Lanna Cooking Recipe on Mobile
Test Plan

By

Name: Kantapong Orprayoon   **ID:** 532115008

Name: Khomkrib   Singh   **ID:** 532115015

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

_____

Mr. Kittitouch Suteeca

Test Plan

Cross Ref. VSE-29110	Coverage Level:	Version :
	V.3.0	V.3.0

Process Ownership	Approving Authority
CAMT	
Scope	**Approved Date**
SENIOR PROJECT	

Document History

Version Number	Record Date	Prepared/ Modified By	Reviewed By	Change Details
V.1.0	4/6/2013	KO,KS	KO,KS	
V.1.1	26/6/2013	KO,KS	KO,KS	- Edit scope of testing - Edit test environment - Edit unit test case
V.2.0	6/8/2013	KO,KS	KO,KS	- Edit Test Duration. - Edit unit test case of web application - Edit system test case of web application. - Input test case for testing process in all of feature of progress II.
V. 2.1	8/9/2013	KO, KS	KO,KS	- Add Acronym and Description of UTC and STC. - Edit name of class AdminsController to UsersController. - Edit the description and expected result of unit test case of web application. - Add false case in unit test case as both unit test of web application. - Edit system test of mobile application.
V. 3.0	2/10/2013	KO, KS	KO,KS	- Edit test duration. - Edit the description of unit test case of Mobile Application. - Edit system test case of Mobile Application. - Edit the input of web application and mobile application unit test case.

Document develop by

....

(Kantapong Orprayoon)

....

(Khomkrib Singh)

(3 October 2013)

Table of Content

Chapter One | Introduction

1.1   Objectives

The objectives of the test plan of Lanna Cooking Recipe on Mobile project is to establish test plan of the unit testing and system testing for make sure that the bugs or the defects are discovered and fixed. the unit testing covers all of implemented methods in the LCR system. The system testing cover the user requirements.

1.2   Scope

This test plan describes the white box-testing and black-box testing are activities to defect the defects in the system and describes the system testing activities for testing a completely integrated system to verify that it meets the user requirements.

1.3   Acronyms and Definitions

1.3.1 Acronyms

LCR = Lanna Cooking Recipe
PMP = Project Management Plan
SDD = Software Design Document
SRS = Software Requirment Specification
KO = Kantapong Orprayoon
KS = Khomkrib Singh
SRS = Software Requirement Specification
UC = Use Case
UI = User Interface
UTC = Unit Test Case
STC = System Test Case

1.3.2 Definitions

Feature   Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of a produce in the language of the product. Used for requirements analysis, design, coding, testing or maintenance. [IEEE90]

Design	The period of time in the software life cycle during which the designs for architecture, software component, interfaces and data are created, documented, and verified to satisfy requirements. [IEEE90]
IEEE	Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and for computer scientists. [IEEE90]
Requirement	(1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or processed by system or system component to satisfy a contract, standard, specification, or other formally imposed document. (3) A documented representation of a condition or capability as in definition (1) or (2). [IEEE90]
Specification	Precise description of an activity or work product which serves as basis or input for further activities or work product. A specification can comprise requirements to a product and how they will be solved. Different parts of a specification (e.g. what is to be done, how it will be done) must not be mixed. [IEEE90]
White box testing	Testing process that focus on internal structure. The tester should know the code inside the program and test it through the code and determines the appropriate outputs.
Black box testing	Process, device or system that focus on input, output and transfer characteristics without knowledge about it internal structure.
Unit Testing	A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the performs as designed.
System testing	A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

Chapter Two | Test Plan and Test Procedure

2.1 Test Objectives

The objective of testing Lanna Cooking Recipe on Mobile project are:

1. All bugs or defects are detected.
2. Those bugs or defects are fixed.
3. Meet user's requirements.
4. All functionalities and features must be there.

2.2 Scope

Lanna Cooking Recipe on Mobile will test by white-box testing and black-box testing techniques that are unit testing and system testing and record the test result in the test record.

2.3 Test Duration

Progress	Date and Duration
Progress Report I	**Perform date:** 16th May 2013 - 18th May 2013(Unit test, System test) **Duration:** 3 days
Progress Report II	**Perform date:** 26th July 2013 - 5th August 2013(Unit test, System test) **Duration:** 10 days
Progress Report III	**Perform date:** 7th September 2013 - 8 September 2013 (Unit test, System test) **Duration:** 2 days
Final Progress Report	**Perform date:** 1st October 2013 - 2rd October 2013 (Unit test, System test) **Duration:** 2 days

2.4 Test Responsibility

Item	Responsibility
Unit test of web application	KS
Unit test of android application	KO
Record unit test of web application	KS
Record unit test of android application	KO
System test of web application	KS
System test of android application	KO
Record system test of web application	KS
Record system test of android application	KO

2.5 Test Strategy

Lanna Cooking Recipe on Mobile test strategy will be follow by:

1. Design test case for each feature.
2. Prepare test data for each feature.
3. Determine expected results.
4. Perform testing on individual feature.
5. Result testing will be record.
6. All test files will be store in the project repository.

2.6 Result of Testing

In the test record the test result will separate into two parts which are:

1. Actual output: The actual outputs that are performed by each test case.
2. Pass/Fail criteria:
    2.1 Pass: the result of actual result is same like expected result.
    2.2 Fail: the result if actual result is not same like expected result.

2.7 Test Environment

2.7.1 Hardware

- Computers

MacBook Air 13-inch

- **Processor**: 1.8 GHz dual-core Intel Core i5 Turbo Boost up to 2.8 GHz
- **Memory:** 4GB, 128 GB flash storage
- **Graphics:** Intel HD Graphics 4000
- **Operating System:** Window 7 Ultimate

ASUS K55V Series

- **Processor:** Intel Core i7-3610QM 2.3 GHz
- **Memory:** 1 TB
- **Graphics:** NVIDIA Geforce GT 630M 2 GB
- **Operating System:** Window 7 Ultimate

2.7.2 Software

- Mozilla Firefox Version 20.0 or later
- Google Chrome Version 19.0 or later
- PHP editor Version 2.22 Beta 2
- Eclipse Juno Version 3.8.2
- Android ADT Version 21.1
- Android Mobile

Chapter Three | Unit Testing of LCR Web Application

Class UsersController

Unit Test Case 01 (UTC-01): login()

Description

　　Test the login() method to see that it can login and can't redirect to login page if using a wrong URL.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to .../users/login and login to web application	".../users/login"	Pass
2	This method can't redirect to users/login by using a wrong URL: .../user/login	".../user/login"	Fail

Unit Test Case 02 (UTC-02): logout()

Description

　　Test the logout() method to see that it can logout and redirect to login page.

Test Cases

ID	Description	Input	Expected Result
1	This method can logout and redirect to login page.	".../users/logout"	Pass

Class RecipeDetailController

Unit Test Case 03 (UTC-03): index()

Description

Test the index() method to see that it can redirect to index page of recipe detail and it can't redirect if using wrong url.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the index page of recipe detail by using URL: .../recipeDetails/index.	".../recipeDetails/index"	Pass
2	This method using wrong URL: .../recipeDetail/index and it can't redirect to index page.	".../recipeDetail/index"	Fail

Unit Test Case 04 (UTC-04): add()

Description

Test the add() method to see that it can redirect to add recipe detail page and it can't redirect if using wrong URL.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the add page of recipe detail by using URL: .../recipeDetails/add	".../recipeDetails/index "	Pass
2	This method using wrong URL: .../recipeDetail/add and it can't redirect to add page.	".../recipeDetail/index"	Fail

Unit Test Case 05(UTC-05): view($recipe_id:int)

Description

   Test the view($recipe_id) method to see that it can redirect to view page of recipe detail follow by id that input and if use wrong recipe id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of recipe detail show the information of recipe detail by using recipe id = 1, URL: .../recipeDetails/view/1	$recipe_id = 1	Pass
2	The method can't redirect to view page of recipe detail by using the recipe id that not have in database, recipe id = 0 URL: .../recipeDetails/view/0	$recipe_id = 0	Fail

Unit Test Case 06 (UTC-06) edit($recipe_id:int)

Description

   Test the edit($recipe_id) method to see that it can redirect to edit page of recipe detail follow by id that input and if use wrong recipe id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of recipe detail show the information of recipe detail from database by using recipe id = 1, URL: .../recipeDetails/edit/1	$recipe_id = 1	Pass
2	The method can't redirect to view page of recipe detail by using the recipe id that not have in database, recipe id = 0 URL: .../recipeDetails/edit/0	$recipe_id = 0	Fail

Unit Test Case 07 (UTC-07): delete($recipe_id:int)

Description

       Test the delete($recipe_id) method to see that it can delete a recipe detail follow by recipe id that input and it can't delete if use wrong recipe id.

Test Cases

ID	Description	Input	Expected Result
1	The method can delete recipe detail by using recipe id = 1, URL: .../recipeDetails/delete/1	$recipe_id = 1	Pass
2	The method can't delete recipe detail by using the recipe id that not have in database, recipe id = 0 URL: .../recipeDetails/delete/0	$recipe_id = 0	Fail

Class IngredientDetilsController

Unit Test Case 08 (UTC-08): index()

Description

       Test the index() method to see that it can redirect to index page of ingredient detail and it can't redirect if using wrong url.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the index page of ingredient detail by using URL: .../ingredientDetails/index.	".../ingredientDetails/index"	Pass
2	This method using wrong URL: .../ingredeintDetail/index and it can't redirect to index page.	".../ingredientDetail/index"	Fail

Unit Test Case 09 (UTC-09): add()

Description

Test the add() method to see that it can redirect to add ingredient detail page and it can't redirect if using wrong URL.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the add page of ingredient detail by using URL: .../ingredientDetails /add	".../ingredientDetails/add "	Pass
2	This method using wrong URL: .../ingredientDetail /add and it can't redirect to add page.	".../ingredientDetail/add"	Fail

Unit Test Case 10 (UTC-10): view($ingredient_id:int)

Description

Test the view($ingredient_id) method to see that it can redirect to view page of ingredient detail follow by id that input and if use wrong ingredient detail id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of ingredient detail show the information of ingredient detail by using ingredient id = 1, URL: .../ingredientDetails /view/1	$ingredient_id = 1	Pass
2	The method can't redirect to view page of ingredient detail by using the ingredient id that not have in database, ingredinet id = 0 URL: .../ingredientDetails/view/0	$ingredient_id = 0	Fail

Unit Test Case 11 (UTC-11): edit($ingredient_id:int)

Description

   Test the edit($ingredient_id) method to see that it can redirect to edit page of ingredient detail follow by id that input and if use wrong ingredient id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of ingredient detail show the information of ingredient detail from database by using ingredient id = 1, URL: .../ingredientDetails/edit/1	$ingredient_id = 1	Pass
2	The method can't redirect to view page of ingredient detail by using the ingredient id that not have in database, ingredient id = 0 URL: .../ingredientDetails/edit/0	$ingredient_id = 0	Fail

Unit Test Case 12 (UTC-12): delete($ingredient_id:int)

Description

   Test the delete($ingredient_id) method to see that it can delete a ingredient detail follow by ingredient id that input and it can't delete if use wrong ingredient id.

Test Cases

ID	Description	Input	Expected Result
1	The method can delete ingredient detail by using ingredient id = 1, URL: .../ingredientDetails/delete/1	$ingredient_id = 1	Pass
2	The method can't delete ingredient detail by using the ingredient id that not have in database, ingreident id = 0 URL: .../ingredientDetails/delete/0	$ingredient_id = 0	Fail

Class IngredientItemsController

Unit Test Case 13 (UTC-13): index()

Description

      Test the index() method to see that it can redirect to index page of ingredient item and it can't redirect if using wrong url.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the index page of ingredient item by using URL: .../ingredientItems/index.	".../ingredientItems/index"	Pass
2	This method using wrong URL: .../ingredientItem/index and it can't redirect to index page.	".../ingredientItem/index"	Fail

Unit Test Case 14 (UTC-14): add()

Description

      Test the add() method to see that it can redirect to add ingredient item page and it can't redirect if using wrong URL.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the add page of ingredient item by using URL: .../ingredientItems/add	".../ingredientItems/add"	Pass
2	This method using wrong URL: .../ingredientItem/add and it can't redirect to add page.	".../ingredientItem/add"	Fail

Unit Test Case 15 (UTC-15): delete($id:int)

Description

      Test the delete($id) method to see that it can delete a ingredient item follow by id that input and it can't delete if use wrong id.

Test Cases

ID	Description	Input	Expected Result
1	The method can delete ingredient item by using id = 1, URL: .../ingredientItems/delete/1	$id = 1	Pass
2	The method can't delete ingredient item by using the id that not have in database, id = 0 URL: .../ingredientItems/delete/0	$id = 0	Fail

Class CookingMethodsController

Unit Test Case 16 (UTC-16): index()

Description

       Test the index() method to see that it can redirect to index page of cooking method and it can't redirect if using wrong url.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the index page of cooking method by using URL: .../cookingMethods/index.	".../cookingMethods/index"	Pass
2	This method using wrong URL: .../cookingMethod/index and it can't redirect to index page.	".../cookingMethod/index"	Fail

Unit Test Case 17 (UTC-17): add()

Description

       Test the add() method to see that it can redirect to add cooking method page and it can't redirect if using wrong URL.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the add page of cooking method by using URL: .../cookingMethods/add	".../cookingMethods/add"	Pass
2	This method using wrong URL: .../cookingMethod/add and it can't redirect to add page.	".../cookingMethod/add"	Fail

Unit Test Case 18(UTC-18): view($id:int)

Description

      Test the view($id) method to see that it can redirect to view page of cooking method follow by id that input and if use wrong id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of cooking method show the information of cooking method by using id = 1, URL: .../cookingMethods /view/1	$id = 1	Pass
2	The method can't redirect to view page of cooking method by using the d that not have in database, id = 0 URL: .../cookingMethods/view/0	$id = 0	Fail

Unit Test Case 19 (UTC-19): edit($id:int)

Description

      Test the edit($id) method to see that it can redirect to edit page of cooking method follow by id that input and if use wrong id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of cooking method show the information of cooking method from database by using id = 1, URL: .../cookingMethods/edit/1	$id = 1	Pass
2	The method can't redirect to view page of cooking method by using the id that not have in database, id = 0 URL: .../cookingMethods/edit/0	$id = 0	Fail

Unit Test Case 20 (UTC-20): delete($id:int)

Description

       Test the delete($id) method to see that it can delete a cooking method follow by id that input and it can't delete if use wrong id.

Test Cases

ID	Description	Input	Expected Result
1	The method can delete cooking method by using id = 1, URL: .../cookingMethods/delete/1	$id = 1	Pass
2	The method can't delete cooking method by using the id that not have in database, id = 0 URL: .../cookingMethods/delete/0	$id = 0	Fail

Class ShopsController

Unit Test Case 21 (UTC-21): index()

Description

       Test the index() method to see that it can redirect to index page of shop and it can't redirect if using wrong url.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the index page of shop by using URL: .../shops/index.	".../shops/index"	Pass
2	This method using wrong URL: .../shops/index and it can't redirect to index page.	".../shop/index"	Fail

Unit Test Case 22 (UTC-22): add()

Description

      Test the add() method to see that it can redirect to add shop page and it can't redirect if using wrong URL.

Test Cases

ID	Description	Input	Expected Result
1	This method can redirect to the add page of shop by using URL: .../shops/add	".../shops/add"	Pass
2	This method using wrong URL: .../shops/add and it can't redirect to add page.	".../shop/add"	Fail

Unit Test Case 23(UTC-23): view($id:int)

Description

      Test the view($id) method to see that it can redirect to view page of shop follow by id that input and if use wrong id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of shop show the information of cooking method by using id = 1, URL: .../shops /view/1	$id = 1	Pass
2	The method can't redirect to view page of shop by using the d that not have in database, id = 0 URL: .../shops/view/0	$id = 0	Fail

Unit Test Case 24 (UTC-24): edit($id:int)

Description

　　Test the edit($id) method to see that it can redirect to edit page of shop follow by id that input and if use wrong id, it can't redirects.

Test Cases

ID	Description	Input	Expected Result
1	The method can redirect to view page of shop show the information of shop from database by using id = 1, URL: .../shops/edit/1	$id = 1	Pass
2	The method can't redirect to view page of shop by using the id that not have in database, id = 0 URL: .../shops/edit/0	$id = 0	Fail

Unit Test Case 25 (UTC-25): delete($id:int)

Description

　　Test the delete($id) method to see that it can delete a shop follow by id that input and it can't delete if use wrong id.

Test Cases

ID	Description	Input	Expected Result
1	The method can delete shop by using id = 1, URL: .../shops/delete/1	$id = 1	Pass
2	The method can't delete shop by using the id that not have in database, id = 0 URL: .../shops/delete/0	$id = 0	Fail

Chapter Four | Unit Testing of LCR Android Application

Unit Test Case 26 (UTC-26): LoginActivity

Description

       Test class "LoginActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "LoginActivity" shows a user interface that match with user interface design id 07.	"LoginActivity"	UI-07

Unit Test Case 27 (UTC-27): RegisterActivity

Description

       Test class "RegisterActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "RegisterActivity" shows a user interface that match with user interface design id 08.	"RegisterActivity"	UI-08

Unit Test Case 28 (UTC-28): HomeActivity

Description

      Test class "HomeActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "HomeActivity" shows a user interface that match with user interface design id 09.	"HomeActivity"	UI-09

Unit Test Case 29 (UTC-29): RecipeListViewAdapter

Description

      Test class "RecipeListViewAdapter " to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class " RecipeListViewAdapter " shows a user interface that match with user interface design id 10.	"RecipeListViewAdapter"	UI-10

Unit Test Case 30 (UTC-30): RecipeSingleItemView

Description

      Test class " RecipeSingleItemView " to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "RecipeSingleItemView" shows a user interface that match with user interface design id 11.	"RecipeSingleItemView"	UI-11

Unit Test Case 31 (UTC-31): RecipeCookingMethodActivity

Description

Test class "RecipeCookingMethodActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "RecipeCookingMethodActivity " shows a user interface that match with user interface design id 12.	"RecipeCookingMethodActivity"	UI-12

Unit Test Case 32 (UTC-32): IngredientListViewAdapter

Description

Test class " IngredientListViewAdapter " to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "IngredientListViewAdapter" shows a user interface that match with user interface design id 13.	"IngredientListViewAdapter"	UI-13

Unit Test Case 33 (UTC-33): IngredientSingleItemView

Description

Test class " IngredientSingleItemView " to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "IngredientDetailMainActivity" shows a user interface that match with user interface design id 14.	"IngredientDetailMainActivity"	UI-14

Unit Test Case 34 (UTC-34): IngredientDetailMainActivity

Description

Test class "IngredientDetailMainActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "IngredientDetailMainActivity" shows a user interface that match with user interface design id 15.	"IngredientMainActivity"	UI-15

Unit Test Case 35 (UTC-35): IngredientFindingActivity

Description

Test class "IngredientFindingActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "IngredientFindingActivity" shows a user interface that match with user interface design id 16.	"IngredientFindingActivity"	UI-16

Unit Test Case 36 (UTC-36): ShopDescriptionActivity

Description

Test class "ShopDescriptionActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class " ShopDescriptionActivity " shows a user interface that match with user interface design id 17.	"ShopDescriptionActivity"	UI-17

Unit Test Case 37 (UTC-37): ShopDescriptionActivity

Description

      Test class "ShopDescriptionActivity" to measure that it displays the user interface and match with the user interface design after user taps on get direction button or not.

Test Cases

ID	Description	Input	Expected Result
1	Class " ShopDescriptionActivity " shows a user interface that match with user interface design id 18.	"ShopDescriptionActivity"	UI-18

Unit Test Case 38 (UTC-38): RandomMainActivity

Description

      Test class "RandomMainActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "RandomMainActivity" shows a user interface that match with user interface design id 19.	"RandomMainActivity"	UI-19

Unit Test Case 39(UTC-39): RandomActionActivity

Description

      Test class "RandomActionActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Inpiut	Expected Result
1	Class "RandomActionActivity" shows a user interface that match with user interface design id 20.	"RandomActionActivity"	UI-20

Unit Test Case 40 (UTC-40): RandomDescriptionActivity

Description

Test class "RandomDescriptionActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "RandomActionActivity" shows a user interface that match with user interface design id 21.	"RandomActionActivity"	UI-21

Unit Test Case 41 (UTC-41): ShoppinglistActivity

Description

Test class "ShoppinglistActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "ShoppinglistActivity" shows a user interface that match with user interface design id 22.	"ShoppinglistActivity"	UI-22

Unit Test Case 42 (UTC-42) ShoppinglistDescription

Description

Test class "ShoppinglistDescription" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "ShoppinglistDescription" shows a user interface that match with user interface design id 23.	"ShoppinglistDescription"	UI-23

Unit Test Case 43 (UTC-43): SettingActivity

Description

      Test class "SettingActivity" to measure that it displays the user interface and match with the user interface design or not.

Test Cases

ID	Description	Input	Expected Result
1	Class "SettingActivity" shows a user interface that match with user interface design id 24.	"SettingActivity"	UI-24

Chapter Five | System Testing of LCR Web Application

System Test Case 01 (STC-01): Admin can access to the web application.

Description

This test case is a system testing for URS-17. Administrator can use login to access into the web application.

Prerequisite

- Username "admin" must be added already in Admin table.
- Password "admin" must be added already in Admin table.

Test Script

1. Admin inputs their username and password.
2. Admin clicks login button.
3. The system will verify with database.
4. The system allows the admin access to index page.

Test Cases

ID	Description	Input	Expected Result
1	Input invalid username and password.	username = abcd password = 1234	Alert message, Invalid username or password.
2	Input some fields.	username = null password = admin	Alert message, Invalid username or password.
3	Input correct username and password.	username = admin password = admin	Can access to index page of web service.
4	Not enter both username and password	username = null password = null	Alert message, can't access to index page.

System Test Case 02 (STC-02): Admin can logout form web application.

Description

   This test case is a system testing for URS-18 Administrator can use logout to exit from the web application.

Prerequisite

- Admin already login.

Test Script

1. Admin clicks logout.
2. Admin logging out of system.

Test Cases

ID	Description	Input	Expected Result
1	Click logout.	logout	The admin logging out from system and redirect to login page.

System Test Case 03 (STC-03): Admin can manage food recipe information.

Description

This test case is a system testing for URS-19 Administrator can manage recipe detail.

Prerequisite

- Admin already login.
- Admin stays on index page of recipe detail.

Test Script

1. Admin enters recipe index page.
2. Admin clicks add button.
3. Admin inputs a recipe detail.
4. Admin clicks save button.
5. The system validates the recipe detail.
6. The system add the recipe detail into database.
7. Admin clicks edit button.
8. Admin edits recipe detail.
9. Admin clicks save button.
10. The system validates the recipe detail.
11. The system updates the recipe detail in database.
12. Admin clicks delete button.
13. The system deletes the recipe detail in database.

Test Cases

ID	Description	Input	Expected Result
1	Input all required field in add recipe detail form with all correct information.	RecipeName = Keang hangle Description = This is a Northern Thailand food. Category = keang Img = kaeng hangle.jpg	The system will shows success message and add recipe detail to database.
2	Input all required field in add recipe detail form with invalid image type.	RecipeName = Keang hangle Description = This is a Northern Thailand food. Category = keang Img = kaeng hangle.gif	The system will shows alert message and a recipe detail does not added to database.
3	Input some required field in add recipe detail form.	RecipeName = Keang hangle Description = null Category = keang Img = kaeng hangle.jpg	The system will shows alert message and a recipe detail does not added to database.
4	Click edit button as recipe detail table.	-	The system will shows edit recipe detail page and recipe detail information from database.
5	Edit recipe detail with all correct information.	RecipeName = Namphrik Num Description = This is a Northern Thailand food. Category = numphrik Img = numphrik num.jpg	The system will shows success message and update recipe detail to database
6	Edit recipe detail with does not input recipe detail image.	RecipeName = Namphrik Num Description = This is a Northern Thailand food.	The system will shows success message and update recipe detail to database and old

		Category = numphrik Img = null	image does not change.
7	Edit recipe detail with does not edit some recipe detail.	RecipeName = Namphrik Num(old detail) Description = This is a Northern Thailand food and it very delicious. Catedory = numphrik (old detail) Img = null	The system will shows success message and update recipe detail to database without update the data that does not edited.
8	Click delete button on option column.	-	The system will shows success message and the recipe detail that selected will deleted from database.

System Test Case 04 (STC-04): Admin can mange ingredient detail.

Description

        This test case is a system testing for URS-20 Administrator can manage ingredient detail.

Prerequisite

- Admin already login.
- Admin stays on index page of ingredient detail.

Test Script

1. Admin enters add ingredient detail page.
2. Admin inputs a ingredient detail.
3. Admin clicks save button.
4. The system validates the ingredient detail.
5. The system adds the ingredient detail into database.
6. Admin clicks edit button.
7. Admin edits ingredient detail.
8. Admin clicks save button.
9. The system validates the ingredient detail.
10. The system updates the ingredient detail in database.
11. Admin clicks delete button.
12. The system deletes the ingredient detail in database.

Test Cases

ID	Description	Input	Expected Result
1	Input all field in add ingredient detail form with all correct information.	IngredientName = Pork Description = This is a pork. Type = meat Img = pork.jpg	The system will shows success message and add ingredient detail to database.
2	Input all required field in add ingredient detail form with all correct information.	IngredientName = Pork Description = This is a pork. Type = meat Img = pork.jpg	The system will shows success message and add ingredient detail to database.
3	Input all required field in add ingredient detail form with invalid image type.	IngredientName = Pork Description = This is a pork. Type = meat Img = pork.gif	The system will shows alert message and a ingredient detail does not added to database.
4	Input some required field in add ingredient detail form.	IngredientName = null Description = This is a pork. Type = meat Img = pork.gif	The system will shows alert message and a ingredient detail does not added to database.
5	Click edit button as ingredient detail table	-	The system will shows edit ingredient detail page and show ingredient detail information from database.
6	Edit ingredient detail with all correct	IngredientName = Chili Description = This is a chili.	The system will shows success message and update ingredient detail to database

		Type = spice Img = chili.jpg	
7	Edit ingredient detail with does not input ingredient detail image.	IngredientName = Chili Description = This is a chili. Type = spice Img = null	The system will shows success message and update ingredient detail to database and old image does not change.
8	Edit ingredient detail with does not edit some ingredient detail.	IngredientName = Chili Description = This is a chili and it very spicy. Type = spice Img = null	The system will shows success message and update ingredient detail to database without update the data that does not edited.
9	Click delete button on option column	-	The system will shows success message and the ingredient detail that selected will deleted from database.

System Test Case 05 (STC-05): Admin can mange ingredient for each menu.

Description

        This test case is a system testing for URS-21 Administrator can manage ingredient for each menu.

Prerequisite

- Admin already login.
- Admin stays on index page of ingredient item.

Test Script

1. Admin enters add ingredient item page.
2. Admin inputs a ingredient item.
3. Admin clicks save button.
4. The system validates the ingredient item.
5. The system adds the ingredient item into database.
6. Admin clicks edit button.
7. Admin edits ingredient item.
8. Admin clicks save button.
9. The system validates the ingredient item.
10. The system updates the ingredient item in database.
11. Admin clicks delete button.
12. The system deletes the ingredient item in database.

ID	Description	Input	Expected Result
1	Input all required field in add ingredient form with all correct information	RecipeID = 1 IngredientID = 1 Amount = 300 Unit = Gm.	The system will shows success message and add ingredient to database.
2	Input some required field in add ingredient form with all correct information	RecipeID = 2 IngredientID = 1 Amount = 300 Unit = null	The system will shows alert message and it does not add ingredient to database.
3	Click edit button as ingredient item table.	-	The system redirects to ingredient item edit page and shows ingredient item data from database.
4	Edit ingredient id and amount.	RecipeID = 1(old data) IngredientID = 2 Amount = 15 Unit = Gm. (old data)	The system will shows success message and add ingredient to database.
5	Click delete button on option column.	-	The system will show success message and delete ingredient that selected in database.

System Test Case 06 (STC-06): Admin can mange cooking method.

Description

This test case is a system testing for URS-22 Administrator can manage cooking method

Prerequisite

- Admin already login.
- Admin stays on index page of cooking method.

Test Script

1. Admin enters add cooking method page.
2. Admin inputs a cooking method.
3. Admin clicks save button.
4. The system validates the cooking method.
5. The system adds the cooking method into database.
6. Admin clicks edit button.
7. Admin edits cooking method.
8. Admin clicks save button.
9. The system validates the cooking method.
10. The system updates the cooking method in database.
11. Admin clicks delete button.
12. The system deletes the cooking method in database.

Test Cases

ID	Description	Input	Expected Result
1	Input all required field in add cooking method form with all correct information	RecipeID = 1 Method Number = 1 Instruction = Slice an onion.	The system will shows success message and add cooking method to database.
2	Input all required field in add cooking method form with invalid image type.	RecipeID = 1 Method Number = 1 Instruction = Slice an onion.	The system will shows alert message and a cooking method does not added to database.
3	Input some required field in add cooking method form.	RecipeID = 1 Method Number = 1 Instruction = null	The system will shows alert message and a cooking method does not added to database.
4	Click edit button as cooking method table	-	The system will shows edit cooking method page and show cooking method information from database.
5	Edit cooking method with all correct information.	Method Number = 2 Instruction = Slice a pork.	The system will shows success message and update cooking method to database
6	Edit cooking method with does not input cooking method image.	Method Number = null Instruction = Slice a pork.	The system will shows success message and update cooking method to database and old image does not change.

7	Edit cooking method with does not edit some cooking method.	Method Number = 3 Instruction = Slice a pork.	The system will shows success message and update cooking method detail to database without update the data that does not edited.
8	Click delete button on option column	-	The system will shows success message and the cooking method that selected will deleted from database.

System Test Case 07 (STC-07): Admin can mange ingredient store location.

Description

       This test case is a system testing for URS-23. Admin can mange finding ingredient store location.

Prerequisite

- Admin already login.
- Admin stays on index page of shop.

Test Script

13. Admin enters add shop page.
14. Admin inputs a shop.
15. Admin clicks save button.
16. The system validates the shop.
17. The system adds the shop into database.
18. Admin clicks edit button.
19. Admin edits shop.
20. Admin clicks save button.
21. The system validates the shop.
22. The system updates the shop in database.
23. Admin clicks delete button.
24. The system deletes the shop in database.

Test Cases

ID	Description	Input	Expected Result
1	Input all required field in add shop form with all correct information	Name = Test1 Description = This is a test1 Sell Ingredient = Hangle powder Address = Tumbol Suthep Street = Suthep City = Chiang Mai State = Thailand Zip = 50200 Latitude = 18.7993 Longitude = 98.9521 Image = ton pa yom market.jpeg	The system will shows success message and add shop location to database.
2	Input all required field in add shop form with not input latitude.	Name = Test1 Description = This is a test1 Sell Ingredient = Hangle powder Address = Tumbol Suthep Street = Suthep City = Chiang Mai State = Thailand Zip = 50200 Latitude = null Longitude = 98.9521 Image = ton pa yom market.jpeg	The system will shows alert message and a shop location does not added to database.
3	Input all required field in add shop form with not input Description.	Name = Test1 Description = null Sell Ingredient = Hangle powder Address = Tumbol Suthep Street = Suthep	The system will shows alert message and a shop location does not added to database.

		City = Chiang Mai State = Thailand Zip = 50200 Latitude = 18.7993 Longitude = 98.9521 Image = ton pa yom market.jpeg	
4	Click edit button as shop table	-	The system will shows edit shop page and show shop information from database.
5	Edit shop with all correct information.	Name = Test2 Description = This is a test2 Sell Ingredient = Hangle powder Address = Tumbol Suthep 2 Street = Suthep 2 City = Chiang Mai State = Thailand Zip = 50200 Latitude = 18.7994 Longitude = 98.9522 Image = ton pa yom market_2.jpeg	The system will shows success message and update cooking method to database
6	Click delete button on option column	-	The system will shows success message and the cooking method that selected will deleted from database.

Chapter Six | System Testing of LCR Android Application

System Test Case 08 (STC-08): User can register with application.

Description

      This test case is a system testing for URS-01. User can use registration to be member before access into the application.

Prerequisite

- E-mail "user@gmail.com" must be added already in Users table.
- Name "user" must be add already in Users table.
- User never registered in this application.

Test Script

1. User taps sign in button.
2. User inputs their information in the registration form.
3. The system validates the registration information.
4. The system adds the user information into database.

Test Cases

ID	Description	Input	Expected Result
1	Input all required field in the registration form with correct information.	E-mail = mo@gmail.com Password = 1234 Name = mo	Show successful message and the user information adds to database and the system will redirects to login page.
2	Input all required field in the registration form with the exiting email.	E-mail = user@gmail.com Password = 1234 Name = mo	Show alert message and the system will allows the user to input their information again.
3	Input all required field in the registration form with the exiting name.	E-mail = mo@gmail.com Password = 1234 Name = user	Show alert message and the system will allows the user to input their information again.
4	Input some required field in the registration form with correct information.	E-mail = mo@gmail.com Password = 1234 Name = null	Show alert message and the system will allows the user to input their information again.

System Test Case 09 (STC-09): User can use login to application.

Description

       This test case is a system testing for URS-02. User can use login to access into the application..

Prerequisite

- User must registered before logging into the application.
- E-mail = "mo@gmail.com" must be added already in Users table.
- Password = "1234" must be added already in Users table.

Test Script

1. User inputs their e-mail and password.
2. User click login button.
3. The system validates e-mail and password.
4. User logging in to application.

Test Cases

ID	Description	Input	Expected Result
1	Input both correct e-mail and password into login form.	E-mail = mo@gmail.com Password = 1234	Show successful message and the system will allows the user to home page.
2	Input incorrect e-mail and correct password into login form.	E-mail = a@gmail.com Password = 1234	Show alert message and redirect to login page.
3	Input correct e-mail and incorrect password into login form.	E-mail = mo@gmail.com Password = 23048	Show alert message and redirect to login page.
4	Not input both e-mail and password.	E-mail = null Password = null	Show alert message and redirect to login page.

System Test Case 10 (STC-10): User can logout from the application.

Description

This test case is a system testing for URS-03. User can logout to exit from the application.

Prerequisite

- User must logging into the application.

Test Script

1. User taps logout in setting menu.
2. User logging out from the system.

Test Cases

ID	Description	Input	Expected Result
1	Tap on logout in setting page.	-	User can logout from the system.

System Test Case 11 (STC-11): User views list of recipe.

Description

This test case is a system testing for URS-04. User can views list of recipe.

Prerequisite

- User must login into the application.

Test Script

1. User clicks on see all recipe button.
2. The system will shows list of recipe details.

Test Cases

ID	Description	Input	Expected Result
1	Click on all recipe button and the system will show: - recipe detail id - recipe detail name - mini image of recipe. Of each recipe that have in database.	-	The system redirects to list of recipe detail page and shows list of recipe details from all recipe that have in database, that consist of - recipe detail id - recipe detail name - mini image of recipe.

System Test Case 12 (STC-12): User views information of recipe.

Description

      This test case is a system testing for URS-05. User can views information of recipe**.**

Prerequisite

-    User must login into the application.
-    User click on see all recipe button.

Test Script

    1. User clicks on any recipe name.
    2. The system will shows information of the recipe that clicked.

Test Cases

ID	Description	Input	Expected Result
1	Click on any recipe name and the system will show: - recipe id - recipe name - recipe description - Image of recipe - Function button are: - Ingredient button - Cooking method button - Facebook button - Share Facebook button	-	The system redirects to the recipe information page of each recipe name that clicked. Its consist of - recipe id - recipe name - recipe description - Image of recipe - Function button are: - Ingredient button - Cooking method button - Facebook button - Share Facebook button
2	Click on ingredient button and the system will show - ingredient id - ingredient name - mini ingredient image Of each ingredient that have in each recipe.	-	The system shows a list of recipe ingredient's. Its consist of - ingredient id - ingredient name - mini ingredient image
3	Click on cooking method and the system will show - cooking method step's - cooking method instruction's	-	The system shows recipe cooking methods. Its consist of - cooking method step's - cooking method instruction's

System Test Case 13 (STC-13): User adds menu to shopping list.

Description

This test case is a system testing for URS-06. User can views add menu to shopping list**.**

Prerequisite

- User must login into the application.
- User taps on see all recipe button.
- User taps on any menu on screen.

Test Script

1. User taps on add to list button.
2. User taps on shopping list button.
3. The System will show the menu that user selected.

Test Cases

ID	Description	Input	Expected Result
1	Tap on add to list button. The system will show a success message.	-	The system will show a success message.
2	Tap on shopping list button. The system will redirect to shopping list page and show a menu id that selected. e.g. Recipe ID: 1	-	The system will show a menu id from that selected. Recipe ID: 1
3	Tap on other menu and add to shopping list again.	-	The system will show a success message.
4	Tap on shopping list button. The system will redirect to shopping list page and show a list of menu that selected e.g. Recipe ID: 1 Recipe ID: 5	-	The system will show a list of menu id that selected. Recipe ID: 1 Recipe ID: 5

System Test Case 14 (STC-14): User deletes menu from shopping list.

Description

       This test case is a system testing for URS-7. User can delete menu from shopping list.

Prerequisite

- User must login into the application.
- User must adds any menu to shopping list.

Test Script

1. User taps on needed recipe id on shopping list page.
2. User taps on delete menu button.
3. The system will delete the menu that selected.

Test Cases

ID	Description	Input	Expected Result
1	Tap on needed recipe id on shopping list page to measure that the system can show list of menu ingredient consist of name, amount, and show delete menu button.	-	The system will show the list of menu ingredient that consist of name and amount of menu ingredient including show delete menu button.
2	Tap on delete menu button to measure that the system can delete menu from shopping list and redirect to shopping list page.	-	The system can delete menu from shopping list and redirect to shopping list page.

System Test Case 15 (STC-15): User views list of menu in shopping list.

Description

      This test case is a system testing for URS-8. User can view list of menu in shopping list.

Prerequisite

- User must login into the application.
- User must adds any menu to shopping list.

Test Script

1. User taps on a shopping list button.
2. The system will show the list of menu that selected.

Test Cases

ID	Description	Input	Expected Result
1	Tap on a shopping list button to measure that the system can show the list of menu that selected from the recipe description page. It will show as a menu id.	-	The system can show the list of menu that selected.

System Test Case 16 (STC-16): User views information of recipe in shopping list.

Description

      This test case is a system testing for URS-9. User can view information of recipe in shopping list

Prerequisite

- User must login into the application.
- User must adds any menu to shopping list.

Test Script

1. User taps on needed recipe id on shopping list page.
2. The system will show a ingredient information of each menu that selected.

Test Cases

ID	Description	Input	Expected Result
1	Tap on needed menu id on shopping list page to measure that the system can show a list of menu ingredient information. It will show name and amount of menu ingredient.	-	The system can show the menu ingredient information when tap on any menu id from shopping list page.

System Test Case 17 (STC-17): User shake device for random food menu.

Description

       This test case is a system testing for URS-10. User can shake a device for random food menu.

Prerequisite

- User must login into the application.

Test Script

1. User taps on a random button.
2. User shakes a device to get a random food menu.

Test Cases

ID	Description	Input	Expected Result
1	Tap on a random button to measure that the system can redirect to random page. It will show a random button.	-	The system can redirect to random page and show a random button.
2	Tap on a random button on random page to measure that the system can display an image to allow user can start shaking.	-	The system can display an image to allow user can start shaking.
3	Shaking a device to get a menu. The System will random menu from database and display it on screen. It will show menu detail such as menu name and menu description.	-	The system can show a random menu and menu detail.

System Test Case 18 (STC-18): User shares menu on Facebook.

Description

      This test case is a system testing for URS-11. User can sharing menu on Facebook social network.

Prerequisite

- User must login into the application.
- User must stay on any recipe detail page.

Test Script

1. User taps on a Facebook button to login with their Facebook account.
2. User taps on a share button.
3. User inputs their message.
4. User taps on a post status button.
5. The system will share the menu on Facebook.

Test Cases

ID	Description	Input	Expected Result
1	Taps on a Facebook button to measure that the system can provide user can login to Facebook with their Facebook account.	-	User can access to Facebook account.
2	Taps on a share button to measure that the system can provide user puts their message and can post the menu image on their Facebook.		The system allow a user to share menu image on their Facebook.

System Test Case 19 (STC-19): User views ingredient store location on Google Map.

Description

   This test case is a system testing for URS-12. User can view ingredient store location around user location on Google map.

Prerequisite

-   User must login into the application.

Test Script

1. User clicks on find ingredient location button.
2. The system shows each of ingredient store location on Google Map.

Test Cases

ID	Description	Input	Expected Result
1	Click on find ingredient location button and the system will show: - Map from Google Map API - Markers of each store location from database.	-	The system shows each of ingredient store location on Google Map. Its consist of - Map from Google Map API - Markers of each store location from database.

System Test Case 20 (STC-20): User views ingredient store location information.

Description

This test case is a system testing for URS-13. User can view ingredient store location information.

Prerequisite

- User must login into the application.
- User must stay on ingredient store location page.

Test Script

1. User taps on any markers in Google Map.
2. The system redirects to ingredient store location information page.

Test Cases

ID	Description	Input	Expected Result
1	Tap on any markers in Google Map to measure that the system will redirect to ingredient store location page. It will show shop image, name, description, address, sell ingredient, and get direction button.	-	The system can redirect to store location description and show image, name, description, address, sell ingredient, and get direction button.

System Test Case 21 (STC-21): User get direction between user and ingredient store location.

Description

This test case is a system testing for URS-14. User can get direction between user and ingredient store location.

Prerequisite

- User must login into the application.
- User must stay in ingredient store location information page.

Test Script

1. User taps on a get direction button.
2. The system will show a direction between user and ingredient store location.

Test Cases

ID	Description	Input	Expected Result
1	Tap on get direction button to measure that the system can provide a direction between current user location and ingredient store location. It will show a direction path, distance, and time.	-	The system can show a direction between current user and ingredient store location include path, distance, and time.

System Test Case 22 (STC-22): User views list of ingredient.

Description

This test case is a system testing for URS-15. User can view list of ingredient.

Prerequisite

- User must login into the application.

Test Script

1. User taps on an ingredient button.
2. The system will show a list of all ingredient in database.

Test Cases

ID	Description	Input	Expected Result
1	Tap on an ingredient button to measure that the system can redirect to view ingredient information page and show a list of all ingredient in database. It will show ingredient id and name.	-	The system will show a list of all ingredient include ingredient id and name from database.

System Test Case 23 (STC-23): User views ingredient information.

Description

This test case is a system testing for URS-16. User can view ingredient information.

Prerequisite

- User must login into the application.
- User must stay on ingredient view page.

Test Script

1. User taps on any ingredients.
2. The system will show ingredient information.

Test Cases

ID	Description	Input	Expected Result
1	Tap on ingredient in the list to measure that the system can redirect to ingredient information page and it will show a name of ingredient, description, image and type.	-	The system can show any ingredient information from database consist of name, description, image, and type of each the ingredient that selected.