|  |  |
| --- | --- |
|  | **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](#_Toc440533173)

[List of Tables 5](#_Toc440533174)

[List of Figures 6](#_Toc440533175)

[Definitions, Acronyms and Abbreviations 7](#_Toc440533176)

[A. REPORT NO. 1: Introduction 8](#_Toc440533177)

[1. Project Information 8](#_Toc440533178)

[2. Introduction 8](#_Toc440533179)

[3. Current Situation 8](#_Toc440533180)

[4. Problem Definition 8](#_Toc440533181)

[5. Proposed Solution 9](#_Toc440533182)

[5.1. Feature functions 9](#_Toc440533183)

[5.2. Advantages 9](#_Toc440533184)

[5.3. Disadvantages 9](#_Toc440533185)

[6. Functional Requirements 9](#_Toc440533186)

[7. Roles and Responsibilities 10](#_Toc440533187)

[B. REPORT NO. 2: Software Project Management Plan 11](#_Toc440533188)

[1. Problem Definition 11](#_Toc440533189)

[1.1. Name of this Capstone Project 11](#_Toc440533190)

[1.2. Problem Abstract 11](#_Toc440533191)

[1.3. Project Overview 11](#_Toc440533192)

[1.3.1. Current Situation 11](#_Toc440533193)

[1.3.2. The Proposed System 11](#_Toc440533194)

[1.3.2.1. Mobile Application 12](#_Toc440533195)

[1.3.2.2. Website 12](#_Toc440533196)

[1.3.3. Boundaries of the System 12](#_Toc440533197)

[1.3.4. Future Plan 12](#_Toc440533198)

[1.3.5. Development Environment 12](#_Toc440533199)

[1.3.5.1. Hardware requirements 12](#_Toc440533200)

[1.3.5.2. Software requirements 13](#_Toc440533201)

[2. Project Organization 13](#_Toc440533202)

[2.1. Software Process Model 13](#_Toc440533203)

[2.2. Roles and Responsibilities 13](#_Toc440533204)

[2.3. Tools and Techniques 14](#_Toc440533205)

[3. Project Management Plan 15](#_Toc440533206)

[3.1. Software Development Life Cycle 15](#_Toc440533207)

[3.2. Phase Details 16](#_Toc440533208)

[3.2.1. Phase 1: Requirement gathering and analysis 16](#_Toc440533209)

[3.2.2. Phase 2: System design 17](#_Toc440533210)

[3.2.3. Phase 3: Implementation 17](#_Toc440533211)

[3.2.4. Phase 4: Testing 17](#_Toc440533212)

[3.2.5. Phase 5: Deployment of System 18](#_Toc440533213)

[3.2.6. Phase 6: Maintenance 18](#_Toc440533214)

[3.3. All Meeting Minutes 18](#_Toc440533215)

[4. Coding Convention 19](#_Toc440533216)

List of Tables

[Table 1 - Roles and Responsibility 10](#_Toc440533117)

[Table 2 - Hardware Requirements for Server 13](#_Toc440533118)

[Table 3 - Hardware Requirements for Mobile 13](#_Toc440533119)

[Table 4 - Roles and Responsibilities Details 14](#_Toc440533120)

[Table 5 - Tools and Techniques 15](#_Toc440533121)

[Table 6 - Software Development Life Cycle 16](#_Toc440533122)

[Table 7 - Phase 1: Requirement gathering and analysis 16](#_Toc440533123)

[Table 8 - Phase 2: System design 17](#_Toc440533124)

[Table 9 - Phase 3: Implementation 17](#_Toc440533125)

[Table 10 - Phase 4: Testing 18](#_Toc440533126)

[Table 11 - Phase 5: Deployment of System 18](#_Toc440533127)

[Table 12 - Phase 6: Maintenance 18](#_Toc440533128)

List of Figures

[Figure 1 - Waterfall model 12](#_Toc440457825)

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.
  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 1 - 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.
  + - 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 get necessary data from outside sources.

* + - * 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
  + - * 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: provide users with trending news about food that they are going to use.
  + - 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 2 - 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 3 - Hardware Requirements for Mobile

* + - * 1. Software requirements
  1. Project Organization
     1. Software Process Model



Figure 1 - Waterfall model

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

* + 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 4 - Roles and Responsibilities Details

* + 1. Tools and Techniques

|  |  |  |
| --- | --- | --- |
| **Tool/Technique** | **Name/Version** | **Description** |
| Operating system | Windows 7 | 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 |
| Front-end | HTML, CSS, jQuery , Javascript, Bootstrap |  |
| HTML Parser | HtmlUnit |  |
| Scheduler Library | Quartz Scheduler |  |

Table 5 - Tools and Techniques

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resources needed** | **Dependencies and Constrains** | **Risks** |
| Requirement Analysis | - Collect requirements.  - Identify and clarify requirements. | - Introduction of system.  - Project task plan.  - Prototypes  - Software requirements specification. | 20 man-days | N/A | - Missing requirements.  - Unclear scope of project.  - Lack of member share of understand. |
| Design | - Architecture design for the system.  - Details 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 requirements. |
| Implementation | - Code GUI  - Code core functions and other related features.  - Unit test and debug. | - Web application.  - Mobile application on Android. | 30 man-days | Depend on “Software design document” | - Mistake while implementing. |
| Testing | - Integration testing.  - System testing.  - Fix bugs. | - Test document. | 10 man-days |  | - Lack of experience.  - Lack of test case. |
| Deployment | - Deploy on web server and mobile. | - Installation guide.  - User manual. | 10 man-days |  | - Lack of experience. |
| Maintenance | Maintain and update the system. | N/A | N/A | N/A | N/A |

Table 6 - Software Development Life Cycle

* + 1. Phase Details
       1. Phase 1: Requirement gathering and analysis

|  |  |  |
| --- | --- | --- |
| **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. Create system | Complete the introduction of system. | GiangNH, LamTT |
| 4. Create SRS document | Write SRS document. | GiangNH, LamTT, ThienNLH, GiapPH |
| 5. Design prototypes | Design GUI prototypes. | GiangNH, LamTT, ThienNLH, GiapPH |

Table 7 - Phase 1: Requirement gathering and analysis

* + - 1. Phase 2: System design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Research technologies | - Study Android application development  - Study web services  - Research frameworks that appropriate with system scope. | GiangNH, LamTT, ThienNLH, GiapPH |
| 2. Design ERD | Create Entity-Relationship diagram | GiangNH, LamTT, ThienNLH, GiapPH |
| 3. Design database | Design logical database and create physical database. | GiangNH, LamTT, ThienNLH, GiapPH |
| 4. Design class diagram | Create class diagram | GiangNH, LamTT, ThienNLH, GiapPH |
| 5. Create design document | Complete design document | GiangNH, LamTT, ThienNLH, GiapPH |

Table 8 - Phase 2: System design

* + - 1. Phase 3: Implementation

|  |  |  |
| --- | --- | --- |
| **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 | Unit test and fix bugs | GiangNH, LamTT, ThienNLH, GiapPH |

Table 9 - Phase 3: Implementation

* + - 1. Phase 4: Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Unit testing | - Write test cases  - Execute test cases and fix bugs | GiangNH, LamTT, ThienNLH, GiapPH |
| 2. Integration testing | - Write test cases  - Execute test cases and fix bugs | GiangNH, LamTT, ThienNLH, GiapPH |
| 3. System testing | - Write test cases  - Execute test cases and fix bugs | GiangNH, LamTT, ThienNLH, GiapPH |
| 4. Create test document | - Create and complete test document | GiangNH, LamTT, ThienNLH, GiapPH |

Table 10 - Phase 4: Testing

* + - 1. Phase 5: Deployment of System

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Deploy web application | Deploy web application on web server | LamTT, ThienNLH, GiapPH |
| 2. Deploy mobile application | Deploy mobile application on real device | GiangNH |
| 3. Create installation guide for web application | Write installation guide for web application | LamTT, ThienNLH, GiapPH |
| 4. Create installation guide for mobile application | Write installation guide for mobile application | GiangNH |
| 5. Create user manual for web application | Write user manual for web application | LamTT, ThienNLH, GiapPH |
| 6. Create user manual for mobile application | Write user manual for mobile application | GiangNH |

Table 11 - Phase 5: Deployment of System

* + - 1. Phase 6: Maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Maintain and update system | Maintain current system  Update new features | N/A |

Table 12 - Phase 6: Maintenance

* + 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>