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

**FPT UNIVERSITY**

|  |
| --- |
| **Capstone Project Document** |
| **Remindful Refrigerator Application** |

|  |  |
| --- | --- |
| **Group 2** | |
| **Group members** | **Nguyễn Hoàng Giang – SE61275**  **Trương Thanh Lâm – SE61257**  **Nguyễn Lê Hoàng Thiện – SE61277**  **Phan Hoàng Giáp – SE61347** |
| **Supervisor** | **Kiều Trọng Khánh** |
| **Ext. Supervisor** | **N/A** |
| **Capstone Project Code** | **RRA** |

- Ho Chi Minh City, January 04, 2016 -

Table of Contents

[Table of Contents 3](#_Toc441487142)

[List of Tables 5](#_Toc441487143)

[List of Figures 6](#_Toc441487144)

[Definitions, Acronyms and Abbreviations 7](#_Toc441487145)

[A. REPORT NO. 1: Introduction 8](#_Toc441487146)

[1. Project Information 8](#_Toc441487147)

[2. Introduction 8](#_Toc441487148)

[3. Current Situation 8](#_Toc441487149)

[4. Problem Definition 8](#_Toc441487150)

[5. Proposed Solution 9](#_Toc441487151)

[5.1. Feature functions 9](#_Toc441487152)

[5.2. Advantages 9](#_Toc441487153)

[5.3. Disadvantages 9](#_Toc441487154)

[6. Functional Requirements 9](#_Toc441487155)

[7. Roles and Responsibilities 10](#_Toc441487156)

[B. REPORT NO. 2: Software Project Management Plan 11](#_Toc441487157)

[1. Problem Definition 11](#_Toc441487158)

[1.1. Name of this Capstone Project 11](#_Toc441487159)

[1.2. Problem Abstract 11](#_Toc441487160)

[1.3. Project Overview 11](#_Toc441487161)

[1.3.1. Current Situation 11](#_Toc441487162)

[1.3.2. The Proposed System 11](#_Toc441487163)

[1.3.2.1. Mobile Application 12](#_Toc441487164)

[1.3.2.2. Website 12](#_Toc441487165)

[1.3.3. Boundaries of the System 12](#_Toc441487166)

[1.3.4. Future Plan 12](#_Toc441487167)

[1.3.5. Development Environment 13](#_Toc441487168)

[1.3.5.1. Hardware requirements 13](#_Toc441487169)

[1.3.5.2. Software requirements 13](#_Toc441487170)

[2. Project Organization 14](#_Toc441487171)

[2.1. Software Process Model 14](#_Toc441487172)

[2.2. Roles and Responsibilities 14](#_Toc441487173)

[2.3. Tools and Techniques 16](#_Toc441487174)

[3. Project Management Plan 16](#_Toc441487175)

[3.1. Software Development Life Cycle 16](#_Toc441487176)

[3.2. Phase Details 17](#_Toc441487177)

[3.2.1. Phase 1: Requirement Analysis and Definition 17](#_Toc441487178)

[3.2.2. Phase 2: System and Software Design 17](#_Toc441487179)

[3.2.3. Phase 3: Implementation and Unit Testing 18](#_Toc441487180)

[3.2.4. Phase 4: Integration and System Testing 18](#_Toc441487181)

[3.2.5. Phase 5: Operation and Maintenance 18](#_Toc441487182)

[3.3. All Meeting Minutes 19](#_Toc441487183)

[4. Coding Convention 19](#_Toc441487184)

[C. REPORT NO. 3: Software Requirement Specification 20](#_Toc441487185)

[1. User Requirement Specification 20](#_Toc441487186)

[1.1. Guest Requirement 20](#_Toc441487187)

[1.2. Member Requirement 20](#_Toc441487188)

[1.3. Staff Requirement 20](#_Toc441487189)

[1.4. Scheduler Requirement 20](#_Toc441487190)

[2. User Requirement Specification 20](#_Toc441487191)

[2.1. External Interface Requirement 20](#_Toc441487192)

[2.2. System Overview Use Case 21](#_Toc441487193)

[2.3. List of Use Case 21](#_Toc441487194)

List of Tables

[Table 1 - Roles and Responsibility 10](#_Toc441487074)

[Table 2 - Hardware Requirements for Server 13](#_Toc441487075)

[Table 3 - Hardware Requirements for Mobile 13](#_Toc441487076)

[Table 4 - Software Requirements 14](#_Toc441487077)

[Table 5 - Roles and Responsibilities Details 16](#_Toc441487078)

[Table 6 - Tools and Techniques 16](#_Toc441487079)

[Table 7 - Software Development Life Cycle 17](#_Toc441487080)

[Table 8 - Phase 1: Requirement gathering and analysis 17](#_Toc441487081)

[Table 9 - Phase 2: System design 18](#_Toc441487082)

[Table 10 - Phase 3: Implementation 18](#_Toc441487083)

[Table 11 - Phase 4: Testing 18](#_Toc441487084)

[Table 12 - Phase 5: Deployment of System 18](#_Toc441487085)

List of Figures

[Figure 1 - Waterfall model 14](#_Toc441446506)

[Figure 2 - System Overview Use Case 21](#_Toc441446507)

Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| RRA | Remindful Refrigerator Application |
| BA | Business Analysis |
| DEV | Developer |
| SRS | Software Requirements Specification |
| ERD | Entity-Relationship Diagram |

1. REPORT NO. 1: Introduction
   1. Project Information

* Project name: Remindful Refrigerator Application
* Project code: RRA
* Product type: mobile application + web
* Start date: 04/01/2016
* End date:
  1. Introduction

In this document, we introduce a solution for tracking food in refrigerator. Almost every family has a refrigerator with a lot of food inside. But they cannot remember everything, especially petty things such as expiry date of food. Based on our analysis and our own experiences, we think of a solution for every people to manage their food easier.

Remindful Refrigerator Application (RRA) is built to solve current problem. RRA is convenient to manage information of all food in fridge, checking which food is expiring and notify to the user so that they can use it earlier instead of throwing it away. We also suggest some dishes which can be made with what the user have in their fridge.

This document also describes our working process in four months, includes our perspective in the system, component design, detailed core workflows. We hope our solution will help resolve the problems that everyone can face to.

* 1. Current Situation

About user’s behavior: Many people prefer to buy a lot of food by once and store them in their fridges for saving time. This behavior causes them to get confused about which food/ingredients they are having, their expiry date and how to cook with them.

Current solutions in the market: Samsung introduced a smart refrigerator product, which uses Android to store inside food’s information. However, those features are attached into Samsung’s own product, which means user have to buy a whole new fridge if they want to use the management function. Besides, its interface is inconvenient for users, as the inputting keyboard is huge and slowing down typing. One more problem is users will have difficulties interact with the product from distance, as they cannot bring their fridge along with them.

* 1. Problem Definition

There are disadvantages of current situation:

* With a lot of food in the fridge, people sometimes do not remember or care about the expiration but keep using those food. Using expired food may affect to people heath, and even if they do not use them, it also be a waste.
* People may be confused with a lot of food and ingredients. They do not know what or how to cook with those food.

About the product of SAMSUNG:

* Advantages:
* Friendly interface.
* Have some additional features such as Music and Video player.
* Disadvantages:
* Not flexible: user must buy the whole fridge to use the provided features.
* Wasting energy: the large tablet integrated with the fridge require more energy to work.
* The system is just a demo.
  1. Proposed Solution

According to the growth of smart phone, we proposed a solution is to build a mobile application called “Remindful Refrigerator Application” (or shortly RRA). RRA includes a mobile app and an admin panel with following functions:

* + 1. Feature functions

RRA provides these following core functions:

* Add food: user can add food information to database.
* Search dishes: according to food that user have in fridge, they can manually search for appropriate dishes when they need.
* Make notification: with expiring food, system will send notification to user.
* Suggest dishes: with expiring food, system will give some dishes that user can make with that food.
  + 1. Advantages
* User can easily keep track on their food. They can know which food should be used first.
* User can get suggestion about dishes can be cook with their food.
* Application is compatible with most of Android smartphone.
* Flexible: application can be used dependently from fridges.
  + 1. Disadvantages
* There are many competitive in market.
  1. Functional Requirements

Functional requirements are listed as below:

* Mobile app:
* New account/Login
* Add new food
* Get notification
* Get suggestion
* Search dishes
* Add dishes to favorite list
* Admin panel:
* Input new resources website
* View report
* Scheduler:
* Send notification
* Make suggestion
* Parse recipes
* Create report
  1. Roles and Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project manager | Supervisor | khanhkt@fpt.edu.vn |
| 2 | Nguyễn Hoàng Giang | Developer | Leader | giangnhse61275@fpt.edu.vn |
| 3 | Trương Thanh Lâm | Developer | Member | lamttse61257@fpt.edu.vn |
| 4 | Nguyễn Lê Hoàng Thiện | Developer | Member | thiennlhse61277@fpt.edu.vn |
| 5 | Phan Hoàng Giáp | Developer | Member | giapphse61347@fpt.edu.vn |

Table - Roles and Responsibility

1. REPORT NO. 2: Software Project Management Plan
   1. Problem Definition
      1. Name of this Capstone Project

* Official name: Remindful Refrigerator Application
* Vietnamese name: Tủ lạnh tiện lợi
* Abbreviation: RRA
  + 1. Problem Abstract

To support users to keeping track of their food, we concentrate on one thing familiar with people nowadays - their smart phones. We provide a mobile application that give users a helpful method to manage food in their fridges, as well as those foods’ expired date. The application also aids users by reminding them about incoming expired date in case they forget. In addition, we develop a web site as an alternative using way for the application in some situations.

Moreover, the application can suggest some dishes based on the ingredients in the fridge. It also provides users with recipes if they are interested in a certain one. To support making such suggestions, we have a team that continuously looking for good dishes from many culinary websites.

* + 1. Project Overview
       1. Current Situation

Some problems encountered in this project:

* User’s behavior: Users may forget to update data of their food when changes occur. Example: buying or using some food
* Rarity of food: Some food may be rare or in short use, hardly finding matching dishes.
* Variety of users: People have different favorite tastes. The work of finding dishes to satisfy all of them is very difficult.
* Gathering data: culinary websites have different interfaces and structures, which makes gathering data more difficult.
  + - 1. The Proposed System

We will build a mobile application as the main functions for users. The application consumes web services from the server to perform its assignments. The application can cache data such as viewed dishes, favorite dishes and personal settings.

We also develop a web site with some same functions from the mobile application to make an alternative. This helps ensuring the availability of the system.

The system has a scheduler service to perform “scheduled” tasks such as notify users and parse information periodically.

We have an internal site for staff team to gather data about dishes and recipes from culinary websites.

* + - * 1. Mobile Application

Mobile Application is the mainly part used by users. It has the following core functions:

* Manage food and their expired date
* Scan barcode with camera for quick adding new food
* Notification about expiring food
* Suggest dishes and view recipes
* Caching data for using without internet connection
  + - * 1. Website

Web Site is an essential part of the system. The user’s site provides some functions like the mobile application, while the staff’s site performs gathering data task.

* For users:
  + Manage food
  + Suggest dishes and view recipes
* For staffs:
  + Get data from culinary sites
    - 1. Boundaries of the System

This system can:

* Allows users to manage food that are put in their refrigerator.
* Notify users about expiring food.
* Suggest dishes to user based on the food user currently have, using data gathered from culinary websites.
  + - 1. Future Plan

Current system has simple managing and suggesting functions. With further research, the system can apply high techniques of data mining to develop expanded suggestion functions, such as:

* Nutrition scaled dishes: suggest dishes with a balanced or suitable nutrition ingredients for different meals of a day (breakfast, lunch, dinner), or for different types of people (diet, growing children).
* Food and news: suggest news about food for user.
  + - 1. Development Environment
         1. Hardware requirements
* For Server

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum** | **Recommended** |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Winder Server 2008 | Winder 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 - Hardware Requirements for Server

* For Mobile

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum** | **Recommended** |
| Internet Connection | Wi-Fi or 3G (4 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.2.2 | Android 4.4.2 |
| Computer Processor | Cortex-A7 Dual-Core 1.3GHz | Cortex-A7 Dual-Core 1.3GHz |
| Computer Memory | 1GB RAM | 2GB or more |

Table - Hardware Requirements for Mobile

* + - * 1. Software requirements

|  |  |  |
| --- | --- | --- |
| **Software** | **Name/Version** | **Description** |
| Operating system | Windows 7 or higher | Operating system and platform for development |
| Environment | Java EE 6 | Specification for developing |
| Modeling tool | StarUML 2.0 | Used for data modeling |
| IDE | Netbeans 7.4 or 8.0.1  Android Studio 1.5.1 | Programming tools |
| DBMS | Microsoft SQL Server 2008 | Used to create and manage the database for the system |
| Source control | TortoiseSVN 1.8.2 | Used for source control |
| Web Browser | Chrome 42 or above  Internet Explorer 9 or above | Testing browser |
| Android Emulator | Genymotion 2.6.0 | Android emulator used for testing mobile application |

Table - Software Requirements

* 1. Project Organization
     1. Software Process Model



Figure - Waterfall model

Reference: SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

The Waterfall Model is a linear-sequential process used in software development. This model requires each phase to be fully completed before the next phase can begin. This model is used for small project, which has no uncertain requirements. We choose waterfall model for RRA project because of these following reasons:

* RRA is a small project, which must be done in a short time (14 weeks) with only four members.
* Team members are inexperience since no one has not done any real project, so that all requirements must be well-defined and stay stable for team to work better. Other models (such as Agile) are not appropriate for us because of the limited skills.
* All project aspects are clear to team members and hence every member can understand the whole system instead of their parts only.
* Team meetings are held frequently to keeps the coherence of the team as well as the project.
  + 1. Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Kiều Trọng Khánh | Project manager | * Specify user requirements * Control the development process * Give out technique and business analysis support |
| **2** | Nguyễn Hoàng Giang | Team leader, BA, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| **3** | Trương Thanh Lâm | Team member, BA, DEV, Tester | * Clarifying requirements * Prepare documents * GUI Design * Designing database * Coding * Create test plan * Testing |
| **4** | Nguyễn Lê Hoàng Thiện | Team member, BA, DEV, Tester | * Clarifying requirements * Prepare documents * GUI Design * Designing database * Coding * Create test plan * Testing |
| **5** | Phan Hoàng Giáp | Team member, BA, DEV, Tester | * Clarifying requirements * Prepare documents * GUI Design * Designing database * Coding * Create test plan * Testing |

Table - Roles and Responsibilities Details

* + 1. Tools and Techniques

|  |  |
| --- | --- |
| **Tool/Technique** | **Name/Version** |
| Front-end | HTML, CSS5, JavaScript, jQuery, Boostrap |
| Back-end | JavaEE, Servlet, JSP |
| Database Management System | Microsoft SQL Server 2008 R2 Express |
| Mobile | Android Studio 1.5.1 |
| Web Server | Apache Tomcat 7.0.42 |
| Scheduler Framework | Quartz Scheduler |
| Parser API | jSoup |

Table - Tools and Techniques

* 1. Project Management Plan
     1. Software Development Life Cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resources needed** | **Dependencies and Constrains** | **Risks** |
| Requirement Analysis and Definition | - Collect requirements.  - Identify and clarify requirements. | - Software requirements specification. | 10 man-days | N/A | - Missing requirements.  - Unclear scope of project.  - Lack of member share of understand. |
| System and Software Design | - Architectural design  - Interface design  - Component design  - Database design | - System architecture  - Database specification  - Interface specification  - Component specification | 15 man-days | Depend on “Requirement Analysis and Definition” | - Lack of experience.  - Not fulfil requirements. |
| Implementation and Unit Testing | - Implement system design into real system. | - Web application.  - Mobile application on Android.  - Physical database. | 60 man-days | Depend on “System and Software design” | - Mistake while implementing. |
| Integration and System Testing | - Execute the system with test cases  - Fix bugs | - Test report. | 15 man-days | Depend on “Implementation and Unit Testing” | - Lack of experience.  - Lack of test case. |
| Operation and Maintenance | - Deploy on web server and mobile. | - Installation guide.  - User manual. | 10 man-days | Depend on “Integration and System Testing” | - Lack of experience. |

Table - Software Development Life Cycle

* + 1. Phase Details
       1. Phase 1: Requirement Analysis and Definition

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Collect requirements | Find which systems currently provide similar service, their strength and weakness. | GiangNH, LamTT, ThienNLH, GiapPH |
| 2. Identify and clarify main functions | Define which main function system should provide. | GiangNH, LamTT, ThienNLH, GiapPH |
| 3. Requirements specification | Define the requirements in details.  Write SRS document. | GiangNH, LamTT, ThienNLH, GiapPH |
| 4. Requirement validation | Check the validity of the requirements and complete SRS document. | GiangNH, LamTT, ThienNLH, GiapPH |

Table - Phase 1: Requirement gathering and analysis

* + - 1. Phase 2: System and Software Design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Architectural design | Define the overall structure of the system.  Define the main components, the relationships between them and how they are distributed. | GiangNH, LamTT, ThienNLH, GiapPH |
| 2. Interface design | Define the interfaces between system components. | GiangNH, LamTT, ThienNLH, GiapPH |
| 3. Component design | Design how each component will operate. | GiangNH, LamTT, ThienNLH, GiapPH |
| 4. Database design | Design the system data structures and how these are to be represented in a database. | GiangNH, LamTT, ThienNLH, GiapPH |

Table - Phase 2: System design

* + - 1. Phase 3: Implementation and Unit Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Implement web back-end | Code functions for web application and web services for mobile to consume. | LamTT |
| 2. Implement web front-end | Code GUI for web application | LamTT |
| 3. Implement mobile application | Code GUI and functions for mobile application | GiangNH |
| 4. Implement scheduler | Code functions of scheduler | GiapPH |
| 5. Implement parser | Code parser | ThienNLH |
| 6. Unit testing | Write test cases and run to ensure that code meets the designs and behaves as expected. | GiangNH, LamTT, ThienNLH, GiapPH |

Table - Phase 3: Implementation

* + - 1. Phase 4: Integration and System Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Component testing | Test each component independently  Fix bugs | GiangNH, LamTT, ThienNLH, GiapPH |
| 2. System testing | Test the whole system.  Fix bugs | GiangNH, LamTT, ThienNLH, GiapPH |
| 3. Acceptance testing | Testing with customer data to check that the system meets the customer’s needs. | GiangNH, LamTT, ThienNLH, GiapPH |

Table - Phase 4: Testing

* + - 1. Phase 5: Operation and Maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Maintain the system | Maintain current system | N/A |

Table - Phase 5: Deployment of System

* + 1. All Meeting Minutes

All meeting minutes are saved at:

<https://github.com/Giang94/CapstoneProject_SmartRefrigerator/tree/master/Doc/Meeting%20Minutes>

* 1. Coding Convention

Using Java coding convention to develop website, parser, web services and mobile application.

Conventions:

* Declaration:
* One declaration per line.
* Declarations should be placed only at the beginning of blocks.
* Naming:
* Variable and method names are in mixed case, with first letter of each internal word capitalized except first word.
* Method names should be verbs.
* Class names should be nouns, in mixed case with first letter of each internal word capitalized.
* Constant names should be all uppercase with words separated by underscore.
* Indentation:
* Avoid lines longer than 80 characters.

References: <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

1. REPORT NO. 3: Software Requirement Specification
   1. User Requirement Specification
      1. Guest Requirement

Guest is a person who doesn’t have access to the system. Guest can only use two functions in the system. To use others functions, guest must login. These are two functions guest can use:

* Register: create a new account so that guest can login to the system.
* Login: after login, guest will have a role in the system and can use appropriate functions.
  + 1. Member Requirement

Member is a person who logged in the system. Member is the main user of the system and can use some function in the system. A member can use these following functions after logged into the system with member account:

* Manage food:
* Add new food
* Edit existing food
* Remove food
* Search dish
* View recipe
* Add dish to favorite list
  + 1. Staff Requirement

Staff is a person with permission to manage some aspect of the system. Staff can use these functions after logged into the system with staff account:

* Add new website
* View log
* Parse recipe
  + 1. Scheduler Requirement

Scheduler is a part of RRA allows system to do some functions automatically. These automatic functions are:

* Parse recipe
* Match dish
* Send notification
* Write log
  1. User Requirement Specification
     1. External Interface Requirement
     2. System Overview Use Case

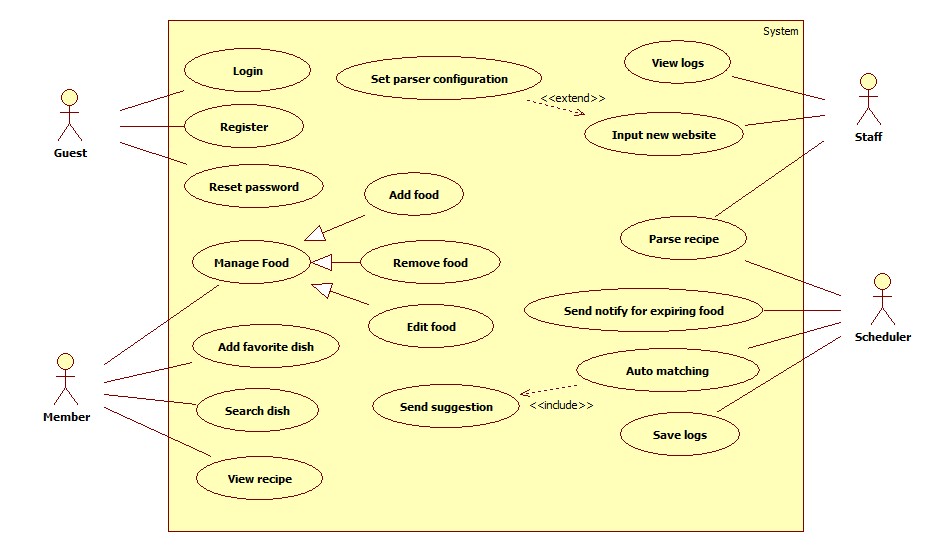


Figure - System Overview Use Case

* + 1. List of Use Case