |
| ***Alternative Scenario:***  ***Exceptions:***  ***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}*  *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***

|  |  |  |
| --- | --- | --- |
| *Step* | *Actor Action* | *System Response* |
| *1* | *Server checks the current time. If it hits configured time, parse process starts.* | * *If fetched link resource is already in the storage, update its information.* * *Generate log file.* |

|  |  |  |
| --- | --- | --- |
| *No* | *Actor Action* | *System Response* |
| *1* | *Data is invalid.* | * *Generate log file.* |

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

* 1. **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.>

* + 1. **Usability**
    2. **Reliability**
    3. **Availability**
    4. **Security**
    5. **Maintainability**
    6. **Portability**
    7. **Performance**

…..

* 1. **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

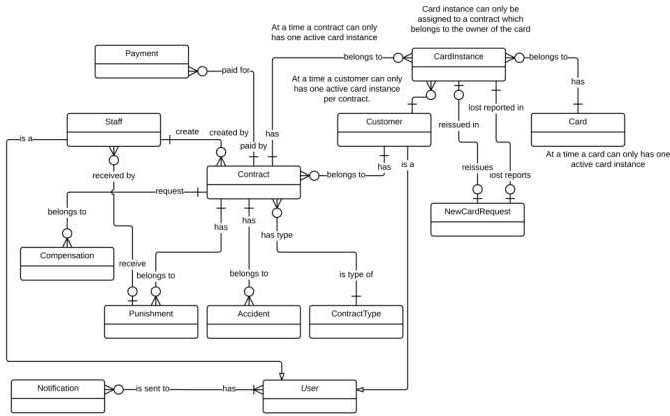


Figure 8 Conceptual diagram

***Data Dictionary***

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

***Table 9 Conceptual Diagram Data Dictionary***

# eport No. 4 Software Design Description

## Design Overview

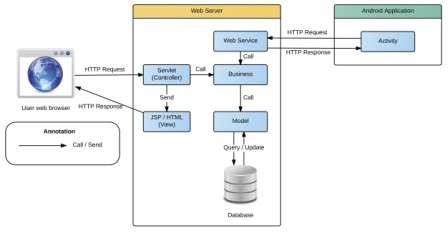
<Nội dung này tham khảo và có thể giữ nguyên và chỉ thay thế các phần phù hợp với đồ án của nhóm. Nhóm có thể viết lại cho hay hơn>

* *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****.***
    1. **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 ...

* + 1. **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***

## Detailed Description

### 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

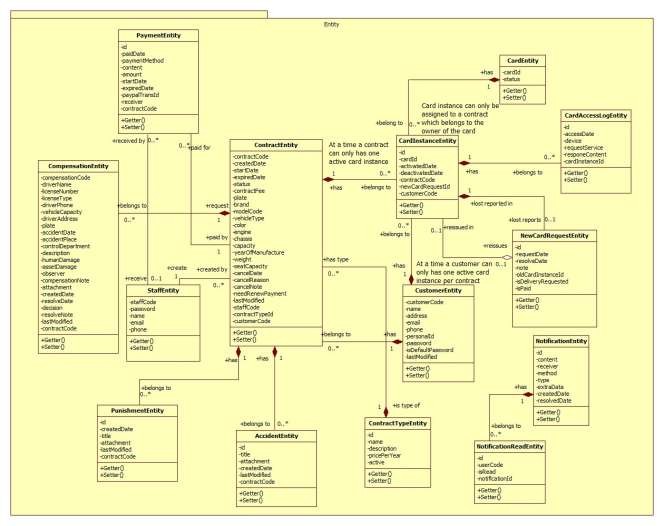


Figure 11 Class Diagram

|  |  |  |
| --- | --- | --- |
| *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.* |

Ví dụ

***Table 11 Class dictionary***

### Class Diagram Explanation

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

* + - * 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**

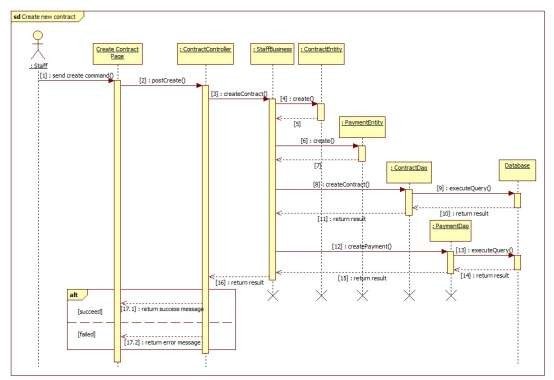
<Sử dụng **sequence diagram là chủ yếu để trình bày nội này**. Sequence diagram cần kết hợp giữa các class đã trình bày ở trên kết hợp với các kiến trúc đã được thuyết minh để có mô hình phù hợp. Đối với ứng **dụng điện thoại di động thì nên sử dụng activity diagram**>

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

Ví dụ

* + - 1. ***Create new contract***

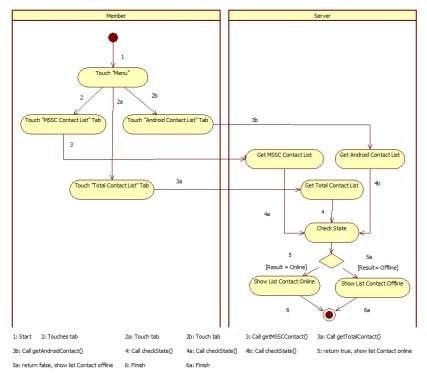
Summary: this diagram show process of staff creates new contract



***Figure 12 Sequence diagram - <Staff> Create new contract***

* + - 1. <Member> View Friend List

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



* + 1. **Interface**

***Figure 13: <Member> View Friend List***

* + - 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* |
| *public ResponseObject getCheckConnection(R r)* | *Check server status* | *Request object r* | *Json Boolean the status of server* | *Boolean* | *JsonProcessi ngException* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *...* |  |  |  |  |  |

Ví dụ

### 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**

* + - 1. ***Guest Interface Design***
         1. ***Login***



***Fields***

***Figure 14: Login***

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

* + 1. **Database Design**
       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>

* + - 1. **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>

* + 1. **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

* 1. ***Document Breakdown***
     1. Definition

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

* + 1. ***Define Problem***

*All content of document is quite difficute for manage so we must re-construc structure of document for using.*

* + 1. ***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 incluses 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*

* + 1. ***Complexity***
       - *In total, the complexity of this algorithm is *
    2. ***Flowchart***

PAGE \\* MERGEFORMAT 1



**Figure 15: Breakdown document flow chart**

#### String Comparison

* + 1. Define Problem

*Given two strings. Calculate their matching percent.*

* + 1. ***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.*
  + 1. ***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 pairs(s) which returns a list of adjacent character pairs of string s.*
* *Then, we use below formula to calculate matching percent.*
  + 1. ***Example***

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

* + *Upper case 2 strings:*

+ *France FRANCE.*



+ *French FRENCH.*

* + *Break string into list of adjacent character pairs:*

+ *FRANCE*



+ *FRENCH*

* + *Calculate its matching percent.*



1. **System Implementation & Test**
   1. **Introduction**
      1. **Overview**

<Mô tả tống 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.

### Test Approach

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

* 1. **Database Relationship Diagram**
     1. **Physical Diagram**

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

* + 1. **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**

* 1. **Performance Measures**

<Cách nhóm ước lượng việc đo đạc hệ thống>

Ví dụ

* + 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.*

* 1. **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.

### Features to be tested

<Tính năng sẽ kiểm thử>

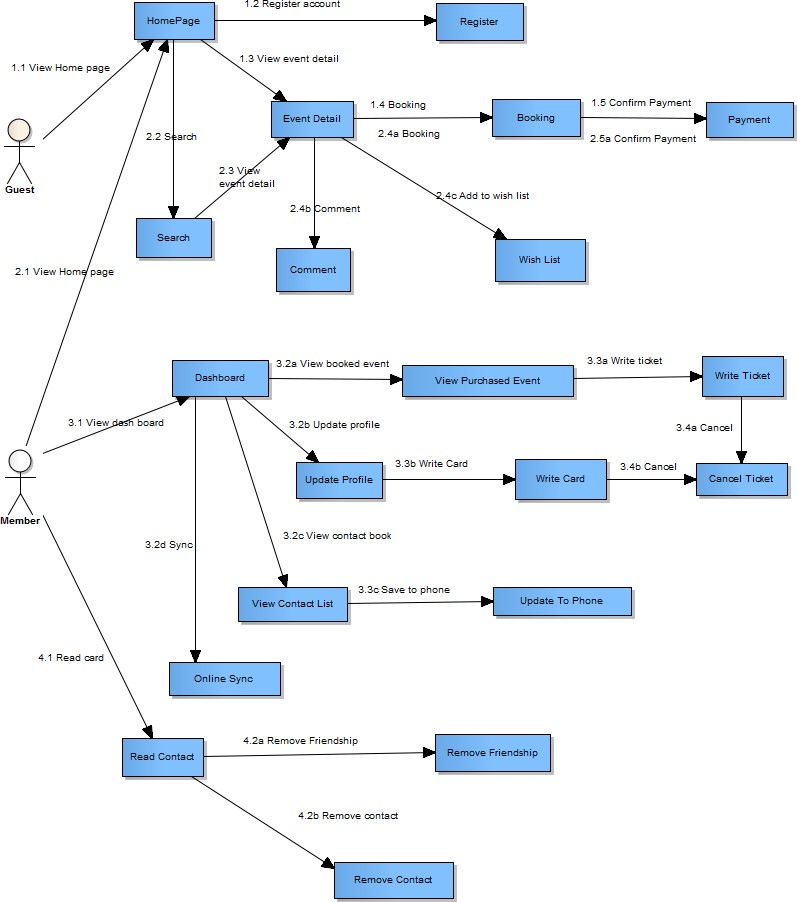
* + 1. **Features not to be tested**

<Tính năng sẽ không kiểm thử>

* 1. **System Testing Test Case**

**<Nên vẽ các workflow tính năng sẽ test để dể 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***

MSSC - Introduction

* + 1. ***Guest Test Case***

***5.1.1 Search Event***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***ID*** | ***Test Case Description*** | ***Test Case Procedure*** | ***Expected output*** | ***Inter-test Case Dependence*** | ***Result*** | ***Test Date*** | ***Note*** |
|  |  |  |  |  |  |  |  |

MSSC - Introduction

47

1. **Software User’s Manual**
   1. **Installation Guide**
      1. **Setting up environment at server side**

The following software must be installed into the server machine:

* + - 1. **Hardware requirements**

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

* + - 1. **Software requirements**

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

* + 1. **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. **Prepare deployment package**
      2. **Configure Server before deploy**
      3. **Deploy web application on server**
    1. **Setting up the environment at client side**
       1. **Setting up for computer**

<Ghi rõ phiên bản tối thiểu để sử dụng>

* 1. **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>>