Spotifood

Test record

By

Mr. Kittinut Saengsri 592115003

Mr. Sirisak Wongwal 592115031

Department of Software Engineering

College of Arts, Media, and Technology

Chiang Mai University

Project Advisor

Dr. Passakorn Phannachitta

Chapter 1 | Introduction

1.1 Identification

This Project Management Plan is the document for planning, scheduling activities and evaluating overall of the project so that it could complete successfully despite all the risks. It provides detailed scheduled, assigned task and identified risks of the project.

1.2 Project Overview

The Spotifood application will guide people find their new dish and help people who are interested in cooking and people who have cooking skill by suggesting the dishes from a word that people search. The AI system will suggest a dish from the personality of people. For example, The people regularly search steamed fish with herbs dish. The AI system will analyze the ingredient tag of dish that is related to the healthy tag. Then, the AI will suggest the people in the type of healthy dishes.

The suggest dish of AI will help those people found new dishes and challenging in cooking every day that makes people not disconnected and bored to cooking. Moreover, we will be a middleman that brings trader of ingredient to people who can’t find the ingredients in any food. The Spotifood help people to buy an ingredient. And the Spotifood will suggest a dish that is similar to personality or suggests from the behavior of people.

1.2.1 Purpose

The main purpose of developing the Spotifood is to help the people who are interested in cooking and who already have skills not bored with cooking or disconnected to prepare the food.

1.2.2 Scope

There are six major parts in Spotifood.

Feature #1: Authentication

Feature #2: Content CRUD

Feature #3: User management

Feature #4: Similar dishes identification

Feature #5: Personalization

Feature #6: ADS ingrediente

1.3 Work Product to be developed

1.3.1 Deliverable

1.3.2 Non-Deliverables

1.4 Acronyms and Definitions

1.4.1 Acronyms

UC Use case

URS User Requirement Specification

SRS Software Requirement Specification

SDD Software Design Document

OS Operating System

VSE Very Small Entity

PM Project Management

SI Software Implementation

IID Iterative and Incremental Development

SCI Software Configuration Item

CRUD Create-Read-Update-Delete

**Chapter Two | Test Procedure**

2.1 Test Objectives

The objectives of EasyOPD System are

1. Identify the faults – Finding the fault which caused the failure.

2. Correct and remove the faults – Fixing errors and remove faults.

3. Ensure that all functionalities are implemented correctly.

2.2 Test Duration Progress Date and Duration

Progress I Perform date:17/05/2018 Duration days: 08/08/2018

Progress II Perform date: 09/08/2018 Duration days:29/08/2018

Final Progress Perform date: 23/09/2018 Duration days: 11/10/2018

2.3 Test Responsibility Item Responsibility

Unit test SW

Record unit test SW

System test SW

Record of system test SW

2.4 Test Strategy EasyOPD performed testing strategy by the following steps: 1. Design test cases

2. Prepare test data

3. Determine the expected result

4. Perform testing

5. Record all results

2.5 Result of Testing The results are separated into two parts:

1. Actual result: The actual results after testing

2. Pass/ Fail result:

a. Pass: The actual result which match with the expected result.

b. Fail: The actual result which does not match with the expected result.

**Chapter Three | Unit Test**

**Unit Test Case ID:** UT01

**Test Method**

signupForm(name: String, email: String, password: String, dob: String, gender: String)

**Test by:** Sirisak Wongwal

**Test case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** | **P/F** |
| **General user** | |
| 1 | All parameters valid form in the database. | {email: “[zmacktem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”, name: “tem”,  dob: “19/09/1997”,  gender: “male”} | | “Register successfully created. Please Sign In” |  |
| 2 | The account has already in database. | {email: “[zmacktem@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”, name: “tem”,  dob: “19/09/1997”,  gender: “male”} | | “Email is taken” |  |
| 3 | The password field is empty. | {email: “[zmacktem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”, name: “”,  dob: “19/09/1997”,  gender: “male”} | | “Password is required” |  |
| 4 | The password field is invalid form. | {email: “[zmacktem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “tem”,  name: “tem”,  dob: “19/09/1997”,  gender: “male”} | | “Password must contain at least 6 characters” |  |

**Unit Test Case ID:** UT02

**Test Method**

editUserForm(name: String, email: String, password: String, about: String)

**Test by:** Kittinut Saengsri

**Test case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** | **P/F** |
| **Registered user** | |
| 1 | All parameters valid form in the database and information in the database is changed. | {email: “[zmacktem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”, name: “tem”,  about: “Hello Tem”} | | Redirect to user’s profile page |  |
| 2 | The account is not correct format. | {email: “[zmacktem2hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”,  name: “tem”,  about: “Hello Tem”} | | “Email must contain @” |  |
| 3 | The password field is not match with email. | {email: “[zmacktem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “tem”,  name: “tem”,  about: “Hello Tem”} | | “Password must contain at least 6 characters” |  |
| 4 | The email field is empty. | {email: “”,  password: “temhere123”, name: “tem”,  about: “Hello Tem”} | | “This email field can not empty” |  |

**Unit Test Case ID:** UT03

**Test Method**

editPostForm(title: String, body: String, bodys: String)

**Test by:** Kittinut Saengsri

**Test case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** | **P/F** |
| **Registered user** | |
| 1 | All parameters valid form in the database and information in the database is changed. | {title:”Omelet”,  body:”1. Egg 2. Olive oil”, bodys: “1. Crack an egg into the bowl. 2. Mix egg together 3. Put it into the pan. 4. Flip when it looks good. 5. Then it ready to serve.”} | | Redirect the user to the user’s profile page. |  |
| 2 | The post form is not correct format title is empty. | {title:””,  body:”1. Egg 2. Olive oil”, bodys: “1. Crack an egg into the bowl. 2. Mix egg together 3. Put it into the pan. 4. Flip when it looks good. 5. Then it ready to serve.”} | | “All fields are required” |  |
| 3 | The post form is not correct format body is empty. | {title:”Omelet”,  body:””,  bodys: “1. Crack an egg into the bowl. 2. Mix egg together 3. Put it into the pan. 4. Flip when it looks good. 5. Then it ready to serve.”} | | “All fields are required” |  |
| 4 | The post form is not correct format bodys is empty. | {title:”Omelet”,  body:”1. Egg 2. Olive oil”, bodys: “”} | | “All fields are required” |  |

**Unit Test Case ID:** UT04

**Test Method**

newPostForm(title: String, body: String, bodys: String)

**Test by:** Kittinut Saengsri

**Test method**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** | **P/F** |
| **Registered user** | |
| 1 | All parameters valid form in the database and post information in the database. | {title:”Omelet”,  body:”1. Egg 2. Olive oil”, bodys: “1. Crack an egg into the bowl. 2. Mix egg together 3. Put it into the pan. 4. Flip when it looks good. 5. Then it ready to serve.”} | | Redirect the user to the user’s profile page. |  |
| 2 | The post form is not correct format title is empty. | {title:””,  body:”1. Egg 2. Olive oil”, bodys: “1. Crack an egg into the bowl. 2. Mix egg together 3. Put it into the pan. 4. Flip when it looks good. 5. Then it ready to serve.”} | | “All fields are required” |  |
| 3 | The post form is not correct format body is empty. | {title:”Omelet”,  body:””,  bodys: “1. Crack an egg into the bowl. 2. Mix egg together 3. Put it into the pan. 4. Flip when it looks good. 5. Then it ready to serve.”} | | “All fields are required” |  |
| 4 | The post form is not correct format bodys is empty. | {title:”Omelet”,  body:”1. Egg 2. Olive oil”, bodys: “”} | | “All fields are required” |  |

**Unit Test Case ID:** UT05

**Test method**

signinForm(email: String, password: String)

**Test by:** Kittinut Saengsri

**Test method**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** | **P/F** |
| **email** | **password** |
| 1 | The email and password are corrected in the database. | “zmack\_tem@hotmail.com” | “temhere123” | Redirect the user to the homepage. |  |
| 2 | The email and password do not exist in the database. | “zmack\_temhotmail.com” | “temhere123” | “User with that email does not exist. Please register” |  |
| 3 | The email and password do not exist in the database. | “” | “temhere123” | “User with that email does not exist. Please register” |  |
| 4 | The email and password do not exist in the database. | “zmacktem@hotmail.com” | “temhere123456” | “Email and password do not match” |  |

**Chapter Four | Appendix A**

Mock object

|  |  |
| --- | --- |
| **User** | **Return** |
| {email: “[zmacktem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”, name: “tem”,  about: “Hello Tem”} | “Success” |
| “” | null |

Mock object

|  |  |
| --- | --- |
| **Post** | **Return** |
| {title:”Omelet”,  body:”1. Egg 2. Olive oil”, bodys: “1. Crack an egg into the bowl. 2. Mix egg together 3. Put it into the pan. 4. Flip when it looks good. 5. Then it ready to serve.”} | “Success” |
| “” | null |