**Spotifood**

**Test plan**

**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**

Document History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Document name | Version | Status | Date | Viewable | Editable | Responsible |
| Spotifood\_TestPlan\_v0.1 | - Add chapter one  - Add chapter two  - Add chapter three  - Add chapter four | Draft | 07/10/2019 | SW, KS, PP | SW, KS | SW, KS |
| Spotifood\_TestPlan\_v1.0 | - Update chapter one  - Update chapter two  - Update chapter three  - Update chapter four | Draft | 07/29/2019 | SW, KS, PP | SW, KS | SW, KS |

\*SW = Sirisak Wongwal

\*KS = Kittinut Saengsri

\*PP = Dr.Passakorn Phannachitta

Table of Contents

[Chapter 1 | Introduction 4](#_Toc15336792)

[1.1 Identification 4](#_Toc15336793)

[1.2 Project Overview 4](#_Toc15336794)

[1.2.1 Purpose 4](#_Toc15336795)

[1.2.2 Scope 4](#_Toc15336796)

[1.3 Acronyms and Definitions 5](#_Toc15336797)

[1.3.1 Acronyms 5](#_Toc15336798)

[Chapter Two | Test Procedure 6](#_Toc15336799)

[2.1 Test Objectives 6](#_Toc15336800)

[2.2 Test Duration Progress Date and Duration 6](#_Toc15336801)

[2.3 Test Responsibility Item Responsibility 6](#_Toc15336802)

[2.4 Test Strategy Spotifood performed testing strategy by the following steps: 6](#_Toc15336803)

[2.5 Result of Testing The results are separated into two parts: 7](#_Toc15336804)

[Chapter Three | Unit Test 7](#_Toc15336805)

[2.1 Data schema 16](#_Toc15336806)

[2.2 Data dictionary 17](#_Toc15336807)

[CHAPTER FOUR | APPENDIX 18](#_Toc15336808)

[4.1 Appendix1 18](#_Toc15336809)

[4.2 Appendix2 18](#_Toc15336810)

# 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 Acronyms and Definitions

### 1.3.1 Acronyms

UC Use Case

URS User Requirement Specification

CRUD Create-Read-Update-Delete

ADS Advertising

ID Identification Data

UT Unit Test

# Chapter Two | Test Procedure

## 2.1 Test Objectives

The objectives of Spotifood 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: 07/10/2019 Duration days: 07/29/2019

## 2.3 Test Responsibility Item Responsibility

Unit test SW,KS

Record unit test SW,KS

System test SW,KS

Record of system test SW,KS

## 2.4 Test Strategy Spotifood 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 matches with the expected result.

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

# Chapter Three | Unit Test

**Unit Test ID:** UT01

**Function:** signUp

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

**Prepared data in the database:** Appendix1

**Test by:** Sirisak Wongwal

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description** | **Input** | **Expected Return** |
| **General user** |
| 1 | All parameters valid form in the database. | {email: “[zmack\_tem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”,  name: “Kittinut Saengsri”,  dob: “19/09/1997”,  gender: “male”} | “Register successfully created. Please Sign In” |
| 2 | The account has already in the database. | {email: “[zmack\_tem@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”,  name: “Kittinut Saengsri”,  dob: “19/09/1997”,  gender: “male”} | “Email is taken” |
| 3 | The password field is empty. | {email: “[zmack\_tem2@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: “[zmack\_tem2@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “tem”,  name: “Kittinut Saengsri”,  dob: “19/09/1997”,  gender: “male”} | “Password must contain at least 6 characters” |

**Unit Test ID:** UT02

**Function:** EditProfile

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

**Prepared data in the database:** Appendix1

**Test by:** Kittinut Saengsri

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

**Unit Test ID:** UT03

**Function:** EditPost

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

**Prepared data in the database:** Appendix2

**Test by:** Kittinut Saengsri

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description** | **Input** | **Expected Return** |
| **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 ID:** UT04

**Function:** NewPost

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

**Prepared data in the database:** Appendix2

**Test by:** Kittinut Saengsri

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description** | **Input** | **Expected Return** |
| **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 ID:** UT05

**Function:** Signin

signinForm(email: String, password: String)

**Prepared data in the database:** Appendix1

**Test by:** Kittinut Saengsri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** |
| **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. | “zmack\_tem@hotmail.com” | “temhere123456” | “Email and password do not match” |

**Unit Test ID:** UT06

**Function:** Editprofile

handleChange(name: String, fileSize: int)

**Prepared data in the database:** Appendix1

**Test by:** Sirisak Wongwal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** |
| **name** | **fileSize** |
| 1 | Set user data name and file size of photo. | “Kittinut Saengsri” | “0” | true |
| 2 | Set user data name and file size but not authorization . | “” | “0” | false |

**Unit Test ID:** UT07

**Function:** Profile

read(userId: String, token: hashed\_Password)

**Prepared data in the database:** Appendix3

**Test by:** Kitinut Saengsri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** |
| **userId** | **hashed\_Password** |
| 1 | User ID and token match. | “5d208546d6f6f23ee441c2fd” | “qiyh4XPJGsOZ2MEAyLkfWqeQ  ” | “redirectToSign: true” |
| 2 | User ID empty and token not empty. | “” | “qiyh4XPJGsOZ2MEAyLkfWqeQ” | “redirectToSign: false” |
| 3 | User ID not empty and token empty. | “5d208546d6f6f23ee441c2fd” | “” | “redirectToSign: false” |

**Unit Test ID:** UT08

**Function:** apiUser

remove(userId, token)

**Prepared data in the database:** Appendix3

**Test by:** Sirisak Wongwal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** |
| **userId** | **token** |
| 1 | Remove user in the database. | “5d208546d6f6f23ee441c2fd” | “855c3697d9979e78ac404c4ba2c66533” | Remove in the database. |
| 2 | Unauthorized user to remove the user. | “” | “” | Show error unauthorized. |

**Unit Test ID:** UT09

**Function:** Signin

signinForm(email: String, password: String)

**Prepared data in the database:** Appendix1

**Test by:** Kitinut Saengsri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** |
| **email** | **password** |
| 1 | Email and password match. | “[zmacktem@hotmail.com](mailto:zmacktem@hotmail.com)” | “temhere123” | “redirectToReferer: true” |
| 2 | Email and password not match. | “[zmacktem@hotmail.com](mailto:zmacktem@hotmail.com)” | “tem” | “redirectToReferer: false” |
| 3 | Email not empty and password empty. | “[zmacktem@hotmail.com](mailto:zmacktem@hotmail.com)” | “” | “redirectToReferer: false” |
| 4 | Email empty and password not empty. | “” | “temhere123” | “redirectToReferer: false” |

**Unit Test ID:** UT10

**Function:** Signup

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

**Prepared data in the database:** Appendix1

**Test by:** Kitinut Saengsri

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description** | **Input** | **Expected Return** |
| **signupForm** |
| 1 | Name, email, dob, gender not empty. | name: “Kittinut Saengsri”  email: “[zmack\_tem@hotmail.com](mailto:zmack_tem@hotmail.com)”  dob: “1997/09/19”  gender: “male” | Return true |
| 2 | Name empty. | name: “”  email: “[zmack\_tem@hotmail.com](mailto:zmack_tem@hotmail.com)”  dob: “1997/09/19”  gender: “male” | Return false |
| 3 | Email empty. | name: “Kittinut Saengsri”  email: “”  dob: “1997/09/19”  gender: “male” | Return false |
| 4 | Dob empty. | name: “Kittinut Saengsri”  email: “[zmack\_tem@hotmail.com](mailto:zmack_tem@hotmail.com)”  dob: “”  gender: “male” | Return false |
| 5 | Gender empty. | name: “Kittinut Saengsri”  email: “[zmack\_tem@hotmail.com](mailto:zmack_tem@hotmail.com)”  dob: “1997/09/19”  gender: “” | Return false |

**Unit Test ID:** UT11

**Function:** apiPost

singlePost(postId)

**Prepared data in the database:** Appendix1

**Test by:** Kitinut Saengsri

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description** | **Input** | **Expected Return** |
| **postId** |
| 1 | Get post from postId in the database. | “5d319cc13f5b8b0fc4038e07” | Get post |
| 2 | Invalid postId to get post in the database. | “555555” | Return false |

**Unit Test ID:** UT12

**Function:** apiPost

update(postId,post)

**Prepared data in the database:** Appendix1

**Test by:** Kitinut Saengsri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Input** | | **Expected Return** |
| **postId** | **post** |
| 1 | Update post information to the database. | “5d319cc13f5b8b0fc4038e07” | “Omelete(japan)” | Return true |
| 2 | Invalid postId to update post in the database. | “555555” | **“**Omelete(japan)**”** | Return false |

**Unit Test ID:** UT13

**Function:** EditProfile

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

**Prepared data in the database:** Appendix1

**Test by:** Kitinut Saengsri

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Description** | **Input** | **Expected Return** |
| **signupForm** |
| 1 | Name, email, dob, gender not empty. | name: “Kittinut Saengsri”  email: “[zmack\_tem@hotmail.com](mailto:zmack_tem@hotmail.com)”  password: “temhere123”  about: “Hi Kittinut Saengsri” | Return true |
| 2 | About empty. | name: “Kittinut Saengsri”  email: “[zmack\_tem@hotmail.com](mailto:zmack_tem@hotmail.com)”  password: “temhere123”  about: “” | Return true |
| 3 | Email empty. | name: “Kittinut Saengsri”  email: “”  password: “temhere123  about: “Hi Kittinut Saengsri” | Return false |
| 4 | Password empty. | name: “Kittinut Saengsri”  email: “[zmack\_tem@hotmail.com](mailto:zmack_tem@hotmail.com)”  password: “”  about: “Hi Kittinut Saengsri” | Return false |

## 2.1 Data schema

{“General user” : {

“Email” : String,

“Password” : String

},

“Registered user” : {

“user\_id” : String,

“Name” : String,

“Email” : String,

“Password” : String,

“Gender” : String,

“Dob” : String

}

}

## 2.2 Data dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Entity | Attribute | Data type | Description |
| General user | Email | String | The email of general user. |
| General user | Password | String | The password of general user. |
| Registered user | user\_id | String | Unique identification of user. |
| Registered user | Name | String | The name of registered user. |
| Registered user | Email | String | The email of registered user. |
| Registered user | Password | String | The password of registered user. |
| Registered user | Gender | String | The gender of registered user. |
| Registered user | Dob | String | Date of birth of registered user. |

# CHAPTER FOUR | APPENDIX

## 4.1 Appendix1

|  |
| --- |
| Authentication |
| {email: “[zmack\_tem@hotmail.com](mailto:zmacktem@hotmail.com)”, password: “temhere123”, name: “Kittinut Saengsri”, dob: “19/09/1997”, gender: “male” , hased\_password: “qiyh4XPJGsOZ2MEAyLkfWqeQ  ” , userId: “5d208546d6f6f23ee441c2fd” , token: “855c3697d9979e78ac404c4ba2c66533”, about: “Hi Kittinut Saengsri” } |
| “” |

## 4.2 Appendix2

|  |
| --- |
| Post |
| {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.” , postId: “5d319cc13f5b8b0fc4038e07”} |
| “” |